Lexique et analyse sémantique de textes - structures, acquisitions, calculs, et jeux de mots

Mathieu Lafourcade

Université Montpellier 2

Mémoire d'Habilitation à Diriger les Recherches

Spécialité : informatique

Lexique et analyse sémantique de textes
-
structures, acquisitions, calculs, et jeux de mots

par

Mathieu Lafourcade

PRÉSENTATION LE 7 DÉCEMBRE 2011 DEVANT LE JURY COMPOSÉ DE :

Christian Boitet	Université Joseph Fourier - Grenoble 1 - LIG	Examinateur
Vladimir Fomichov	Higher School of Economics National Research University, Moscou	Rapporteur
Violaine Prince	Université Montpellier 2 - LIRMM	Examinatrice
Christian Retoré	Université Bordeaux 1 - LaBRI	Rapporteur
Eric Wehrli	Université de Genève - LATL	Examinateur
Michael Zock	Université de Marseille - LIF	Examinateur
Pierre Zweigenbaum	LIMSI-CNRS	Rapporteur

Version du 28 novembre 2011 à 9:25

Résumé

L'analyse sémantique de textes nécessite en préalable la construction d'objets relevant de la sémantique lexicale. Les vecteurs d'idées et les réseaux lexicaux semblent de bons candidats et constituent ensemble des structures complémentaires. Toutefois, faut-il encore être capable dans la pratique de les construire. Les vecteurs d'idées peuvent être calculés à partir de corpus de définitions de dictionnaires, de thésaurus ou encore de textes. Ils peuvent se décliner en des vecteurs conceptuels, des vecteurs anonymes ou des vecteurs lexicaux - chaque type présentant un équilibre différent entre précision, couverture et praticité. Quant aux réseaux lexicaux, ils peuvent être acquis efficacement via des jeux, et c'est précisément l'objet du projet JeuxDeMots. L'analyse sémantique peut être abordée par l'analyse thématique, et ainsi servir de moyen de calcul à des vecteurs d'idées (bouclage). Nous pouvons modéliser l'analyse comme un problème d'activation et de propagation. La multiplicité des critères pouvant intervenir dans une analyse sémantique, et la difficulté inhérente à définir une fonction de contrôle satisfaisante, nous amène à explorer l'usage de métaheuristiques bio-inspirées. Plus précisément, nous introduisons un modèle d'analyse par colonies de fourmis artificielles. À partir d'un texte, l'analyse vise a construire un graphe contenant les objets du texte (les mots), des objets identifiés comme pertinents (des syntagmes, des concepts) ainsi que des relations pondérées et typées entre ces objets.

Mots-clés

Traitement Automatique des Langues, analyse sémantique de textes, sémantique lexicale, vecteurs d'idées, réseaux lexico-sémantiques, acquisition lexicale, jeux sérieux.

Abstract

The semantic analysis of texts requires beforehand the building of objects related to lexical semantics. Idea vectors and lexical networks seems to be adequate for such a purpose and are complementary. However, one should still be able to construct them in practice. Vectors can be computed with definition corpora extracted from dictionaries, with thesaurii or with plain texts. They can be derived as conceptual vectors, anonymous vectors or lexical vectors - each of those being a particular balance between precision, coverage and practicality. Concerning lexical networks, they can be efficiently constructed through serious games, which is precisely the goal of the JeuxDeMots project. The semantic analysis can be tackled from the thematic analysis, and can serve as computing means for idea vectors. We can modelise the analysis problem as actviations and propagations. The numerous criteria occuring in the semantic analysis and the difficulties related to the proper definition of a control function, lead us to explore metaheuristics inspired from nature. More precisely, we introduce an analysis moodel based on artificial ant colonies. From a given text, the analysis aims at building a graph holding objects of the text (words, phrases, sentences, etc.), highlighting objects considered as relevant (phrases and concepts) as well as typed and weighted relations between those objects.

Keywords

Natural Language Processing, text semantic analysis, lexical semantics, idea vectors, lexical network, lexical acquisition, serious games.

Avant-propos

Dans le cadre du Traitement Automatique du Langage Naturel (TALN), mes thèmes de recherches concernent l'analyse sémantique et l'acquisition de ressources lexicales comme supports à cette analyse. L'exposé de mes travaux porte sur les problèmes liés à la représentation en sémantique vectorielle lexicale, l'acquisition de ces données, ainsi que leur validation et exploitation. Plusieurs définitions et mises en œuvre de l'analyse sémantique sont proposées à l'aide d'algorithmes de propagation. L'acquisition des données en sémantique lexicale est un problème difficile qui peut trouver une solution opératoire via des jeux en ligne proposés à des internautes non spécialistes en linguistique. Ce document se veut être une synthèse des travaux que j'ai menés sur ces questions depuis 1995. Chaque partie aborde une thématique particulière en essayant à la fois d'en présenter les grande lignes, d'offrir souvent une reformulation synthétique avec des résultats non publiés par ailleurs, et enfin d'inclure une ou plusieurs publications représentatives.

Remerciements

Je tiens à exprimer ma reconnaissance à tous ceux qui de près ou de loin ont été impliqués dans les travaux que je présente ici. Je ne pourrais les nommer tous, mais je pense en particulier :

à Ch. Boitet et à V. Prince, *mes mentors*, qui durant toutes ces années ont su non seulement m'encourager, mais également impulser, raturer, biffer, triturer, malaxer, digérer, reformuler ma production scientifique ;

à J. Chauché pour avoir été l'initiateur des vecteurs sémantiques, précurseurs des vecteurs d'idées, mais également pour avoir réalisé Sygmart et SygFran sans lesquels bien peu aurait été fait ;

à F. Guinand avec lequel nous avons joué aux crypto-entomologues dont les approches bioinspirées ont elles-mêmes été inspiratrices d'un modèle d'analyse sémantique de textes à colonies de fourmis artificielles ;

à D. Schwab avec qui de nombreuses idées présentes dans ce mémoire ont mûri et qui a su leur faire prendre vie et surtout les diffuser dans d'autres lieux ;

à V. Zampa sans qui PtiClic n'aurait pu voir le jour ;

à l'ensemble de mes amis et collègues de Malaisie, Thaïlande et Vietnam pour les différents projets que nous avons menés ensembles ;

à l'ensemble de la communauté de JeuxDeMots, non seulement pour avoir joué encore et encore, mais pour un (pas si) petit groupe parmi eux, d'avoir eu envie de faire évoluer l'idée et le logiciel pour devenir ce qu'il est maintenant. Ma reconnaissance en particulier à Caillouteux, k@tsof, Lyn, Mym, N@t, niniefaitlesmots, ..Syl.. et tout ceux que j'oublie.

aux membres du LIRMM et d'ailleurs pour avoir su se prêter amicalement à de nombreuses expériences autour de l'acquisition lexicale ; aux étudiants de master informatique 1 et 2, qui ont bien voulu jouer les cobayes avec enthousiasme ;

aux relecteurs de ce mémoire, en particulier Agnès, Alain, Cédric, Christian, Didier, ma mère Pierrette, Violaine et Virginie qui ont été d'une efficacité redoutable et d'une patience insoupçonnée ;

à Zélie, à David et à Brigitte qui ont supporté tout cela.

Table des matières

Table des figures

Introduction

De nombreuses applications du TALN (Traitement Automatique des Langues Naturelles), comme l'indexation de textes, la traduction automatique, ou encore le résumé automatique, sont potentiellement demandeuses d'une analyse sémantique aussi fine que possible des textes. Il peut s'agir, par exemple, d'extraire la thématique générale d'un document (ou à une granularité plus fine, des paragraphes ou des phrases) sous la forme de sélection de concepts prédéterminés. L'extraction de termes-clés présents dans le texte est également potentiellement utile en indexation de documents pour des moteurs de recherche, avec possiblement le *calcul* de termes absents du texte mais thématiquement pertinents. L'identification des syntagmes, de leurs fonctions syntaxiques et des relations qu'ils peuvent entretenir entre eux est nécessaire en traduction automatique. En résumé automatique, les demandes peuvent être différentes selon qu'il s'agit de contraction (où dans ce cas, des groupes prépositionnels supprimables seront recherchés), d'abstraction (où les relations saillantes entre syntagmes devront être identifiées en vue d'un paraphrasage), ou d'extraction (où l'identification des phrases-clés — celles à la fois porteuses du sens du texte et saillantes — est nécessaire). Quoi qu'il en soit, au moins trois questions délicates se posent si nous nous plaçons dans le cadre d'une analyse sémantique généraliste : 1) quels résultats — structures — souhaitons-nous obtenir ? 2) quels algorithmes seraient susceptibles de calculer ces résultats, et 3) quelles connaissances peuvent servir de support aux algorithmes ?

L'*analyse sémantique de texte* peut être définie comme une tâche visant à effectuer un certain nombre de traitements relatifs à la *sémantique lexicale* (de façon restrictive) ou à la *compréhension du sens* (selon une perspective globale). Nous adoptons volontairement une approche généraliste ne visant aucune application en particulier. Nous pouvons citer parmi les tâches possibles : la levée des ambiguïtés lexicales, le rattachement correct des groupes prépositionnels (parmi ceux autorisés par la syntaxe), la résolution des références (pronoms, adjectifs possessifs, identités, etc.), l'identification des rôles prédicatifs (agent, patient, instrument, etc.), de l'explicitation d'idées ou de concepts implicites. Bien entendu, cette liste est loin d'être exhaustive, et savoir précisément quels sont les traitements utiles n'est pas une question facile.

Des lexiques, des vecteurs et des réseaux

L'acquisition et la structuration des ressources lexicales sont des problèmes en eux-mêmes. Si nous y adjoignons la problématique de la représentation du sens, nous nous trouvons alors à l'intersection entre les bases de données lexicales et la sémantique lexicale. Les ressources s'organisent traditionnellement en lexiques qui constituent des listes d'éléments plus ou moins structurés. Le point d'entrée est usuellement qualifié de *vedette* et sera la forme lemmatique dans la plupart des dictionnaires d'usage, monolingues ou multilingues. Dans le contexte que nous considérons, les lexiques sont d'abord à usage calculatoire (ou machinal, à opposer à un usage humain). Toutefois, les possibilités d'exploration, de lecture et d'exploitation par des humains sont souhaitées, ne serait-ce que pour

vérifier la qualité des données ou les confronter à l'usage. La multiplication sur Internet des dictionnaires à vocation contributive s'affranchissant avec plus ou moins de bonheur des approches lexicographiques traditionnelles en est une illustration. La présentation de ces informations à des humains nécessite en général un traitement informatique, qui est une forme soit de (pré)compilation, soit de mise en cache en fonction des usages. La compilation pourra produire des structures permettant de comparer entre eux des objets lexicaux. Deux types de modèles pour représenter de l'information lexicale (et ontologique) sont envisagés ici : les *vecteurs d'idées* et les *réseaux lexicaux*.

Une première expérience de constitution de lexique via les projets de dictionnaires Français-Anglais-Malais (avec l'Université des Sciences de Malaisie à Penang, FeM [Gut *et al.*, 1996]), puis Français-Anglais-Thai (Université Chulalongkorm à Bangkok en Thaïlande, FeT), et Français-Anglais-Vietnamien (Université de Danang au Vietnam, FeV) a clairement mis en évidence certaines difficultés non seulement dans un cadre multilingue, mais également selon une optique visant à produire des données destinées aux individus mais aussi à un usage machinal [Lafourcade, 1998]. Ces premières expériences ont toutefois permis non seulement d'obtenir des lexiques exploitables ultérieurement, mais également de dégager les problèmes émaillant leur constitution (problèmes de l'inadéquation des outils, de la parcellisation non redondante des données à produire, et difficultés liées à la mise à disposition efficace sous forme électronique). Ces travaux ont évolué vers un cadre plus général par la constitution de bases de données lexicales multilingues avec le projet Papillon (avec G. Sérasset et M. Mangeot, [Mangeot-Lerebours *et al.*, 2003]). L'approche préconise la constitution d'une base pivot d'acceptions interlingues, qui correspondent à des unités de sens, à la fois (potentiellement) reliées entre elles et reliées aux formes monolingues. Cependant, d'un point de vue pratique, ces acceptions restent des symboles, dont l'interprétation n'est pas encodée.

L'encodage au niveau lexical, soit du champ thématique, soit d'une projection du sens, peut être réalisé à l'aide de vecteurs. Les vecteurs conceptuels permettent de représenter efficacement les idées associées à un segment textuel (sens, mot, groupe, texte). Ils forment une structure d'espace vectoriel. Selon leur mode de calcul, ils peuvent représenter un champ thématique, un champ ontologique (relation *est-un*), ou un champ antonymique (travaux avec D. Schwab, [Schwab *et al.*, 2002]). Une fonction de comparaison entre deux vecteurs est la distance angulaire (l'angle que forment deux vecteurs). Celle-ci permet ainsi de retrouver une notion de voisinage et de définir un préordre ou un ordre. Les techniques vectorielles permettent d'obtenir un fort rappel, en particulier pour l'indexation de documents, mais peuvent manquer de précision (par exemple, deux quasi-synonymes peuvent avoir des vecteurs très proches, ce qui n'est pas toujours souhaitable).

Les réseaux lexicaux constituent une approche orthogonale aux approches vectorielles. Ils forment une structure de graphe dont les arêtes sont dotées de valeurs numériques (en l'occurrence et en ce qui nous concerne, un type ou une étiquette, et un poids). Dans ce qui suit, nous considérerons les deux termes de *graphe* et *réseau* comme faisant référence au même type d'objet. L'étiquetage des relations entre les termes permet de représenter autant d'aspects différents que souhaité, qu'ils soient paradigmatiques, syntagmatiques ou ontologiques. Toutefois, s'il existe bien une relation de voisinage entre termes du réseau via la distance entre deux points, celle-ci reste en toute généralité beaucoup plus complexe à calculer qu'entre deux vecteurs.

Pour modéliser des fonctions lexicales [Mel'čuk, 1988, Mel'čuk *et al.*, 1995, Mel'čuk, 1996], le fait de combiner des relations et des vecteurs produit des résultats intéressants (travaux avec V. Prince sur la synonymie relative [Lafourcade & Prince, 2001a] et avec D. Schwab sur l'antonymie relative [Schwab *et al.*, 2002]). La notion d'horizon conceptuel a été introduite afin de rendre compte d'une barrière au-delà de laquelle les vecteurs conceptuels de termes très généraux se ne distinguent plus de termes spécifiques. C'est une des limites des vecteurs conceptuels et de leur application aux fonctions lexicales.

Calculer des vecteurs d'idées

Calculer des vecteurs d'idées peut se faire de multiples façons, qu'il est possible de catégoriser en fonction des données de départ (des textes, des définitions, un réseau lexical, des listes de termes,

etc.) et du type d'algorithme utilisé.

Le corpus d'apprentissage peut être un ensemble de définitions de dictionnaires (à usage humain) qui vient en général en complément à un thésaurus. La méthode de calcul employée peut être, par ordre de complexité croissante : la somme vectorielle des termes saillants (après un filtrage de type TF*IDF), la propagation en remontée sur l'arbre d'analyse, et la propagation en remontée-descente itérée sur l'arbre d'analyse.

Si le corpus d'apprentissage est un réseau lexical, on s'affranchit d'une bonne partie des ambiguïtés présentes dans les définitions. Par contre, selon la ressource utilisée, il n'est pas certain que les termes présents soient systématiquement identifiés comme ambigus. De plus, à partir du réseau lexical, il est possible d'avoir des vecteurs conceptuels étiquetés — c'est-à-dire, qui couvrent une facette particulière (plus seulement les idées associées ou la thématique, mais également les agents ou patients typiques, etc.). Sur les documents, le calcul peut se faire avec un couple de vecteurs en récurrence croisée (travaux avec M. Bouklit, [Bouklit & Lafourcade, 2006]).

La question de la réduction de dimension des espaces vectoriels produits se pose également. Le modèle d'Analyse Sémantique Latente (LSA, [Deerwester *et al.*, 1990b]) la met en œuvre à la fois pour diminuer les structures à manipuler, et pour réduire le bruit. Il semblerait toutefois que l'efficacité voire la pertinence de la décomposition en valeurs singulières et de la réduction de dimension soit de plus en plus remise en cause [Gamallo & Bordag, 2011].

Nous avons entrepris une évaluation sur une combinaison partielle des approches possibles, où il apparaît (sans surprise) que plus la source d'apprentissage est explicite, meilleurs sont les résultats. Pour cela, les réseaux lexicaux sont plus efficaces que les définitions de dictionnaires (nous pourrions argumenter que cela n'est pas étonnant, car le travail d'extraction a déjà été fait). Enfin, la qualité des résultats semble covariante avec la taille des vecteurs, la *quantité relative* de bruit étant globalement constante. Par contre, il est vraisemblable que la *quantité absolue* de bruit générée selon la méthode ou la source d'information puisse aussi être covariante à la taille des vecteurs.

Capturer des relations lexicales et identifier des usages

La possibilité effective d'une acquisition d'informations lexicales via une activité ludique a été démontrée à travers le projet JeuxDeMots ([Lafourcade, 2007], [Joubert & Lafourcade, 2008b], [Joubert & Lafourcade, 2008a]). Cette acquisition prend la forme de la construction incrémentale d'un réseau lexical où les relations sont orientées, typées et pondérées. L'activité ludique est ici la motivation qui pousse les utilisateurs à aboutir à une construction par consensus populaire, sans négociation. Les joueurs n'ont pas besoin d'avoir conscience qu'ils participent à la construction d'une ressource lexicale pour jouer. En effet, lors d'une partie, les joueurs ne sont pas en contact et ne peuvent donc pas *négocier* leurs réponses. PtiCLic (travaux avec V. Zampa, [Lafourcade & Zampa, 2009b], [Lafourcade & Zampa, 2009a]) est une variante de JeuxDeMots mettant l'accent sur la consolidation du réseau via une activité de réattribution de relations pour des couples de termes.

Il est possible de calculer des vecteurs d'idées par émergence à partir du réseau construit dans JeuxDeMots, de façon incrémentale, au fur et à mesure de la construction du réseau. Cette approche est à opposer à celle imposant un recalcul global de l'ensemble des vecteurs (comme dans le cas de LSA).

Contrairement à une approche à partir de définitions, le réseau lexical de JeuxDeMots ne fournit pas directement de sens pour les termes. Toutefois, par identification des sous-cliques maximales ancrées sur une entrée, il est possible de déduire au moins partiellement des usages pour chaque terme. Un usage est la projection d'une acception (au sens classique de la lexicographie) sur un contexte particulier (souvent implicite dans les dictionnaires). L'ensemble des acceptions

est contenu dans l'ensemble des usages (il suffit que le contexte soit général pour qu'un usage corresponde exactement à une acception). Par exemple, le terme *sapin* a comme usage général, entre autres, *sapin>arbre* et comme usage particulier *sapin>Noël*. Disposer des usages de sens semble plus intéressant pour la désambiguïsation lexicale, car ils paraissent souvent plus fidèles aux représentations mentales des locuteurs que les découpages dictionnairiques classiques (travaux avec A. Joubert, [Joubert & Lafourcade, 2008a], [Lafourcade & Joubert, 2010]).

Les usages identifiés pour un terme sont donc inclus dans le réseau lexical, et ce faisant sont indirectement réinjectés dans le jeu. Lors d'une partie, les joueurs peuvent être confrontés à un usage (par exemple *que vous évoque le terme* sapin>arbre *?*) et le renseigner. De plus, ils peuvent dorénavant sélectionner l'usage approprié pour les termes qu'ils proposent durant une partie. Ce bouclage est à l'origine d'un raffinement progressif des termes et des relations dans le réseau. Les champs thématiques associés aux termes deviennent de plus en plus précis au fur à mesure que le réseau se construit, et que ses termes se désambiguïsent.

En environ trois ans de jeu, plus d'un million de relations, entre environ 100 000 termes, ont été capturées. Il semblerait que la distribution des forces d'activation des relations entre termes se conforme à une loi de puissance (pour être précis, sans doute plutôt une loi de Mandelbrot de la forme $f(n) = K/(a+bn)^c$). Il semblerait aussi que la distribution des termes du réseau en fonction du nombre de relations entrantes suive cette même loi. Le réseau construit via JeuxDeMots couvre une partie conséquente de la *longue traîne* de la distribution.

Analyses thématiques et sémantiques

L'*analyse thématique* sera ici vue comme le calcul d'une structure (en général un vecteur d'idées) permettant de représenter le ou les champs lexicaux d'une texte. Pour ce faire, il est possible de procéder de façon statistique (avec des résultats souvent médiocres, d'autant plus que les textes ou segments textuels sont courts) ou bien faire appel à une analyse sémantique, particulièrement utile pour la sélection des acceptions des termes du texte.

L'*analyse sémantique* sera considérée ici comme le calcul qui, à partir d'un texte, produit une structure 1) offrant un support pour traiter un certain nombre de phénomènes linguistiques, et/ou 2) fournissant une ou plusieurs solutions aux problèmes dus à ces mêmes phénomènes. Nous distinguons les deux cas, qui peuvent se manifester simultanément, et souvent se soutenir mutuellement. Par exemple, dans le cadre de la désambiguïsation lexicale, une analyse peut pondérer par ordre de préférence les sens des termes en contexte (premier cas) ou ne retenir que ceux qui sont possibles. La désambiguïsation lexicale peut soutenir le rattachement des groupes, soit thématiquement (*L'avocat a plaidé pour son client à la cour.*), soit sémantiquement (*L'avocat a mangé une pomme dans la cour.*). Dans le premier cas, la connaissance des champs lexicaux majoritaires suffit à faire émerger *avocat>justice* plutôt que *avocat>fruit*. Par contre, dans le second exemple, les thèmes majoritaires sont liés à la nourriture et induisent une interprétation erronée avec la sélection de *avocat>justice*. Des relations prédicatives (concernant le verbe *manger*) ainsi que des opérations minimales d'induction (un *avocat>justice* est un homme ; un homme peut manger ; une *pomme>fruit* peut typiquement être mangée) doivent prendre la suite de l'approche thématique.

Un algorithme, dit de *propagation*, fait circuler des vecteurs d'idées dans une structure de graphe. Il peut s'agir d'un arbre d'analyse morpho-syntaxique : plusieurs variantes sont possibles non seulement selon la nature des ressources lexicales disponibles, mais aussi selon l'application visée (indexation ou traduction). La structure de graphe peut aussi être plus proche de celle des graphes conceptuels, comme c'est le cas pour le projet UNL [1], mais dans ce cas la propagation peut être délicate à mettre en œuvre. En effet, il faut à la fois tenir compte des cycles présents, et aussi de la

1. http://www.vai.dia.fi.upm.es/ing/projects/unl/index.htm

nature sémantique des relations du graphe. Cependant, ce type d'algorithme est 1) trop localiste et produit difficilement des relations à longue portée (entre phrases ou paragraphes) et 2) ne permet pas d'effectuer des modifications structurelles de l'environnement pouvant être lues comme une partie de la solution.

Les algorithmes à colonies de fourmis (travaux avec F. Guinand [Gui2010], et D. Schwab [Schwab, 2005]) permettent de résoudre ce type de problème tout en offrant un certain nombre d'avantages : simplification du contrôle, parallélisation possible, modification possible de l'environnement durant l'exécution, etc. Ici, seuls les vecteurs d'idées sont utilisés comme marqueurs d'information, les fourmis étant des agents transporteurs. La *stigmergie* (communication indirecte par modification de l'environnement) est réalisée via des phéromones artificielles qui sont des marquages se dissipant légèrement à chaque cycle. De plus, sous certaines conditions, les agents peuvent construire des passerelles entre les objets de l'environnement, permettant de proche en proche de sortir d'une démarche strictement localiste. Ainsi, les agents construisent la solution à la fois par renforcement/rejet d'éléments ou par création/destruction de liens entre les objets.

L'adjonction d'un réseau lexical (travaux avec D. Schwab [Schwab, 2005]) permet de dépasser les limites d'un processus fondé uniquement sur des informations thématiques (en plus de celle de l'arbre morphosyntaxique - travaux de J. Chauché avec Sygmart et Sygfran). Il est ainsi possible d'exploiter plus finement des informations prédicatives et valencielles (agent, patient, instrument), des informations de typicalité (lieux typiques : *cheval* → *pré*, moments typiques : *cadeau* → *anniversaire*, etc.) et des fonctions lexicales et ontologiques (synonymie, antonymie, hyponymie, hyperonymie, méronymie, holonymie, etc). Des travaux sur le rattachement de groupes prépositionnels (avec N. Gala, [Gala & Lafourcade, 2007]) ont montré que, dans environ 80% des cas, une information thématique est suffisante pour obtenir un rattachement correct. Dans les autres cas, des relations de typicalité (lieux typiques, instruments typiques, moments typiques) et la prise en compte des restrictions sémantiques sur l'élément régi par la préposition sont nécessaires pour obtenir un rattachement qui correspond à l'intuition.

Applications et ouvertures

Des travaux précédents découlent un certain nombre d'applications qui sortent du cadre strict du TALN, ainsi qu'une ouverture vers quelques pistes de recherche (correspondant à des travaux déjà amorcés).

Le modèle de JeuxDeMots peut trouver des variantes intéressantes permettant à la fois d'acquérir des informations lexicales dont nous ne disposons pas encore (polarité, relation d'inhibition), et de raffiner une relation d'association libre vers une fonction lexicale plus précise. Il s'agit essentiellement de fonctions lexicales ou ontologiques importantes, souvent peu lexicalisées (comme *produit*, *producteur*, par exemple). Le traitement de ces fonctions lexicales est difficile à mettre en œuvre dans le modèle d'origine de JeuxDeMots, car elles présentent un aspect ludique limité : trop peu de réponses, trop immédiates, relativement peu ambiguës. De plus, la sélection automatique de termes intéressants, ou tout simplement valides, pour ces fonctions lexicales peut être difficile et relativement bruitée. Par contre, la reconnaissance d'intrus en contexte peut fournir des informations susceptibles d'aider à l'identification d'usage de termes (cf. partie sur les cliques d'usage avec A. Joubert, [Lafourcade & Joubert, 2010]), ainsi qu'à l'établissement de relations à valeur négative (correspondant à une impossibilité pertinente, par exemple : *autruche agent* voler*).

L'évaluation qualitative du réseau doit également être considérée. Elle peut se faire de façon classique par une approche manuelle via un échantillonnage. Toutefois, nous avons opté dans un premier temps pour une évaluation indirecte via un *jeu de devinette* du type *trouver le mot sur le bout de la langue* [Joubert *et al.*, 2011]. À partir d'un nombre réduit d'indices, est-il possible de faire retrouver

un mot donné ? Des algorithmes extrêmement simples fondés sur l'intersection de vecteurs d'idées calculés à partir du réseau lexical de JeuxDeMots permettent d'obtenir au tout venant un taux de réussite de 70-75%.

Tous les algorithmes d'analyse présentés postulent la pré-existence d'une structure morpho-syntaxique (arbre de constituants ou de dépendance) ou encore une forme approchante de graphes conceptuels (travaux en rapport avec le projet UNL). Il est possible de fournir des pistes de recherche pour que cette partie de l'analyse soit aussi effectuée à l'aide d'algorithmes à fourmis par l'exploitation d'informations disponibles sous formes de réseaux lexicaux. Le texte de départ prend alors la forme d'une chaîne de termes, à partir de laquelle une analyse globale est effectuée (d'où le terme d'holistique). Nous cherchons à nous affranchir de la notion de phase d'analyse (classiquement morphologique, syntaxique, sémantique, logique, etc.) et à viser davantage la résolution de microphénomènes qui, globalement, permettrait de résoudre tout ou partie des problèmes rencontrés. Enfin, des expériences préliminaires ont semblé démontrer que l'adjonction de relations négatives, induisant des phénomènes d'inhibition, augmente de façon très significative à la fois la qualité du résultat et la vitesse de convergence.

Les algorithmes à fourmis exploitant le couplage réseau/vecteurs peuvent être exploités en *ingénierie des modèles* afin de faire du calcul de similarité entre classes (et attributs, ou méthodes) (travaux de R. Falleri avec M. Huchard, [Falleri *et al.*, 2010] et [Falleri *et al.*, 2009]). Il s'agit d'adjoindre des informations lexicales et ontologiques à des processus qui en disposent d'assez peu traditionnellement, et ainsi d'être capable d'effectuer automatiquement des fusions partielles de modèles par identification des objets similaires. Les processus en jeu ici font aussi intervenir des bouclages entre les corpus (ici des modèles de classes) et les sous-réseaux lexicaux construits. Ces travaux se poursuivent dans la direction de la construction d'ontologies de domaines de spécialité, de façon non négociée (dans l'esprit du consensus populaire de JeuxDeMots).

La taille du réseau lexical de JeuxDeMots (plus de 1 200 000 relations, entre 100 000 termes) semble constituer un matériau intéressant afin de mettre à l'épreuve et concevoir de nouveaux algorithmes de visualisation et/ou de classification de lexique (travaux avec M. Hascoët et G. Artignan, [Artignan *et al.*, 2009]). Du point de vue de la clusterisation de termes, les algorithmes doivent faire face à la polysémie des termes du réseau, et reconstituer une ontologie acceptable, à défaut d'être identique à une ontologie classique, ne semble pas trivial.

Un certain nombre d'idées, pour certaines transversales aux sujets abordés, est développé au long de ce document :

> ***Complémentarité***. *Les structures vectorielles, les structures de graphes et les structures ensemblistes pour la représentation en sémantique lexicale sont complémentaires.*
>
> ***Consensus populaire***. *Acquérir des informations lexicales et en particulier des relations entre mots, à l'aide de jeux de consensus populaire non négocié, est une approche opérationnelle.*
>
> ***Acquisition permanente***. *L'acquisition d'information lexicale peut et a intérêt à se faire de façon itérée au sein d'un processus permanent.*
>
> ***Raffinement des structures par bouclage***. *L'identification des usages de termes et le raffinement des relations peut et a intérêt à s'inscrire au sein d'une boucle entre les utilisateurs et les processus.*
>
> ***Activation/inhibition***. *L'analyse sémantique de texte profite au moins autant de l'activation des relations entre termes que de leur inhibition.*

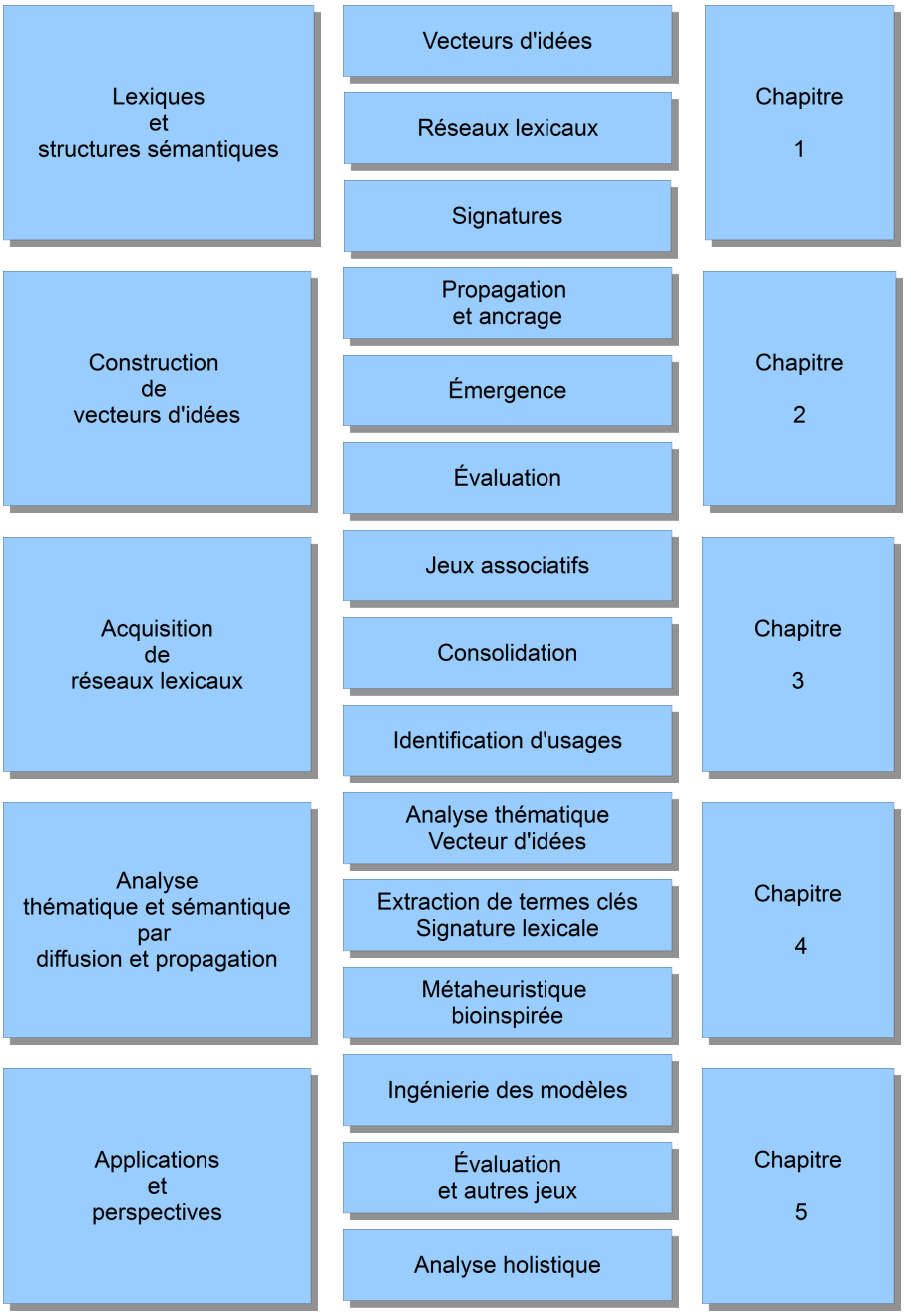


FIGURE 1 – Organisation des chapitres de ce mémoire

CHAPITRE 1

Lexiques et structures sémantiques

Ce chapitre présente les principaux objets que nous manipulons par la suite, à savoir : les vecteurs d'idées, les réseaux lexicaux et les signatures. Nous ne présentons pas en détail comment ces objets peuvent être construits, réservant cela pour les chapitres suivants. Toutefois, nous introduisons les opérations et les caractéristiques les plus marquantes de la façon la plus synthétique possible en essayant de ne pas être redondant avec les articles inclus.

Articles joints

M. Mangeot-Lerebours, G. Sérasset, et M. Lafourcade. *Construction collaborative d'une base lexicale multilingue - Le Projet Papillon* TAL. Volume 44, 1/2, 2003, pages 151 à 176.

M. Lafourcade, V. Prince, D. Schwab. *Vecteurs conceptuels et structuration émergente de terminologies* TAL. Volume 43, 1/2, 2002, pages 43 à 72.

Encadrement - Fabien Jalabert [Jalabert, 2003] (Master 2), Frédéric Rodrigo [Rodrigo, 2004] (Master 2) et Didier Schwab [Schwab, 2001] (DEA) et [Schwab, 2005] (Doctorat).

De quels types de ressources lexicales avons-nous besoin pour les applications du TAL, en particulier celles visant à l'analyse sémantique de textes ? La structure générale d'un tel lexique est l'association entre un terme et un objet représentant sa sémantique (de façon très partielle). Les objets présentés ici sont les vecteurs d'idées, les réseaux lexicaux et les signatures lexicales. Idéalement, de tels lexiques sont exploitables par la machine mais également utiles aux utilisateurs humains.

1.1 Dictionnaires, lexiques et ressources lexicales

Historiquement, les collections d'objets lexicaux qui nous intéressent ici sont plus ou moins structurées et riches en informations, et lorsqu'elles sont destinées à des utilisateurs humains sont alors souvent qualifiées de *dictionnaires*. Traditionnellement, les *lexiques* sont des dictionnaires correspondant à un domaine spécialisé (lexiques et glossaires sont sans doute des objets très proches). Une *ressource lexicale* est beaucoup plus générale et souvent le terme même de *ressource* laisse entendre qu'une utilisation par la machine pour des processus relevant du TAL est envisagée. Cette utilisation par des machines n'est toutefois pas exclusive de celle par des humains.

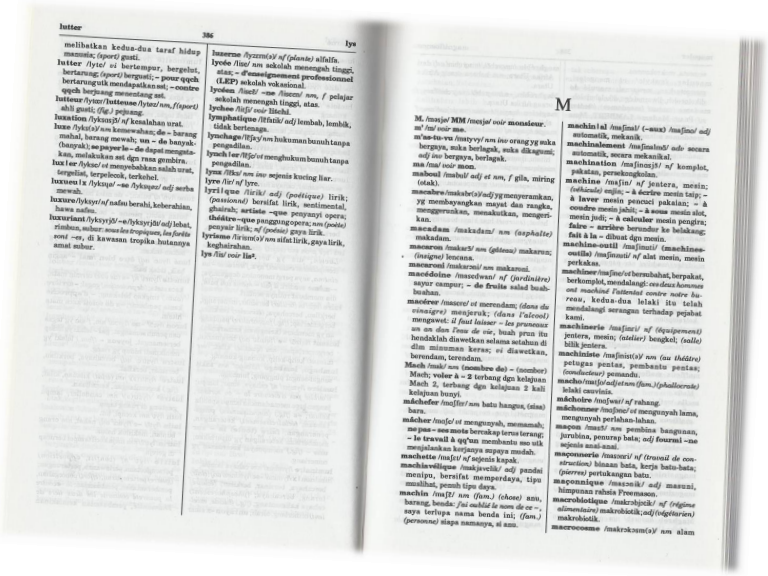


FIGURE 1.1 – Exemple de page du dictionnaire FeM (version imprimée de 1996, [Gut *et al.*, 1996]).

1.1.1 Dictionnaires furcoïdes multilingues

Le projet de dictionnaire FeM (pour French-English-Malay - Français-Anglais-Malais) a permis de construire un dictionnaire trilingue du français vers l'anglais et le malais (en coopération avec l'Université des Sciences de Malaisie - USM - à Penang, l'ambassade de France et du *Dewan Bahasa dan Pustaka* à Kuala Lumpur [1]). Structurellement, les équivalents dans les deux langues cibles sont côte à côte, donnant à l'ensemble une organisation en fourche, d'où la dénomination de *furcoïdes*. Le FeM contient environ 20 000 entrées et 11 000 locutions. Le dictionnaire a été mis en ligne dès 1995 [Lafourcade, 2002b], puis édité sous forme imprimée en 1996 [Gut *et al.*, 1996], suivi d'une seconde version électronique en 1998 [Laf2003]. La chronologie a été plus précisément :

- 1995 (version 0.5) : mise en ligne en mai sous le serveur Alex (développé par M. Lafourcade et déployé à l'USM) ;
- 1996 (version 1.0) : version imprimée distribuée par le *Dewan Bahasa dan Pustaka* (DBP, Kuala Lumpur) [Gut *et al.*, 1996] avec uniquement le français et le malais, distribution sur disquettes Alex et WinHelp de Winsoft [2] comme outils de consultation ;
- 1998 (version 1.5) : complétion de l'anglais par P. Lafourcade, seconde version sur CD-ROM contenant une version statique du dictionnaire [3] et d'une version dynamique via un outil dictionnaire en Java (développé par G. Sérasset) ;
- 2003 (version 2.0) : révision de la partie malaise (par L. Metzger et la Maison du Monde Malais [4]), inclusion dans le projet Papillon, et mise à disposition sur le serveur web de Papillon ainsi que sur un serveur spécifique [5]) [Laf2003].

Serveur de dictionnaires

Le serveur de dictionnaires déployé à l'Université des Sciences à Penang, en mai 1995, a été, à notre connaissance, le premier serveur de ce type sur le Web (avec une technologie originale

1. DBP : http://prpm.dbp.gov.my/
2. Winsoft International : http://www.winsoft-international.com
3. version statique du FeM, pour le dictionnaire général : http://www.lirmm.fr/~lafourcade/HTMFEM_HTMFEMCS/HTMFEM/FEM.HTM et pour le glossaire informatique http://www.lirmm.fr/~lafourcade/HTMFEM_HTMFEMCS/HTMFEMCS/FEMCS.HTM
4. Maison du Monde Malais : http://www.univ-lr.fr/mmm/
5. Serveur FeM au LIG : http://www-clips.imag.fr/cgi-bin/geta/fem/fem.pl

FIGURE 1.2 – Exemple d'affichage du serveur FeM (hébergé au LIG à Grenoble). Le contrôle des informations à afficher est systématiquement joint à l'entrée courante.

[Lafourcade & Sérasset, 1998]). Suite à son succès, ce projet s'est décliné pour le Thaï (FeT) et le Vietnamien (FeV), l'ensemble de ces projets ayant été nommé *Fe**. La problématique de l'encodage et des transcriptions de caractères avec l'affichage simultané de plusieurs systèmes d'écriture avait été abordée dans [Bur2000].

Accompagnant le FeM, un lexique trilingue (Français-Anglais-Malais) du domaine de l'informatique a également été construit et mis à disposition via le serveur de dictionnaires.

La méthode de construction de ces ressources a été relativement classique et essentiellement manuelle, dans un environnement de travail simplifié (sous un traitement de texte du commerce avec des styles de paragraphes comme modalité de typage). Aucun programme lexicographique spécifique n'a été utilisé, à l'exclusion des outils développés en interne pour le projet (convertisseurs de formats et vérificateur de cohérence, générateur des documents Framemaker pour la version imprimée, etc.). Une analyse des difficultés de construction liées à ces projets a été présentée dans [Lafourcade, 1998].

Un objectif annexe de ces travaux a été d'obtenir des données lexicales exploitables pour des tâches de TAL. Les sens en langue cible ont été annotés par des gloses de façon à ce que le lecteur puisse facilement les identifier comme il est d'usage dans les dictionnaires bilingues. Le choix des gloses a été fait de façon à ce qu'une glose soit elle-même une entrée du dictionnaire. La déclinaison en service de dictionnaire accessible sur le Web, ainsi que d'agent dictionnaire, a été vue, à raison, comme un outil pour traducteurs occasionnels ([Lafourcade & Chauché, 1998]).

1.1.2 Bases lexicales multilingues par acceptions

Le *projet Papillon* [Mangeot-Lerebours *et al.*, 2003] a permis de construire une base lexicale multilingue de façon collaborative, ainsi que son environnement informatique associé. Une fois encore, le serveur local obtenu devait être, et est encore effectivement utilisable et utilisé par les traducteurs humains (aussi bien professionnels qu'occasionnels), et peut également fournir des données directement exploitables par des applications informatiques. L'approche par acceptions offre une base solide pour le croisement de dictionnaires, ou encore la création automatisée de dictionnaires bilingues lors de l'ajout d'un lexique dans une nouvelle langue. Dans ([Mangeot-Lerebours *et al.*, 2003] - article joint), l'architecture est présentée ainsi que la stratégie permettant de peupler *a priori* une telle base. Les données des projets FeM, FeT et FeV, ainsi que beaucoup d'autres ont été importées dans le projet Papillon.

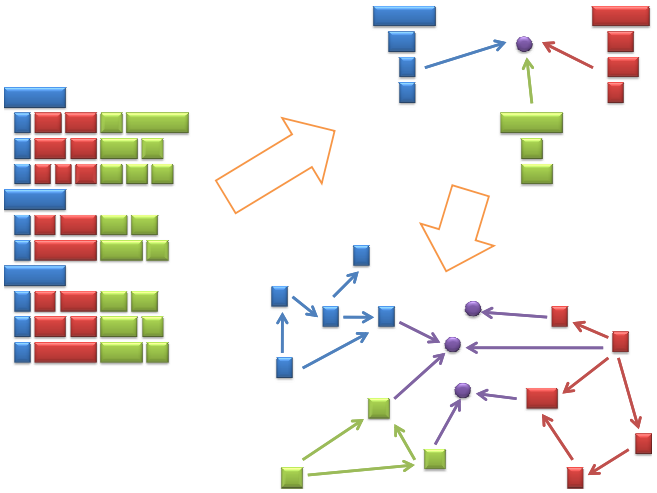


FIGURE 1.3 – *Réseau-ification* des lexiques. À gauche, la structure d'un dictionnaire furcoïde classique. En haut à droite, la structure d'une base lexicale multilingue par acceptions. En bas à droite, la structure d'un réseau lexical (multilingue également).

Dans [Jalabert & Lafourcade, 2002], la question du nommage des sens d'une entrée a été abordée, et a constitué un début d'automatisation du travail fait sur les gloses dans les projets Fe*. La mécanisation d'un tel processus à partir de définitions de dictionnaire reste difficile. Lors du projet JeuxDeMots (c.f. chapitre 3), il s'est avéré que pour cette question précise la mise à contribution de joueurs est efficace. La question de la construction à faible coût (calculatoire et humain) de vecteurs conceptuels multilingues a été explorée dans [Lafourcade *et al.*, 2004] et celle de réseaux lexicaux multilingues de façon contributive a été abordée dans [Zampa & Lafourcade, 2009a].

1.1.3 Lexiques = réseaux ?

Depuis les projets Fe*, Papillon et plus récemment JeuxDeMots, nous remarquerons que la structure du lexique a tendance à se transformer en réseau. Outre que la structure de graphe est à la fois générique et aisée à implémenter, nous supposons qu'elle est adaptée aux modèles de structure lexicale (multilingue ou non) que nous manipulons. Les informations s'organisent naturellement en arbres, et l'adjonction de liens transverses entre éléments du lexique constitue intuitivement des graphes.

1.2 Vecteur d'idées : une structure d'espace

L'approche vectorielle en TAL est issue de la sémantique distributionnelle (Harris, Hirschman) et son implémentation la plus connue en recherche documentaire est le système SMART [Salton & MacGill, 1983], qui a abouti au modèle dit "vectoriel standard" (Salton, MacGill). Les aspects thématiques des segments de textes (documents, paragraphes, syntagmes, etc.) peuvent être représentés par des vecteurs d'idées. De tels vecteurs sont utilisés depuis longtemps pour la représentation du sens par le modèle LSI [Deerwester *et al.*, 1990a] et dans les études de l'analyse de la sémantique latente (LSA) en psycholinguistique. En linguistique computationnelle, J. Chauché [Chauché, 1990] a proposé un formalisme pour la projection de la notion linguistique de champ sémantique dans un espace vectoriel dont le présent modèle est inspiré.

Un élément textuel est donc associé à un vecteur d'un espace de dimension finie. Dans le modèle classique, chaque dimension est associée à un élément textuel (terme d'indexation), cette association étant l'objet d'un apprentissage le plus souvent automatique. Plutôt que de relier dimensions et éléments textuels, il est possible de se placer dans le cadre de la linguistique componentielle (de Hjelmsle) et de considérer les dimensions comme autant de champs sémantiques. Dans tous les cas, l'attrait que suscitent les vecteurs provient des opérations mathématiques relativement simples permettant de les manipuler ou de les comparer. La difficulté principale vient de l'interprétation linguistique que l'on peut attacher à ces opérations. Le détail des opérations associées aux vecteurs est donné en annexe de ce chapitre.

L'hypothèse qui considère un ensemble de concepts comme un générateur de l'espace lexical d'une langue a été longuement discutée et décrite dans [Roget, 1852a]. Cette hypothèse est connue sous le nom d'*hypothèse thésaurus*. Les mots polysémiques combinent les différents vecteurs correspondant aux différents sens, et à chaque sens peut être associé un vecteur (voir [Lafourcade & Sandford, 1999]) qu'il soit lexicalisé ou non.

1.2.1 Vecteurs conceptuels et vecteurs anonymes

Un **espace conceptuel** est composé d'un nombre fini de dimensions, chacune correspondant à un concept. Cet ensemble de concepts, défini *a priori*, est généralement issu d'un thésaurus (Larousse [Larousse, 1992b] ou Roget [Roget, 1852a], par exemple). L'index du thésaurus fournit l'association entre un item lexical et le ou les concept(s) au(x)quel(s) il est associé. Dans le cas du thésaurus Larousse, 873 concepts sont ainsi définis, et chaque terme monosémique renvoie en général à un de ces concepts, ou plus rarement, à plusieurs. Les termes polysémiques, soit renvoient directement à plusieurs concepts, soit énumèrent leurs sens qui eux sont associés aux concepts. Donc, chaque composante d'un **vecteur conceptuel** est une valeur de $\mathbb{R}^+$, la correspondance de la composante au concept étant connue et définie *a priori*.

Il est difficile de soutenir l'hypothèse que les concepts sont indépendants ; d'ailleurs, un thésaurus en fournit la plupart du temps une organisation arborescente. Nous remarquerons que chez Salton (entre autres), *on fait comme si* les composantes des vecteurs étaient indépendantes pour l'organisation des calculs, mais que personne n'est dupe. Donc, deux possibilités s'offrent à nous, la première étant d'ignorer le problème (faire comme si), la seconde étant de définir un mode de construction des vecteurs reflétant l'interdépendance des concepts (voir chapitre 2). Dans ce dernier cas, l'espace conceptuel est un générateur d'un espace de dimension plus petite.

L'interprétation de la valeur d'une composante étant thématique, il n'est pas aisé de modéliser ces valeurs par tout $\mathbb{R}$ (c'est-à-dire avec des valeurs négatives). En particulier, la notion de similarité négative est difficile à interpréter. Il s'agirait vraisemblablement d'éléments inhibés. Cependant, un cosinus négatif entre deux vecteurs est-il vraiment interprétable comme une opposition entre les objets correspondants ? Et si oui, de quelle nature ? Par exemple, il y a une opposition de nature différente entre *chaud* et *froid* d'une part et *migraine* et *aspirine* d'autre part. LSA rencontre le même problème avec la possibilité de similarité négative entre deux objets, mais sa déclinaison probabiliste (PLSA) s'en affranchit.

Dans un **espace anonyme**, la dimension est connue, mais pas l'interprétation associée à chaque dimension. Il n'est pas nécessaire de partir d'un ensemble défini de concepts, ce qui facilite grandement la construction. Toutefois, un **vecteur anonyme** n'est pas facilement décodable car il n'y a pas d'associations directes entre composantes et concepts. Par contre, il est intéressant que la dimension (bien que fixée au départ) soit aussi grande que souhaitée, et ainsi d'obtenir une structure où l'occupation de l'espace s'ajuste en fonction du lexique (général, de spécialité ou mixte). Une construction itérée permet en outre d'obtenir un ajustement dynamique s'effectuant lors de l'ajout de nouveaux termes ou de nouvelles définitions (l'ajout d'un lexique terminologique de spécialité peut alors se faire incrémentalement sans un recalcul complet). La construction d'un tel espace est détaillée dans

le chapitre 2.

Nous noterons que les vecteurs de ce type sont similaires à ceux obtenus par LSA, si ce n'est qu'aucune réduction de dimensionnalité n'est impliquée. À ce propos, il faut signaler un résultat intéressant : toutes nos expériences de réduction aussi bien sur les vecteurs anonymes calculés par émergence que sur les vecteurs conceptuels, n'ont abouti qu'à une réduction de la précision et à l'augmentation du bruit. Ce type de résultat semble confirmé dans [Gamallo & Bordag, 2011].

1.2.2 Opérations sur les vecteurs

Le détail des opérations sur les vecteurs est donné en annexe de ce chapitre. La fonction de comparaison classique entre deux vecteurs est la **similarité cosinus**, qui n'est rien d'autre que le produit scalaire entre deux vecteurs (supposés normés). Une fonction de distance, ayant donc les trois propriétés de réflexivité, symétrie et inégalité triangulaire, sera souvent préférée à la similarité. Ces propriétés permettent d'avoir une interprétation géométrique naturelle des manipulations faites sur ces vecteurs. Le passage de la similarité cosinus à la **distance angulaire** D_A se fait via l'arcosinus du produit scalaire.

De la notion de distance angulaire découle la notion de **voisinage**. Soit $\mathcal{E}$ un espace peuplé de n vecteurs $V_1, \cdots, V_n$, le voisinage d'un vecteur V, noté $\vartheta(V)$, est l'ensemble des vecteurs de $\mathcal{E}$, ordonné par leur distance à V :

$$V_i \leq_{A,V} V_j \quad \text{ssi} \quad D_A(V, V_i) \leq D_A(V, V_j)$$

$\mathcal{E}$ peut alors être ordonné d'une ou plusieurs façons par un tri topologique tel que :

$$(\forall i, j) \quad i \leq j \Rightarrow V_i \leq_{A,V} V_j$$

Alternativement, le voisinage pouvant être vu comme un ensemble partiellement ordonné de couples (vecteur, distance), les opérations ensemblistes classiques s'appliquent à condition de préciser quelle est l'opération d'agrégation pour calculer le poids de chaque élément dupliqué (le *max*, le *min*, la *moyenne arithmétique*, la *moyenne géométrique*, etc.).

Les vecteurs peuvent être additionnés (opération notée $\oplus$), le vecteur résultat étant alors l'idée moyenne de ses arguments. Le produit terme à terme de deux vecteurs (noté $\otimes$) peut être interprété comme l'intersection des idées présentes dans chacun des vecteurs. La combinaison des deux offre une opération de contextualisation (notée Γ), à la base de la définition de la synonymie relative [Lafourcade & Prince, 2001a].

En toute généralité, nous avons :

$$\forall f \qquad \vartheta(V_1 \oplus V_2) \neq \vartheta(V_1) \cup_f \vartheta(V_2)$$

Quelle que soit l'opération d'agrégation f choisie, le voisinage de la somme de deux vecteurs n'est pas identique à la composition par f des voisinages de chacun de ces vecteurs. Nous rappelons que l'opération d'agrégation définit l'ordre des éléments du voisinage en définissant le mode de calcul de la distance à V. Dans le max du maximum (*max*), la distance de V_i à V sera la plus grande des distances entre ces mêmes vecteurs dans chacun des deux voisinages. Nous procéderons de façon similaire pour *min*, les différentes moyennes, etc.

Il est certes possible de précalculer un voisinage (en pratique approximatif car de taille réduite) pour chaque terme du lexique, mais le voisinage de l'addition des vecteurs de ces deux termes doit être recalculé. Il ne peut pas, dans le cas général, être calculé par composition des voisinages de chacun des termes.

1.2.3 Vecteurs et fonctions lexicales

Le modèle des vecteurs d'idées permet le calcul de relations entre termes, en particulier concernant la synonymie et l'antonymie relatives. D. Schwab a mené des travaux sur l'antonymie et a montré qu'il était possible de construire une fonction vectorielle sur trois types d'antonymie ([Schwab, 2001]) [Schwab *et al.*, 2002]. Certaines des fonctions lexicales, telles que définies par I. Mel'čuk, peuvent être modélisées à l'aide d'une approche vectorielle. Il s'agira essentiellement et de façon quelque peu surprenante, de celles qui sont ontologiques et *a priori* les moins lexicalisées.

Une analyse en profondeur de la structuration terminologique à l'aide des vecteurs conceptuels dans [Lafourcade *et al.*, 2002a] a montré que ces derniers pouvaient fournir un support intéressant à la modélisation des fonctions ontologiques (comme l'hyperonymie) et lexicales (comme la synonymie et l'antonymie - [Lafourcade & Prince, 2001b]).

1.2.4 Construction et utilisation de vecteurs

Le détail de la construction de vecteurs conceptuels et anonymes étant donné dans le chapitre suivant, nous nous bornons ici à en esquisser les principes. Pour les vecteurs conceptuels, un noyau de vecteurs élémentaires est nécessaire. Pour simplifier, si nous avons un espace vectoriel défini sur n concepts, nous leur associons n vecteurs unité de cet espace, et posons qu'ils en forment une base. La construction d'un vecteur d'un terme se fera donc par combinaison linéaire des vecteurs de base.

Dans le cas des vecteurs anonymes, nous ne disposons pas d'un ensemble de concepts initiaux, une alternative doit donc être trouvée. Les vecteurs des termes seront donc initialement tirés au hasard dans l'espace et une application itérative de fonctions d'agrégation et de séparation sera effectuée (à l'image d'un ensemble de particules animées par une force d'attraction à longue portée, et d'une fonction de répulsion à courte portée).

Les vecteurs conceptuels ont largement été utilisés pour l'étude du peuplement automatique d'une partie de la base lexicale du projet Papillon [Mangeot-Lerebours *et al.*, 2003]. Il ont également été mis à profit dans le cadre de l'aide à la construction d'ontologies sur un domaine spécifique [Abrouk & Lafourcade, 2006]. Les vecteurs conceptuels ont également été utilisés comme éléments d'indexation de textes, lors d'une expérience informelle de construction d'un moteur d'indexation et recherche de textes (projet Converse). Ce qu'il faut en retenir, c'est que le rappel est particulièrement bon, mais la précision faible, et *a fortiori* lorsqu'il s'agit de requêtes fortement lexicalisées (par exemple, des entités nommées).

Selon le domaine d'application, une base de vecteurs conceptuels construite dans un cadre général peut s'avérer avoir une précision insuffisante. C'est pourquoi nous avons défini dans [Lafourcade *et al.*, 2002a, Lafourcade, 2002a] le passage de vecteurs d'un espace vers un autre (fonction de pliage et dépliage). Un résultat remarquable est que :

l'extension locale de l'espace induit sa contraction en dehors de la localité.

Nous observons en effet que, les termes d'un domaine de spécialité pour lesquels l'espace est étendu (par dépliage ou dilatation) voient leurs vecteurs se discriminer, mais qu'à l'inverse les termes en dehors de la spécialité voient leurs vecteurs devenir plus similaires (contraction). Il semble que ce résultat est une conséquence inévitable de l'approche vectorielle et de la fonction de similarité.

Nous avons montré dans [Lafourcade, 2001b] que les vecteurs conceptuels peuvent dans le cadre de la traduction automatique, fournir une approche intéressante pour le transfert lexical. Plusieurs autres travaux ont exploré plus avant cette approche, en particulier sur la création d'une matrice de transfert en français et anglais (J.-M. Delorme - encadrement de V. Prince et J. Chauché, [Delorme, 2003]) et sur la construction de vecteurs par ressources bilingues [Rodrigo, 2004].

1.3 Réseau lexical : une structure de graphe

Un réseau lexical est une structure de graphe mettant en jeu des nœuds correspondant au sens le plus réducteur, à des éléments du lexique, tandis que les arcs correspondent à des relations binaires. Contrairement aux vecteurs, ce type de structure offre une précision plus forte mais un rappel moindre.

Un réseau lexical est donc une structure discrète où les objets sont directement reliés, contrairement aux vecteurs où la relation à d'autres termes est approchée localement via une fonction de similarité et globalement via la fonction de voisinage correspondante. Nous noterons que le type de la relation dans le réseau trouve son pendant dans la définition de la fonction de similarité. Il est aussi possible de voir le réseau comme une définition des termes en extension par rapport aux autres termes du lexique. Un vecteur correspond à une définition en intension. Un terme du réseau explicite ses voisins, et il est possible d'en déduire ses caractéristiques. Dualement, un vecteur est une énumération de caractéristiques dont il est possible de déduire des voisinages.

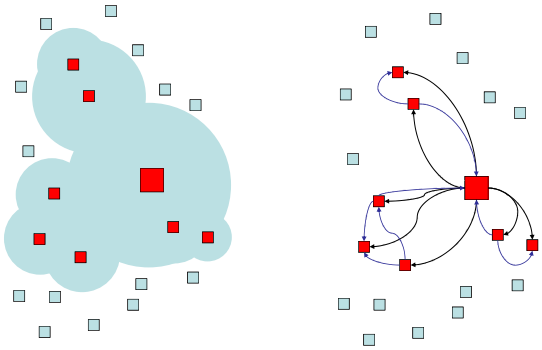


FIGURE 1.4 – Vecteurs et réseau, des structures duales vis-à-vis du voisinage ?

1.3.1 Définition générale

Un réseau lexical est un graphe où les nœuds représentent des termes (ou n'importe quel type d'objet lexical) et les arcs des relations lexicales ou d'ordre sémantique (ontologiques, prédicatives, etc.). En toute généralité, les nœuds contiennent des segments textuels, aussi bien des termes simples que composés, des locutions ou encore des raffinements de sens (par exemple *souris>rongeur* et *souris>informatique*). Lorsque les nœuds correspondent à des sens ou des concepts, on parle généralement de réseaux sémantiques. Les nœuds et les arcs sont pondérés, traduisant respectivement (et de façon extrêmement vague) l'importance en usage du terme et la force de la relation.

Princeton WordNet ([Fellbaum, 1988], [Harabagiu *et al.*, 1999], url [6]) est le réseau lexical (disponible pour l'anglais) qui fait référence dans le domaine. Il contient environ 150 000 termes et 180 000 relations. Les relations principales sont taxonomiques (hyperonymie et l'hyponymie directe ou héritée - l'holonymie et la méronymie). L'organisation des entrées est réalisée autour de *synsets*, des regroupements de lexèmes supposés synonymes (ou quasi-synonymes). Des définitions dictionnairiques classiques ainsi que des formes logiques de ces dernières (eXtended Wordnet de l'Université des Sciences du Texas-Dallas, pour WordNet 3) sont également présentes.

Pour le français, B. Sagot ([Sagot & Fier, 2008], url [7]) a construit de façon semi-automatique le

6. http://wordnetweb.princeton.edu/perl/webwn
7. http://alpage.inria.fr/~sagot/wolf.html

WOLF qui est un *Wordnet Libre du Français*. Le WOLF a été construit à partir de WordNet et de diverses ressources multilingues, à l'aide de processus de croisement automatique.

HowNet ([Zhendong, 2009], url [8]) est une base de connaissances bilingue (anglais-chinois) contenant des relations sémantiques entre des lexèmes, des concepts et des attributs. Les unités atomiques de sens (1 500 sémèmes) sont combinées de façon à construire des concepts. La base contient 65 000 concepts pour le chinois et environ 75 000 pour la partie anglaise. Les informations contenues dans HowNet pour chaque entrée, sont en premier lieu l'hyperonyme suivi de traits distinctifs en nombres variables. Cette définition sémantique/conceptuelle est alors un *sous-ensemble ordonné* de l'ensemble des sémèmes. Par exemple, une entrée simplifiée a la forme suivante (pour l'anglais uniquement) :

> E (entrée) = journaliste
> DEF (définition) = human | (related) occupation | (agent) gather | (agent) compile | (related) news

Dans la définition ci-dessus, la relation est typée (le type est donné entre parenthèses). Ici, un type peut être soit l'association générale (related), soit un sujet possible d'un sémème verbal (agent).

Le réseau du français obtenu de manière contributive avec JeuxDeMots contient environ 232 000 termes et plus d'un million de relations (1 200 000 environ). Une cinquantaine de types de relations est disponible. Un terme polysémique est destiné à être raffiné en ses usages (une entrée le sera effectivement ou pas, selon l'état d'avancement de l'indexation).

1.3.2 Réseaux et fonctions lexicales

Les fonctions lexicales sont représentées explicitement via les relations du réseau lexical (voir figure 1.5). Par exemple, un lien d'antonymie entre les termes *chaud* et *froid* sera interprété comme une valeur de la fonction *anto* : anto(chaud) = froid. Une *fonction lexicale* de Mel'čuk a comme valeur un ensemble de lexèmes, par exemple magn(fièvre) = {forte, élevée, de cheval}. Cependant, on note, par abus de langage, magn(fièvre) = élevée et non élevée ∈ magn(fièvre).

En pratique, il est fort probable qu'un réseau lexical ne soit jamais complet et qu'il soit nécessaire d'avoir à parcourir le graphe pour évaluer des valeurs de fonctions lexicales entre deux termes. Par exemple, nous avons :

- *coucou* $\xrightarrow{isa}$ *oiseau*
- *oiseau* $\xrightarrow{isa}$ *animal*
- *oiseau* $\xrightarrow{isa}$ *auge* — *(Maçonnerie)* Sorte de hotte dont les manœuvres se servent pour porter le mortier sur leurs épaules. (d'après http://fr.wiktionary.org/wiki/oiseau)

Nous pouvons certes en déduire que *coucou* $\xrightarrow{isa}$ *animal* mais également que **coucou* $\xrightarrow{isa}$ *auge* ce qui est manifestement faux. D'une part, nous sommes confrontés à la polysémie du lexique. Ensuite, à moins de nous limiter à un réseau lexical selon une vue réductrice et ontologique, une acception d'un terme peut avoir plusieurs hyperonymes aussi bien de hauteurs différentes, que de vues contrastives. Par exemple :

- *chat* $\xrightarrow{isa}$ *félin*
- *chat* $\xrightarrow{isa}$ *mammifère*
- *chat* $\xrightarrow{isa}$ *animal*

- *chat* $\xrightarrow{isa}$ *animal de compagnie*
- *chat* $\xrightarrow{isa}$ *carnivore*
- *chat* $\xrightarrow{isa}$ *vertébré*
- *chat* $\xrightarrow{isa}$ *chasseur*

Toutes ces relations sont correctes, et simplement, soit elles ne se focalisent pas sur les mêmes aspects (ou facettes), soit elles court-circuitent une partie plus ou moins importante de la hiérarchie

8. http://www.keenage.com

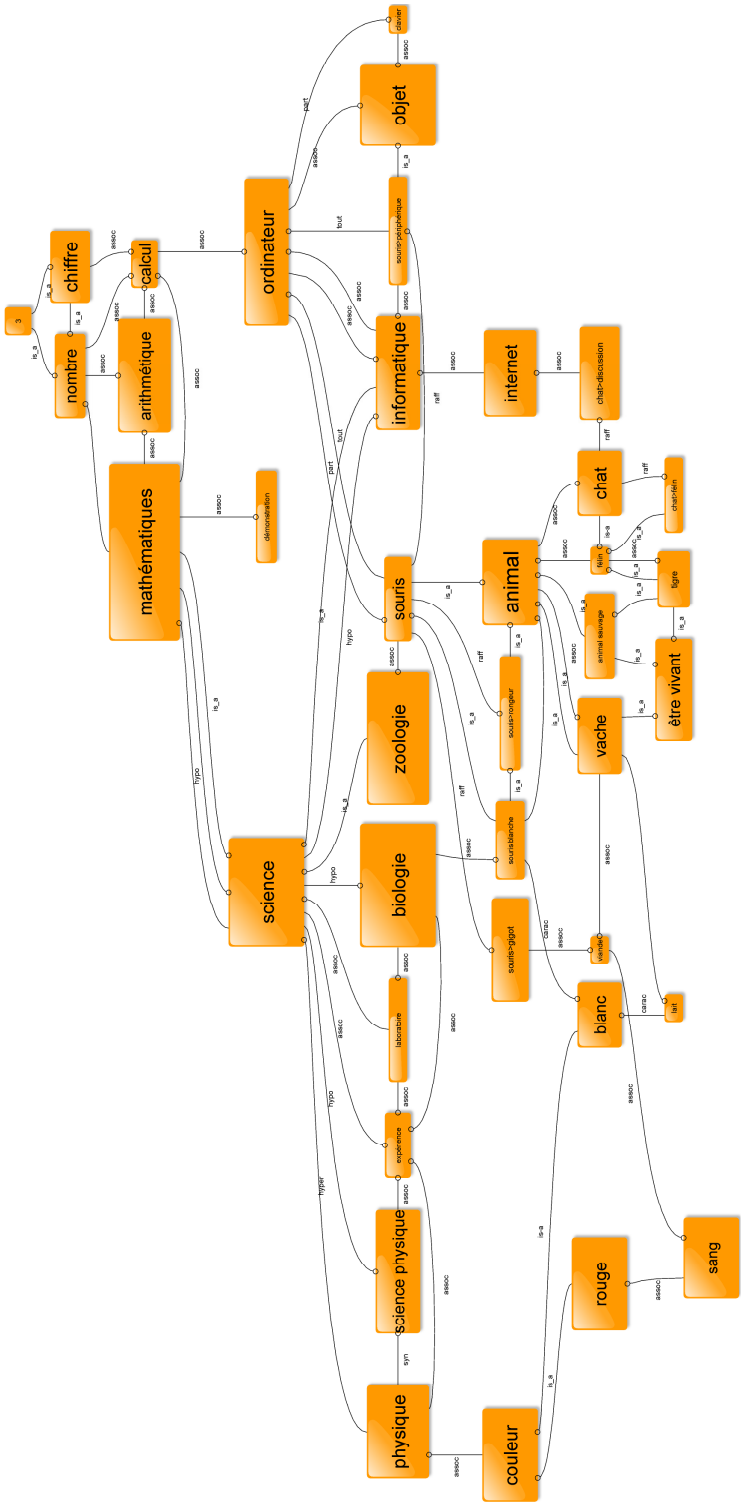


FIGURE 1.5 – Un extrait simplifié de réseau lexical. La taille d'un nœud est fonction de la fréquence d'usage du terme associé.

d'hyperonymes (pour peu que nous puissions en définir une unique précisément).

La question de savoir s'il est nécessaire ou désirable de *nettoyer* le réseau lexical reste largement ouverte. Dans ce cadre *nettoyer* veut dire le transformer de façon à obtenir une ontologie *propre* (hiérarchie sans cycle, sans redondance, sous la forme d'un treillis ou d'un arbre si nous excluons l'héritage multiple) qui, par exemple, pourrait ressembler à cela :

- *chat* $\xrightarrow{isa}$ *félin*
- *félin* $\xrightarrow{isa}$ *mammifère*
- *mammifère* $\xrightarrow{isa}$ *animal*
- *chat* $\xrightarrow{isa}$ *animal de compagnie*
- *félin* $\xrightarrow{isa}$ *carnivore*
- *mammifère* $\xrightarrow{isa}$ *vertébré*
- *chat* $\xrightarrow{isa}$ *chasseur*

Les relations ajoutées sont clairement désirables, mais il nous semble que dans le cadre du TAL et de l'analyse de texte, un nettoyage trop poussé qui supprimerait la redondance est une fausse bonne idée. La présence de poids dans les relations permet de hiérarchiser l'importance relative de chacune d'elles. De ce fait, la multiplicité des relations est plus un avantage qu'un handicap.

1.3.3 Construction de relations et mixité

L'**extraction automatique de relations** entre termes à partir de corpus a été l'objet de nombreux travaux, notamment sur l'hyperonymie et l'hyponymie. Hearst [Hearst, 1992] puis Pantel [Pantel & Pennacchiotti, 2006] utilisent des patrons lexico-syntaxiques afin de repérer et extraire des candidats. Une telle approche n'est pas toujours aisée, les patrons étant délicats à créer, et les ambiguïtés lexicales ou syntaxiques omniprésentes (la synonymie pouvant prendre la même forme syntaxique que l'hyperonymie voire l'antonymie). De plus, les patrons sont bien entendu particulièrement dépendants de la langue, mais également du niveau de langue et sans doute en partie du domaine de spécialité.

Dans [Lafourcade, 2003a], une approche à l'aide de patrons flous combinés et de vecteurs conceptuels et redondants a été adoptée afin d'extraire et de discriminer la méronymie de l'hyperonymie (les deux pouvant prendre des formes syntaxiques similaires). Finalement, devant de telles difficultés, l'approche que nous avons adoptée pour le vocabulaire général (chapitre 3) consiste à solliciter des non spécialistes de **manière contributive**. La combinaison d'un graphe de relation lexicale (sous une forme approchante de réseaux sémantiques) et de vecteurs d'idées semble constituer une approche intéressante pour la modélisation de l'antonymie et de l'hyperonymie ([Lafourcade & Prince, 2004], [Schwab *et al.*, 2005], [Schwab *et al.*, 2007]). Toutefois, l'approche délexicalisée représentée par les vecteurs d'idées atteint sa limite à mesure de l'éloignement de l'*horizon* constitué par les concepts. Dans une classification ontologique, la remontée vers des termes abstraits (au-delà des concepts de l'espace) nous confronte à des vecteurs que l'on ne peut que difficilement distinguer de ceux de termes très spécifiques. La conceptualité des vecteurs d'idées est un atout très clair pour le rapprochement thématique entre termes, mais devient un inconvénient majeur dans les situations où la lexicalisation prend plus d'importance que l'appartenance au champ lexical. Cela arrive particulièrement quand le vocabulaire employé est très spécifique, ou à l'inverse quand il demeure dans un domaine très général.

1.4 Signature : une structure ensembliste lexicalisée

Un inconvénient majeur des vecteurs d'idées est qu'ils font disparaître les termes dont ils sont issus au profit de concepts (ils sont anonymisants). Par exemple, si nous supposons que les termes *chat* et *matou* sont très proches, il sera difficile de distinguer, uniquement sur la base des vecteurs, deux segments textuels contenant l'un ou l'autre. Par ailleurs, le réseau lexical ne constitue pas une structure de résultats ou d'indexation facilement manipulable. La structure de signature est un compromis entre les deux approches, et peut être vue comme un *vecteur lexicalisé* dont la dimension est variable.

Une *signature lexicale* (nous parlerons parfois improprement de *vecteur lexical*) d'une forme lexicale est un ensemble fini typé de termes pondérés, 2-normé (voir annexe de ce chapitre - normes). Le type correspond à une combinaison de relations lexicales. Nous considérerons que le type vide correspond par défaut à une association thématique (une *idée*). La cardinalité d'une signature est quelconque, toutefois, une signature vide (de cardinalité 0) sera estimée comme mal formée (ou associée à un terme pour lequel nous n'avons aucune information).

Par exemple, nous avons les signatures suivantes :

> S(chat) = (félin :0.47 ; animal :0.31 ; chien :0.25 ; chaton :0.22 ; griffe :0.2 ; patte :0.19 ; queue :0.18 ; souris :0.18 ; chat sauvage :0.17 ; poil :0.17 ; MSN :0.17 ; moustache :0.17 ; miauler :0.16 ; dormir :0.16 ; mammifère :0.16 ; ronronner :0.15 ; être vivant :0.14 ; minou :0.14 ; animal de compagnie :0.14 ; chatte :0.13 ; félidé :0.1 ; chat domestique :0.09 ; siamois :0.09 ; tigre :0.08 ; griffer :0.08 ; pattes :0.08 ; minet :0.07 ; caresser :0.07 ; chasser :0.07 ; griffes :0.06 ; yeux :0.06 ; manger :0.06 ; chat de gouttière :0.05 ; matou :0.05 ; oreille :0.05 ; chatter :0.05 ; animal domestique :0.05 ; angora :0.05 ; Vétérinaire :0.04 ; miaou :0.04 ; se laver :0.04 ; gouttière :0.04 ; persan :0.04 ; Félix :0.03 ; lait :0.03 ; les animaux :0.03 ; s'étirer :0.03 ; chat>félin :0.03 ; boire :0.03 ; moustaches :0.03 ; poils :0.03 ; titi :0.03 ; jouer :0.03 ; tête :0.03 ; coussinet :0.03 ; sphinx :0.03, $\dots$)

> S(chat, hyperonyme) = (félin :0.71 ; animal :0.59 ; mammifère :0.24 ; félidé :0.17 ; être vivant :0.13 ; animal de compagnie :0.12 ; bête :0.08 ; compagnon :0.07 ; carnivore :0.06 ; quadrupède :0.05 ; animal domestique :0.05 ; vertébré :0.05 ; chasseur :0.05 ; discussion :0.04 ; jeu :0.03 ; ingrédient de cuisine :0.02 ; lieu :0.01)

> S(*chat*>*félin*, hyperonyme) = (animal :0.81 ; félin :0.4 ; animal>zoologie :0.32 ; animal de compagnie :0.27 ; être vivant :0.06 ; compagnon :0.01 ; animal domestique :0.01 ; félidé :0.01 ; ingrédient de cuisine :0 ; quadrupède :0 ; lieu :-0.04)

> Le terme *chat* est ambigu avec au moins trois acceptions notables, à savoir : l'animal, la conversation via internet (le tchat) et le jeu d'enfant (jouer à chat). Ceci explique que parmi les hyperonymes de *chat* nous trouvions *discussion* et *jeu*. L'acception dominante est clairement l'animal si l'on se fie à la distribution des termes associés.

En toute généralité, les pondérations sont dans $\mathbb{R}$, c'est-à-dire qu'elles peuvent être négatives. Cela permet ainsi de représenter des *interdictions*. Par exemple :

> S(autruche, agent^{-1}) = (courir :0.3 ; avaler :0.3 ; manger>se nourrir :0.21 ; manger :0.21 ; pondre :0.21 ; voler :-0.39 ; voler>déplacement aérien :-0.73)

Il est possible d'associer une signature lexicale à tout type d'objet : terme, syntagme ou texte. La taille d'une signature lexicale n'est pas fixe (contrairement aux vecteurs) et peut varier dans le temps (à mesure de l'indexation). De plus, ses composantes n'étant pas conceptuelles mais lexicales (à nouveau contrairement aux vecteurs), elles peuvent être ambigües.

> S(avocat, caractéristique) = (vert :0.56 ; mûr :0.37 ; véreux :0.31 ; pelé :0.31 ; véreux>corrompu :0.28 ; pourri :0.28 ; vert>couleur :0.22 ; véreux>plein de vers :0.16 ; corrompu :0.16)

Dans la signature ci-dessus, le terme *pourri* est ambigu et ses raffinements éventuels ne sont pas présents.

Pourquoi définir cette notion de signatures ? Les signatures lexicales sont typées et (c'est lié) permettent des activations négatives avec des possibilités d'interprétation. Surtout, contrairement aux vecteurs (saltoniens, notamment), il s'agit d'associations et non pas d'une représentation du contenu (sauf à considérer qu'un terme, à l'image d'un texte dispose d'un *contenu*). D'un point de vue pratique, il est possible de considérer les signatures lexicales comme une généralisation typée des vecteurs saltoniens.

1.4.1 Fonction d'activation

Les fonctions de similarité vectorielle peuvent être utilisées pour les signatures. Toutefois, il est intéressant de définir la pseudo similarité suivante, dite *fonction d'activation* $Act(A, B, r)$ entre deux termes A et B pour un type r. On note $A[B]$ la valeur de B dans la signature normée de A, et réciproquement $B[A]$ la valeur de A dans la signature normée de B. Ces valeurs sont comprises entre -1 et 1 (quelle que soit la norme utilisée). Nous définissons, dans un premier temps, la fonction d'activation restreinte à des signatures à valeurs dans $\mathbb{R}^+$:

$$\text{Act}_+ : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{N}^+ \quad \rightarrow \quad \mathbb{R}^+$$
$$\text{Act}_+(A, B, r) = max(A[B], B[A], sim(A, B))$$

où $sim(A, B)$ est la similarité entre les deux signatures normées. Cette fonction retourne la valeur d'activation (entre A et B) ayant la plus grande valeur si elle reste supérieure à la ressemblance entre les deux signatures (la similarité). L'ajout de la similarité est utile pour des signatures de cohyponymes ne s'activant pas mutuellement, mais partageant de nombreux termes. Nous avons, par exemple :

- Act(mouche, insecte, idée) = 0,77
- Act(mouche, cible, idée) = 0,20
- Act(mouche, grain de beauté, idée) = 0,77
- Act(mouche, abeille, idée) = 0,60

On peut étendre la définition de la fonction d'activation à des signatures à valeurs dans $\mathbb{R}$:

$$\text{Act} : \mathbb{R} \times \mathbb{R} \times \mathbb{N}^+ \quad \rightarrow \quad \mathbb{R}$$
$$\text{Act}(A, B, r) = \begin{cases} min(A[B], B[A]) & \text{si} \quad min(A[B], B[A]) < 0 \\ \text{Act}_+(A, B, r) & \text{sinon} \end{cases}$$

Cette fonction retourne la valeur d'activation (positive ou négative) ayant la plus grande valeur absolue en donnant priorité aux activations négatives (inhibition). Nous avons, par exemple :

- Act(oiseau, voler, idée) = 0,44
- Act(oiseau, voler>déplacement aérien, idée) = 0,.48
- Act(oiseau>animal, voler, idée) = 0,32
- Act(oiseau>animal, voler>déplacement aérien, idée) = 0,56
- Act(autruche, oiseau, idée) = 0,64
- Act(autruche, voler, idée) = -0,1
- Act(autruche, voler>déplacement aérien, idée) = -0,36

(Ces valeurs sont réellement celles calculées à partir de l'état du réseau au moment de l'écriture de ces lignes. Le lecteur intéressé peut se rendre à `http://www.lirmm.fr/jeuxdemots/intern_lexical_signature.php`*.)*

La fonction d'activation est une mesure de la spécificité qui existe entre deux termes (via leurs signatures lexicales). La polysémie a clairement un effet de diminution de l'activation normalement attendue entre deux termes.

Distances d'activation

De cette fonction d'activation, il est possible, si on le souhaite, de définir directement deux distances (où les propriétés de réflexivité, de symétrie, et d'inégalité triangulaire seront vérifiées).

$$D_{\text{Act}}(A, B, r) = \frac{1 - \text{Act}(A, B, r)}{2}$$
$$D_{A,\text{Act}}(A, B, r) = \arccos(\text{Act}(A, B, r))$$

La distance D_{Act} est la conversion directe en distance de la fonction d'activation, avec un domaine image égal à $[0; 1]$. La distance $D_{A,\text{Act}}$ est angulaire avec un domaine égal à $[0; \pi]$. Nous avons, par exemple :

- D_{Act}(oiseau, voler, idée) = 0,56
- D_{Act}(oiseau, voler>déplacement aérien, idée) = 0,52
- D_{Act}(oiseau>animal, voler, idée) = 0,68
- D_{Act}(oiseau>animal, voler>déplacement aérien, idée) = 0,44
- D_{Act}(autruche, oiseau, idée) = 0,36
- D_{Act}(autruche, voler, idée) = 1,1
- D_{Act}(autruche, voler>déplacement aérien, idée) = 1,36

1.4.2 Autres opérations

Les opérations d'addition, de produit terme à terme et de contextualisation (forte et faible), peuvent être appliquées aux signatures lexicales, avec la même interprétation que pour les vecteurs. La similarité peut aussi être utilisée, à condition de prendre soin des éventuelles valeurs négatives, soit en les annulant, soit en considérant leur valeur absolue.

Enfin, le voisinage d'une signature S peut aussi être construit par simple extension de la définition sur les vecteurs. La fonction de voisinage peut être un révélateur d'ambiguïté ou de malformation du contenu du vecteur V. En effet, intuitivement les n premiers termes de la signature doivent entretenir un rapport fort avec les p premiers voisins. Il est possible de déterminer n comme étant les termes ayant les activations les plus fortes et ayant une couverture raisonnable de la signature (par exemple 80%). De façon similaire, p peut être déterminé comme le nombre de termes dont les vecteurs sont à une distance angulaire raisonnable de V (par exemple inférieure à $\pi/4$).

1.4.3 Construction et applications

La construction d'une signature peut se faire pour un terme de façon triviale à partir d'un réseau lexical, par sommation des relations sortantes et entrantes du terme concerné. Les valeurs d'activation sont ensuite 2-normées. À supposer que le réseau lexical est de bonne qualité, ce mode de construction s'affranchit de tout bruit. La construction d'une signature lexicale pour un texte peut être réalisée par extraction de mots-clés (voir chapitre 4). La qualité de la signature dépendra de celle des processus d'extraction.

Dans [Bouklit & Lafourcade, 2006], les signatures lexicales sont calculées pour des documents du Web (des hypertextes) et propagées selon le graphe induit par les hyperliens. Deux types de signatures sont calculés, les signatures *entrantes* (des documents qui citent le document concerné) et *sortantes* (du document concerné combiné aux signatures entrantes des documents cités). Cette récursion croisée amène à un calcul itéré de deux signatures pour chaque document. Le calcul de la (dis)similarité entre les deux signatures d'un document pouvait être à la base de la détection de documents cités pour ce qu'ils ne sont pas (identification de contenus problématiques).

Conclusion du chapitre 1

Nous avons introduit trois types d'objet utilisables aussi bien comme représentations pour un lexique en sémantique lexicale que dans le cadre d'une analyse de texte : les *vecteurs*, les *réseaux*, et les *signatures*. Dans les trois cas, la qualité de la représentation dépend fortement du mode de construction et des données utilisées (corpus, lexiques, etc.). Le réseau lexical fournit la meilleure précision, mais la comparaison de deux objets n'est pas forcément aisée. Vecteurs et signatures permettent une comparaison efficace via le calcul de similarité, de distance d'activation ou de distance

angulaire. Les vecteurs ont une dimension fixe définie *a priori*, offrant une plus grande efficacité que les signatures en terme de calcul.

Pouvons-nous caractériser plus avant ces types d'objets et tenter une classification ? Il est possible de se fonder sur les caractéristiques suivantes portant sur la nature de la dimensionnalité, le statut des objets associés et le mode de typage :

– **Dimensionnalité - ouverte ou fermée** : les vecteurs sont de dimension finie, par contre les signatures lexicales et les voisins d'un terme d'un réseau, constituent des ensembles ouverts. Les vecteurs saltoniens sont de dimension finie (au moins en théorie) ainsi que ceux de LSA ou HAL.

– **Objets associés - concepts, termes ou aucun** : les vecteurs conceptuels associent à chaque composante un concept, en ce sens ils sont délexicalisés. Les signatures lexicales sont des ensembles de termes, tout comme les vecteurs saltoniens traditionnels sont des vecteurs de termes. Les vecteurs anonymes n'ont pas d'objets associés aux composantes, ils ne sont donc pas directement décodables (c'est le cas des vecteurs de LSA après réduction de dimension).

– **Typage - global ou local** : Il est possible d'associer un vecteur ou une signature à un type de relation, mais ce type est fixé globalement pour chacune des composantes du vecteur. Dans le cas d'un réseau lexical, par contre, chaque relation est typée localement. Une structure de réseau impose un typage local (sauf dans le cas dégénéré où le graphe ne dispose que d'un type de relation, mais dans ce cas, il est formellement équivalent à un ensemble de signatures).

	Dim	**Association**	**Typage**
vecteurs conceptuels	fermée	concepts	global
vecteurs saltoniens	ouvert	termes	-
vecteurs anonymes (émergence, LSA, HAL)	fermée	-	global
signatures lexicales	ouverte	termes	global
réseaux lexicaux	ouverte	termes	local

De cette typologie, nous entrevoyons qu'il pourrait exister des types d'objets dont nous n'avons pas parlé. Par exemple, nous pourrions penser à des *signatures conceptuelles* ou des *réseaux conceptuels* (qui seraient des objets différents des graphes conceptuels, ceci-dit). Toutefois, si nous supposons que l'ensemble des concepts est défini *a priori*, alors une *signature conceptuelle* est équivalente à un vecteur conceptuel (la dimension est fixe). Une dimension ouverte implique qu'il existe une association à un type d'objet pour chaque composante, il ne semble donc pas imaginable d'avoir des *réseaux anonymes* ou des *signatures anonymes*.

Articles adjoints au chapitre 1

M. Mangeot-Lerebours, G. Sérasset, et M. Lafourcade. *Construction collaborative d'une base lexicale multilingue - Le Projet Papillon.* TAL, Volume 44, 1/2, 2003, pages 151 à 176.

M. Lafourcade, V. Prince, D. Schwab. *Vecteurs conceptuels et structuration émergente de terminologies.* TAL, Volume 43, 1/2, 2002, pages 43 à 72.

Annexe : opérations sur les vecteurs

Ce qui suit uniformise et remplace ce qui a pu être écrit à propos des opérations sur les vecteurs dans les articles cités ou inclus.

Notations

L'écriture x_i désigne la i-ème composante du vecteur $X = (x_1, \ldots, x_n)$. $dim(X)$ est la dimension de X (son nombre de composantes). Afin d'alléger l'écriture, nous poserons dans la suite que $dim(X) = n$. Dans ce qui suit, nous notons $\mathcal{E}$, l'espace vectoriel sur lequel sont définis les vecteurs.

Une histoire de normes

Il existe plusieurs normes possibles pour des vecteurs, la forme la plus générale étant :

$$\|X\|_p = (\sum_{i=1}^{n} x_i^p)^{1/p} \tag{1.1}$$

Toutefois, seules les normes pour les valeurs de p suivantes nous intéressent ici :

$$p = 1 : \qquad \|X\|_1 = \sum_{i=1}^{n} |x_i| \tag{1.2}$$

Cette norme correspond à un déplacement à angle droit sur un damier.

$$p = 2 : \qquad \|X\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{1.3}$$

Il s'agit ici de la norme euclidienne, qui est celle habituellement utilisée pour mesurer la distance entre deux points de l'espace usuel.

$$\|X\|_\infty = \lim_{p \to +\infty} \|X\|_p = \max(|x_1|, \ldots, |x_n|) \tag{1.4}$$

Normer un vecteur consiste à diviser toutes ses composantes par la norme du vecteur. Toutefois, quelle norme choisir ? Si on souhaite que le vecteur ait une longueur unitaire, la norme euclidienne (norme-2) sera utilisée. Par contre si on souhaite que la somme des composantes soit égale à 1, afin d'utiliser le vecteur comme un vecteur de probabilités, la norme-1 sera utilisée (car avec la norme-1, la somme des composantes vaut 1 et donc chaque composante peut être assimilée à une probabilité). Enfin, si on veut comparer proportionnellement toutes les composantes du vecteur à la composante maximum, la norme infinie sera utilisée (en pratique, une approximation avec un p assez grand). Sauf mention contraire, dans ce qui suit, on supposera les vecteurs 2-normés.

Similarité et dissimilarité

Souvent utilisée en recherche documentaire, la mesure de *similarité* $sim(X, Y)$ s'exprime comme le produit scalaire des vecteurs X et Y divisé par le produit de leur norme. Nous supposons ici que les composantes des vecteurs sont toutes positives ou nulles.

$$sim(X, Y) = \cos(\widehat{X, Y}) = \frac{X \cdot Y}{\|X\| \times \|Y\|} \tag{1.5}$$

La *dissimilarité* est une mesure de ce qui est différent entre les deux vecteurs :

$$dissim(X, Y) = \sin(\widehat{X, Y}) \tag{1.6}$$

Nous avons la propriété suivante :

$$sim(X,Y)^2 + dissim(X,Y)^2 = 1 \tag{1.7}$$

La fonction cotangente peut être interprétée comme une fonction de poids d'un vecteur en fonction d'un autre.

$$\cot(X,Y) = \frac{sim(X,Y)}{dissim(X,Y)} \tag{1.8}$$

Cette fonction est particulièrement utile pour pondérer des termes en fonction d'un contexte.

Distance angulaire

Nous introduisons également la *distance angulaire*, notée D_A, dérivée de la mesure de similarité. Intuitivement, cette distance constitue une évaluation de la *proximité thématique*, c'est une mesure de l'angle entre les deux vecteurs. Ces vecteurs sont normalisés (l'espace vectoriel considéré est normé).

$$D_A(X,Y) = \arccos(sim(X,Y)) \tag{1.9}$$

La distance angulaire est une application $D_A : \mathcal{E} \times \mathcal{E} \longrightarrow \mathbb{R}$. $\mathcal{E}$ est l'espace vectoriel considéré.
– symétrie : $\forall X, Y \in \mathcal{E}, D_A(X,Y) = D_A(Y,X)$
– séparation : $\forall X, Y \in \mathcal{E}, D_A(X,Y) = 0 \Leftrightarrow X = Y$
– inégalité triangulaire : $\forall X, Y, Z \in \mathcal{E}, D_A(X,Z) \leq D_A(X,Y) + D_A(Y,Z)$
L'ensemble $\mathcal{E}$ est un espace métrique.

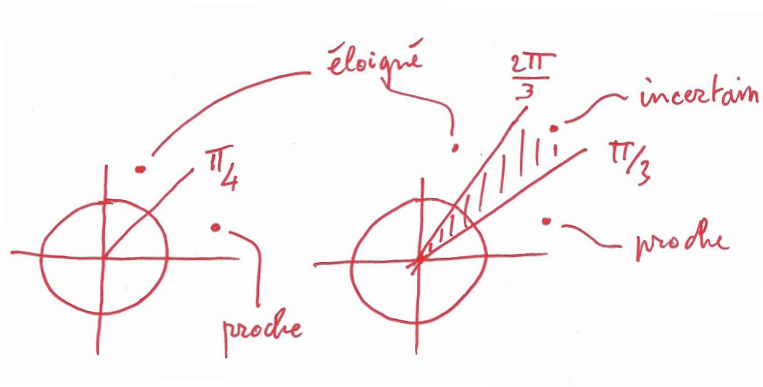


FIGURE 1.6 – Que veut dire que deux vecteurs sont proches ?

Décider dans l'absolu si deux vecteurs sont *proches* ou non est extrêmement subjectif, et c'est pourquoi, dans la mesure du possible, nous préfèrerons des comparaisons relatives (avec donc au moins trois vecteurs). Toutefois, nous pouvons adopter deux *postures*. La première consiste à dire que si deux vecteurs se ressemblent plus qu'ils ne sont différents (c'est-à-dire $\text{sim}(X,Y) > \text{dissim}(X,Y)$ ou encore $D_A(X,Y) \leq \pi/4$) alors ils seront proches. La seconde approche définit un *no-man's land* entre $\pi/3$ et $2\pi/3$ pour lequel on ne se prononcera pas. En-deçà de $\pi/3$, les vecteurs seront proches et au-delà de $2\pi/3$, ils seront éloignés. La seconde méthode est intéressante dans certains modes de calcul en ce qu'elle rend possible une hystérésis.

Somme vectorielle

Soient X et Y deux vecteurs, leur *somme vectorielle* Z est définie par :

$$\mathcal{E}^2 \to \mathcal{E} \ : Z = X + Y \quad | \quad z_i = x_i + y_i \tag{1.10}$$

où z_i (resp. x_i, y_i) représente la i-ème composante du vecteur Z (resp. X, Y).
Soient X et Y deux vecteurs, leur *somme vectorielle normée* Z est définie par :

$$\mathcal{E}^2 \to \mathcal{E} \ : Z = X \oplus Y \quad | \quad z_i = \frac{x_i + y_i}{\|X + Y\|_2} \tag{1.11}$$

L'opérateur $\oplus$ est idempotent et nous avons $X \oplus X = X$. Le vecteur nul $\vec{\mathbf{0}}$ est l'élément neutre de la somme vectorielle et, par définition, nous posons :

$$\vec{\mathbf{0}} \oplus \vec{\mathbf{0}} = \vec{\mathbf{0}}. \tag{1.12}$$

De ce qui précède, les propriétés de rapprochement (local et généralisé) peuvent être déduites :

$$D_A(X \oplus X, Y \oplus X) = D_A(X, Y \oplus X) \leq D_A(X, Y) \tag{1.13}$$

$$D_A(X \oplus Z, Y \oplus Z) \leq D_A(X, Y) \tag{1.14}$$

Soit $\{V_{(1)}, \ldots, V_{(p)}\}$ un ensemble de p vecteurs. On note $v_{(k)j}$ la j-ième composante du vecteur $V_{(k)}$, et on note v_j la j-ième composante du vecteur V. La somme vectorielle est généralisée à n'importe quel nombre de vecteurs par :

$$\mathcal{E}^p \to \mathcal{E} \ : V = \sum_{i=1}^{p} V_{(i)} \quad | \quad v_j = \sum_{k=1}^{p} v_{(k)j} \tag{1.15}$$

La somme vectorielle normée est généralisée à n'importe quel nombre de vecteurs par :

$$\mathcal{E}^p \to \mathcal{E} \ : V = \bigoplus_{i=1}^{p} V_{(i)} \quad | \quad v_j = \frac{\sum_{k=1}^{p} v_{(k)j}}{\| \sum_{i=1}^{p} V_{(i)} \|_2} \tag{1.16}$$

La somme vectorielle normée de deux vecteurs donne un vecteur équidistant en terme d'angle des deux premiers vecteurs. Il s'agit en fait d'une moyenne des vecteurs sommés. En tant qu'opération sur les vecteurs d'idées, la somme vectorielle normée peut être vue comme l'union des idées contenues dans les termes.

Il doit être souligné que si on souhaite conserver des proportions égales lors de l'addition de deux vecteurs, ceux-ci doivent être normés avec la norme euclidienne avant leur addition. La somme de deux vecteurs normés à l'aide des autres normes risque de produire des effets indésirables. En particulier, faire la somme de deux vecteurs de probabilités (normés 1).

Produit terme à terme

Soient X et Y deux vecteurs, leur *produit terme à terme* V est défini par :

$$\mathcal{E}^2 \to \mathcal{E} \ : Z = X \odot Y \quad | \quad z_i = x_i y_i \tag{1.17}$$

Soient X et Y deux vecteurs, leur *produit terme à terme normalisé* V est défini par :

$$\mathcal{E}^2 \to \mathcal{E} \ : Z = X \otimes Y \quad | \quad z_i = \sqrt{x_i y_i} \tag{1.18}$$

Cet opérateur est idempotent ($X \otimes X = X$) et $\vec{\mathbf{0}}$ est absorbant ($X \otimes \vec{\mathbf{0}} = \vec{\mathbf{0}}$). Il peut être généralisé à n'importe quel nombre de vecteurs par :

$$\mathcal{E}^p \to \mathcal{E} \ : V = \bigotimes_{i=1}^{p} V_{(k)} \quad | \quad v_j = \sqrt[p]{\prod_{k=1}^{p} v_{(k)j}} \tag{1.19}$$

L'opérateur $\otimes$ peut être interprété comme un opérateur d'intersection entre vecteurs. Si l'intersection entre deux vecteurs est le vecteur nul, alors ils n'ont rien en commun. On a, de plus, la propriété suivante :

$$\forall X \neq \vec{\mathbf{0}} \quad \forall Y \neq \vec{\mathbf{0}} \quad X \otimes Y = \vec{\mathbf{0}} \Leftrightarrow D_A(X,Y) = \frac{\pi}{2} \tag{1.20}$$

Comme nous venons de le dire, l'opérateur $\otimes$ peut être vu comme une intersection des vecteurs. Du point de vue des vecteurs d'idées, cette opération permet donc de sélectionner les idées communes à un ensemble de termes. Il est utilisé en particulier dans l'opération de contextualisation faible.

Contextualisation faible

Lorsque deux termes sont en présence, pour chacun d'eux, certaines idées se trouvent sélectionnées par le contexte que constitue l'autre terme. Ce phénomène de *contextualisation* consiste à augmenter chaque vecteur de ce qu'il a de commun avec l'autre. Comme nous venons de le voir, les idées communes à deux termes sont données par le produit terme à terme. Ainsi, nous pouvons définir la contextualisation faible $\gamma(X,Y)$ des vecteurs X par Y par :

$$\mathcal{E}^2 \to \mathcal{E} \ : \gamma(X,Y) = X \oplus (X \odot Y) \tag{1.21}$$

Cette fonction n'est pas symétrique. L'opérateur γ est idempotent ($\gamma(X,X) = X$) et le vecteur nul est un élément neutre ($\gamma(X,\vec{\mathbf{0}}) = X \oplus \vec{\mathbf{0}} = X$).

La propriété de *rapprochement* suivante peut être tirée :

$$D_A(\gamma(X,Y),\gamma(Y,X)) \leq D_A(\gamma(X,Y),Y) \leq D_A(X,Y) \tag{1.22}$$

$$D_A(\gamma(X,Y),\gamma(Y,X)) \leq D_A(X,\gamma(Y,X)) \leq D_A(X,Y) \tag{1.23}$$

La contextualisation faible $\gamma(X,Y)$ rapproche les vecteurs X de Y proportionnellement à leur intersection.

Coefficient de variation

La moyenne arithmétique d'un vecteur X $\mu(X)$ de dimension n est :

$$\mu(X) = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1.24}$$

La variance Var(V) et l'écart type $\sigma(X)$ sont donnés par les formules :

$$\text{Var}(X) = \frac{\sum_{i=1}^{n}(x_i - \mu(X))^2}{n} \quad et \quad \sigma(X) = \sqrt{\text{Var}(X)} \tag{1.25}$$

Enfin, le *coefficient de variation* c est donné par :

$$c = \frac{\sigma(X)}{\mu(X)} \tag{1.26}$$

Le coefficient de variation n'est défini que lorsque $\mu \neq 0$. Toutefois, il peut être arbitrairement étendu pour tenir compte du vecteur nul :

$$c(\vec{0}) = 0 \tag{1.27}$$

Dans le cadre des vecteurs d'idées, on peut voir le coefficient de variation c comme une mesure statistique normalisée (sans unité) de la *conceptualité* du vecteur V. Il est d'autant plus important que les composantes du vecteur sont contrastées, et vaut 0 si elles ont toutes la même valeur $\mu(X)$, soit $\frac{\sqrt{n}}{n}$ (n étant la taille du vecteur) si X est normalisé ($\|X\|_2 = 1$).

De façon identique, un vecteur plat non nul de taille n (vecteur unitaire $\vec{1}$ normé) aura donc une variance nulle si toutes les composantes valent $\frac{\sqrt{n}}{n}$. La variance est maximale si le vecteur est booléen (une composante à 1 et toutes les autres à 0). Plus précisément, nous avons comme variance et moyenne maximum :

$$\text{Var}_{\text{max}}(X) = \frac{1}{n} - \frac{1}{n^2} \qquad \mu_{\text{max}}(X) = \frac{1}{n} \tag{1.28}$$

Donc, nous avons pour le coefficient de variation maximum :

$$c_{max}(X) = \frac{\sqrt{\text{Var}_{\text{max}}(X)}}{\mu_{\text{max}}(X)} = \sqrt{n-1} \tag{1.29}$$

Puissance de vecteur

Soient X un vecteur et p un réel positif, la *mise à la puissance de* X par p est définie par :

$$\mathcal{E} \times \mathbb{R}^+ \to \mathcal{E} \ : Z = X^p \quad | \quad z_i = x_i^p \tag{1.30}$$

Le vecteur résultat est généralement normalisé. Cette opération est utile pour augmenter (ou diminuer) le contraste d'un vecteur, c'est-à-dire augmenter (ou diminuer) son coefficient de variation.

Construction collaborative d'une base lexicale multilingue

Le projet Papillon

Mathieu Mangeot-Lerebours* — **Gilles Sérasset**** — **Mathieu Lafourcade*****

* *National Institute of Informatics*
Hitotsubashi 2-1-2-1913 Chiyoda-ku Tokyo 101-8430 Japan
mangeot@nii.ac.jp

** *GETA-CLIPS, IMAG, Université Joseph Fourier*
BP 53, 38041 Grenoble cedex 9
Gilles.Serasset@imag.fr

*** *TAL-LIRMM, Université de Montpellier II*
161, rue Ada, 34392 Montpellier cedex 5
lafourcade@lirmm.fr

RÉSUMÉ. *Nous présentons le projet Papillon dédié la construction d'une base lexicale multilingue linguistiquement riche. Ce projet s'appuie sur le principe de construction collaborative, qui permet à chacun, professionel ou amateur, institution ou individu, de contribuer, dans la mesure de ses moyens, à ce grand chantier. Pour qu'un tel travail collaboratif puisse s'amorcer, il est nécessaire de fournir un ensemble conséquent d'informations lexicales multilingues, sur lesquels les contributeurs pourront s'appuyer. Après avoir présenté l'architectures linguistique, lexicale et informatique du projet Papillon, nous détaillons la méthode utilisée pour créer les informations initiales mises à disposition des contributeurs.*

ABSTRACT. *This paper presents the Papillon project dedicated to the building of a linguistically rich multilingual lexical database. This project is based on collaborative construction principle, which allows each one, professional or amateur, institution or individual, to contribute, with its own means, to this building task. For such a collaboratif work to be effective, it is necessary to provide a important set of multilingual lexical information, that will be the base of the contributors' work. After a presentation of the linguistic, lexical and software architectures of the Papillon project, we detail the method used to create the initial lexical information.*

MOTS-CLÉS : *Base lexicale multilingue, Dictionnaire, travail collaboratif.*

KEYWORDS: *Multilingual lexical database, dictionary, collaborative work.*

2e soumission à *Traitement Automatique des Langues*, le 30 mars 2003.

2 2e soumission à *Traitement Automatique des Langues.*

1. Introduction

Qu'elle soit implicite ou explicite, la connaissance linguistique reste un constituant fondamental des systèmes de traitement des langues. Le coût généralement constaté de création d'une connaissance lexicale explicite (un dictionnaire) est l'un des freins majeurs dans le développement d'un système de traitement des langues (TAL).

De la même manière, malgré le nombre et la diversité des dictionnaires à usage humain, il reste de nombreux trous à combler. Ainsi, un francophone ne peut actuellement trouver de dictionnaire bilingue français-japonais lui donnant une transcription utilisable des traductions en kanji (idéogrammes japonais) et lui fournissant des informations qui lui sont nécessaires (les spécificateurs numériques du japonais par exemple). Ces besoins sont encore plus flagrants pour des locuteurs de langues moins représentées au niveau lexical.

Dans cet article, nous présentons tout d'abord les motivations du projet Papillon dont l'objectif est de combler ce manque en construisant une base lexicale fortement multilingue offrant des informations linguistiquement riches. Les coûts de construction d'une telle base sont réduits par l'adoption d'une stratégie (présentée en 2.2) basée sur le modèle « open source » où les données disponibles se voient constamment enrichies par des contributions d'utilisateurs aux compétences diverses. Enfin, les coûts restants sont rendus acceptables par l'adoption d'une structure linguistique et lexicale (détaillées en 2.3) favorisant la réutilisation des données construites.

Nous décrivons ensuite l'implémentation du serveur de communauté au travers duquel se fait le travail de construction de cette base. Après avoir donné une vue d'ensemble du serveur Papillon (en 3.1), et présenté les principes de représentation des différentes structures de données manipulées (en 3.2), nous détaillons les méthodes utilisées pour offrir un service d'accès unifié aux diverses données disponibles sur le site (en 3.3).

La stratégie adoptée implique un travail initial de construction d'une amorce de base lexicale contenant un ensemble d'entrées initiales non détaillées, qui servira de base aux contributions des utilisateurs. L'architecture interlingue de la base rend cette construction relativement difficile. Nous présentons donc les outils (en 4.1) et méthodes (en 4.2) mises en œuvre pour cette étape d'amorçage.

2. Le projet Papillon

2.1. *Motivations du projet*

Le projet Papillon a été initié suite à différents constats :

– Il n'existe pas à l'heure actuelle de dictionnaire français-japonais électroniques et gratuits. De plus, les dictionnaires existants sont en général conçus pour les Japonais. La transcription des kanjis (idéogrammes japonais) est, dans la plupart des cas, omise. Les francophones ne peuvent donc pas se servir de ces dictionnaires à moins de

savoir lire le japonais. De plus, d'autres informations nécessaires pour s'exprimer en japonais font aussi défaut. Il existe par exemple, une grande variété de spécificateurs numériques en japonais. Certains échappent à toute logique. Il est donc indispensable que cette information soit accessible.

– Pour un francophone, il est beaucoup plus difficile d'obtenir des informations lexicales sur le malais ou le thaï que sur l'anglais.

Les besoins en données lexicales restent donc importants, non seulement pour un utilisateur humain, mais aussi pour les systèmes de traitements des langues, non seulement pour un francophone, mais pour tout utilisateur humain quelle que soit sa langue.

La principale difficulté réside dans les coûts prohibitifs de construction de grandes quantités de données. Par exemple, le projet Electronic Dictionary Research (EDR) de construction d'un dictionnaire japonais-anglais a nécessité plus de 1200 hommes années de travail. Son prix de vente, 14 000 €environ, est très inférieur aux coûts réels de construction qui ne seront probablement jamais rentabilisés. Il est cependant encore trop élevé pour un particulier. De ce fait, seules des institutions peuvent l'acquérir. De plus, les données fournies à ce prix sont utilisables principalement par certains systèmes de traduction automatique fondés sur des techniques particulières.

Le projet Papillon met en œuvre plusieurs stratégies pour réduire ces coûts et les rendre acceptables :

– En utilisant une structure lexicale suffisamment générale et complète pour que la plupart des applications du TAL y trouvent (de manière directe ou indirecte) les données dont elles ont besoin.

– En offrant des outils simples permettant à de non-spécialistes de partager leur connaissance naturelle de leur langue maternelle. La compétence des spécialistes étant utilisée afin de nettoyer et valider les informations ainsi obtenues.

– En construisant une base multilingue fondée sur une approche interlingue par acceptions, qui permet, en factorisant l'ensemble des connaissances bilingues disponibles, de s'appuyer sur les langues bien dotées pour avancer sur les langues moins représentées.

– Enfin, en appliquant le paradigme de construction « open-source » à la construction de données lexicales : chaque utilisateur contribue bénévolement à la base lexicale et les ressources sont ensuite disponibles gratuitement pour tous.

L'utilisation du paradigme « open-source » a déjà été utilisée dans des projets similaires de construction collaboratives de données lexicales sur le Web, parfois depuis plusieurs années. Le projet EDICT de construction de dictionnaire japonais-anglais dirigé par Jim Breen, professeur à l'université Monash en Australie, a démarré, il y a plus de 10 ans. De plus, des projets parallèles d'adaptation de ce dictionnaire à d'autres langues, comme le français conduit par Jean-Marc Desperrier ([DES 02]), ont démarré avec succès. D'autres projets de construction bilingue de dictionnaires incluant le japonais ont été lancés plus récemment comme SAIKAM, japonais-thaï et WaDoKuJiten, allemand-japonais.

4 2e soumission à *Traitement Automatique des Langues.*

C'est l'utilisation conjointe de l'ensemble des stratégies énoncées qui est novatrice. Nous pensons en effet que chacune des trois premières stratégies renforce l'impact du paradigme « open-source ». La première, en couvrant de nombreux besoins, permet d'impliquer des spécialistes du TAL qui apporteront leur pierre à l'édifice. La seconde, en permettant à des utilisateurs non-spécialistes de s'impliquer dans le projet, élargie le nombre de contributeurs potentiels. La troisième, en proposant une approche multilingue dès le début du projet, nous permet d'impliquer des partenaires de nombreux pays.

Lancé en 2000 par Emmanuel Planas, François Brown de Colstoun et Mutsuko Tomokiyo, le projet Papillon a été lancé en partenariat avec le National Institute of Informatics à Tokyo (Frédéric Andrès). Après trois séminaires (dont 2 à Tokyo et 1 à Grenoble), de nombreux partenaires se sont manifestés et ont souhaité rejoindre le projet : Jim Breen, auteur du dictionnaire EDICT (Université Monash, Australie), Francis Bond (NTT, Keihanna), Yves Lepage (ATR, Keihanna), Ulrich Appel, auteur du dictionnaire allemand-japonais WaDoKuJiten, Jean-Marc Desperrier, responsable de l'adaptation au français du dictionnaire EDICT, l'université Kasetsart et le NECTEC (Bangkok, Thaïlande), l'Universiti Sains Malaysia (Penang, Malaisie), les universités de Da Nang et de Hanoi (Vietnam), etc. Actuellement, les langues couvertes sont l'allemand, l'anglais, le français, le japonais, le lao, le malais, le thaï, le vietnamien et, très récemment, le chinois. Des contacts sont en cours concernant les langues indiennes.

2.2. *Stratégie de construction de la base lexicale multilingue*

Le succès du projet Papillon dépend de sa capacité à intégrer des informations fragmentaires de toutes natures dans un modèle unique. Ces informations peuvent provenir de dictionnaires existants ou d'utilisateurs contributeurs. Dans le premier cas, il s'agit d'informations cohérentes, disponibles dans un modèle propre, duquel nous extrayons les informations que nous souhaitons représenter dans le modèle de la base lexicale multilingue Papillon. Dans le second cas, il s'agit d'informations parcellaires exprimées dans le modèle de la base Papillon, sous forme de modification de données existantes.

Les contributions ne peuvent donc exister que s'il existe un ensemble minimal d'informations lexicales sur lesquelles les contributeurs apporterons des modifications (ajout, correction ou suppression). Il est donc primordial d'adopter une stratégie « en largeur » qui commence par une étape d'amorçage dont le but est d'obtenir automatiquement, à partir de dictionnaires existants, une première base lexicale contenant de nombreuses entrées associées à des informations minimales. Cette étape d'amorçage est complexe du fait de l'utilisation d'une architecture lexicale interlingue. Nous la détaillons dans la partie 4.

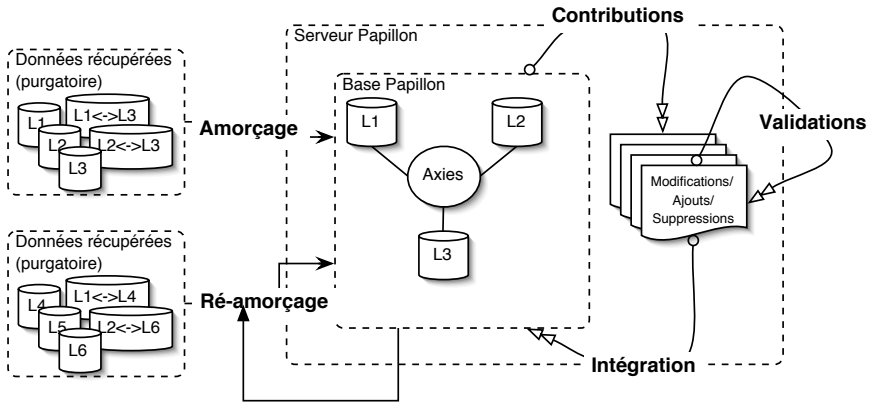


Figure 1. *Stratégie de construction de la base Papillon*

Une fois ce premier travail effectué, la base obtenue est installée sur le serveur Papillon. Les contributions sont effectuées sur cette base qui entre dans une phase de constante évolution.

Nous distinguons trois tâches dans l'évolution de cette base :

– **La contribution** où chaque utilisateur, spécialiste ou non, peut proposer des modifications (ajout d'informations dans des lexies existantes, ajout de lexies, suppression de lexies). Cette tâche pourra également être effectuée par des agents automatiques (acquisition de données à partir de corpus, etc.).

– **La validation** où les contributions seront soumises, directement ou indirectement à l'accord des utilisateurs, spécialistes ou non (des outils spécifiques seront développés afin de recueillir ces validations auprès d'utilisatrices non-spécialistes). Cette tâche pourra également être effectuée par des agents automatiques (confrontation à des corpus, ou à des ressources existantes, etc.).

– **L'intégration** où des utilisateurs de confiance acceptent ou rejètent les contributions (validées ou non) pour qu'elles soient effectivement appliquées à la base.

Cette méthode permet un certain contrôle sur la qualité de la base, mais pose un problème aux contributeurs qui ne souhaitent pas de délai entre le moment où ils contribuent et le moment où la contribution est prise en compte. Pour éviter ce délai, les modifications sont stockées dans l'espace personnel du contributeur et sont appliquées automatiquement à chaque fois qu'il consulte les entrées concernées. Ainsi les contributions sont instantanément visibles par le contributeur qui pourra aussi les partager avec d'autres utilisateurs.

6 2e soumission à *Traitement Automatique des Langues.*

2.3. *Architectures linguistique et lexicale*

2.3.1. *Macrostructure de la base lexicale*

La macrostructure de la base lexicale Papillon a été définie dans [SÉR 94b] et expérimentée à petite échelle pour la construction d'une petite base lexicale multilingue par [BLA 95].

Cette macrostructure est fondée sur la notion d'acceptions. Chaque dictionnaire monolingue est vu comme un ensemble d'acceptions d'une langue. Les liens entre les langues sont établis grâce à un dictionnaire pivot qui contient un ensemble d'acceptions interlingues (que nous appelons axies).

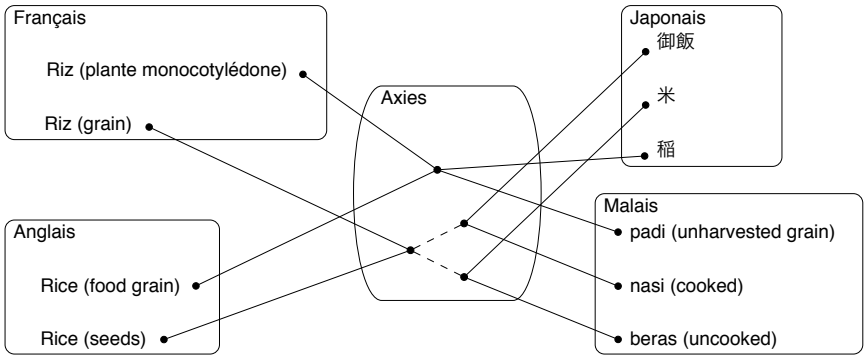


Figure 2. *Utilisation d'un lien inter axies pour représenter les phénomènes contrastifs de l'équivalence lexicale.*

Les vrais problèmes contrastifs de l'équivalence lexicale (qu'il ne faut pas confondre avec la polysémie monolingue, l'homonymie ou la synonymie comme [MEL 01] l'explique clairement) sont représentés grâce à un lien entre axies. Ce phénomène se retrouve dans l'exemple de la traduction du mot « Riz » dans 4 langues. Pour cet exemple, nous avons utilisé les acceptions définies dans des dictionnaires monolingues du commerce.

Chaque langue définit une acception correspondant au sens français désignant le riz comme une plante céréalière. Il est donc aisé de relier chacune de ces acceptions monolingues à une axie unique.

Par contre, ni le français, ni l'anglais ne définissent d'acceptions différentes suivant que le « riz » (considéré comme aliment) est cuit ou non alors que le japonais et le malais font cette distinction. Une axie ne peut être reliée à la fois aux acceptions malaises de « nasi » et « berak », à moins qu'on ne souhaite les considérer comme des synonymes (ce qui serait une erreur ici). Il faut donc créer 3 axies différentes pour relier ces acceptions monolingues : la première correspond aux acceptions de « nasi » et de « 御飯 » (gohan) ; la seconde correspond aux acceptions de « beras » et de

« 米 » (kome) et la dernière correspond aux acceptions de « riz » et de « rice ». Les traductions sont ensuite établies par l'ajout de deux liens entre la dernière axie et les deux premières.

Il faut noter que ce lien inter-axie ne représente que le fait que les deux acceptions reliées peuvent être utilisées comme traduction l'une de l'autre. Il ne porte pas de sémantique particulière et ne doit pas être confondu avec un lien ontologique.

2.3.2. *Microstructure des articles*

La structure de chacune des unités du lexique est issue de la théorie sens-texte ([MEL 84, MEL 88, MEL 92, MEL 96]). En effet, cette théorie étant indépendante des langues, elle nous permet de manipuler une structure unique pour toutes les langues de la base (à certaines nuances prêt, exposées plus bas).

Dans la théorie sens-texte, et dans le dictionnaire explicatif et combinatoire (DEC) qui en est sa composante lexicale, les articles sont nommés des lexies. Nous reprenons ici la définition d'une lexie de [POL 02] :

> Une lexie, aussi appelée unité lexicale, est un regroupement 1) de mots-formes ou 2) de constructions linguistiques qui ne se distinguent que par la flexion.
>
> Dans le premier cas, il s'agit de lexèmes et dans le second cas, de locutions.
>
> Chaque lexie (lexème ou locution) est associée à un sens donné. Que l'on retrouve dans le signifié de chacun des signes (mots-formes ou constructions linguistiques) auxquels elle correspond.

Les lexies sont ensuite regroupées en vocables. Nous reprenons ici la définition d'un vocable de [POL 02] :

> Un vocable est un regroupement de lexies qui sont associées aux mêmes signifiants et qui ont un lien sémantique évident.

Dans les dictionnaires monolingues de la base Papillon, nous reprenons la notion de lexie, qui constitue l'unité du lexique. Ces lexies correspondent à la notion d'acceptions monolingues évoquée dans le paragraphe précédent. Par contre, la notion de vocable n'est pas explicitement exprimée dans la base lexicale Papillon, cette notion se retrouvera au niveau de l'interface entre la base et ses utilisateurs.

Le DEC est actuellement constitué de 4 volumes regroupant 558 vocables en tout. C'est un dictionnaire expérimental avec une structure assez complexe et qui ne peut (encore) servir à un usage général. C'est pourquoi un projet de simplification du DEC (le projet DiCo, [POL 00]) a été lancé récemment par Alain Polguère et Igor Mel'čuk avec l'aide des étudiants de l'Observatoire de Linguistique Sens-Texte de l'université de Montréal au Canada.

Néanmoins, nous avons souhaité formaliser de manière plus détaillée certaines parties présentes implicitement dans la structure Dico.

Ainsi, dans l'exemple de la figure 3, nous représentons explicitement le fait que la fonction « Qsyn » (quasi synonymes) a trois valeurs, réparties dans deux groupes distincts : « assassinat » et « homicide » d'une part, « crime » d'autre part. Cette

Nom de l'unité lexicale	Meurtre.1
Propriétés grammaticales	nom, masc
Formule sémantique	action de tuer : PAR L'individu X DE L'individu Y
Régime	X = I = de N, A-poss Y = II = de N, A-poss
Fonctions lexicales	QSyn assassinat, homicide#1 ; crime V0 tuer A0 meurtrier-adj S1 auteur [de ART]//meurtrier-n /* Nom pour X*/
Exemples	La mésentente pourrait être le mobile du meurtre.
Idiomes	_appel au meurtre_, _crier au meurtre_

Figure 3. *Exemple d'une entrée du DiCo.*

répartition permet de noter qu'il existe une plus grande distance sémantique entre « meurtre » et « crime » qu'entre « meurtre » et « homicide ». Enfin, nous explicitons le lien qui est établi entre le sens 1 de « homicide » et cette lexie.

De plus, nous avons adopté un mécanisme qui nous permet d'adapter légèrement cette structure aux particularités des différentes langues de la base. Ainsi, les valeurs possibles comme propriétés grammaticales du thaï et du français sont différentes. Enfin, nous avons rajouté au dictionnaire du japonais les notions particulières de niveau de politesse, niveau d'usage et niveau de référence. Cette structure lexicale, exprimée en XML, est détaillée § 3.2.2.

3. Le serveur contributif Papillon

3.1. *Vue d'ensemble*

La construction, la gestion, la maintenance et une partie de l'exploitation de la base lexicale multilingue Papillon se fait par l'intermédiaire d'un site de communauté entièrement dynamique. Ce site a été construit entièrement en java, avec le serveur d'application « Enhydra » et la base de donnée « PostgreSQL ». L'ensemble des données utilisées dans ce site sont au format XML et sont exprimées en Unicode (UTF-8). Tous les outils utilisés sont des outils « open source ». Ce site est accessible à l'URL : `http ://www.papillon-dictionary.org/`.

3.1.1. *Services disponibles*

Outre les services inhérents à la création collaborative d'une base lexicale multilingue (interface de consultation/modification de la base Papillon, gestion des contributions, des utilisateurs, de l'historique, etc.), nous avons souhaité fournir 3 autres services pour rendre le site à la fois plus pratique et plus attractif :

– **Un service d'archivage de la liste de discussion des utilisateurs**. Ce service,

quasi systématique dans les sites de communauté, présente ici une particularité. En effet, les discussions se font dans différentes langues et les messages échangés arrivent dans des encodages divers (ISO-LATIN1, SJIS, EUC, UTF8, etc.). Ces encodages doivent être correctement reconnus et transcris en UTF8 afin d'être archives. Nous avons dû adapter les outils standard pour cela.

– **Un service de partage d'informations entre les utilisateurs**. Ce service permet à un utilisateur du site Papillon d'y ajouter des informations qu'il trouve pertinentes pour les autres utilisateurs. L'interface de ce service a été particulièrement soignée, afin que tout utilisateur, quel que soit son niveau en informatique, puisse obtenir un résultat très satisfaisant. L'utilisateur se contente de rédiger un document en HTML avec n'importe quel éditeur du commerce. Ce document peut contenir divers fichier HTML (avec des liens internes ou externes), des images ou d'autres données. Il se contente ensuite de transférer ce document sur le site (en utilisant son client http préféré). Ce document sera analysé, le code html sera corrigé automatiquement, et il sera intégré au site Papillon. Il prendra la même forme que les autres pages (mêmes entêtes, mêmes fonctionnalités, même comportement général). Ce service permet de plus de gérer des documents multilingues : le document est disponible dans plusieurs langues et les lecteurs verront automatiquement la version qui leur convient.

– **Un service d'accès unifié à de nombreux dictionnaires**. Ce service permet à tout utilisateur d'accéder, par une interface unique, à de nombreux dictionnaires monolingues et bilingues. Un utilisateur cherchant un mot dans une langue pourra obtenir les entrées correspondant à ce mot dans l'ensemble des dictionnaires disponibles. Avec ce service, nous espérons attirer des utilisateurs qui ne sont, dans un premier temps que « consommateurs » de données lexicales. Avec le temps, nous pensons que certains d'entre eux deviendront, à leur tour, « producteurs » de données.

3.1.2. *Organisation des données*

Les données lexicales disponibles sur le site Papillon sont diverses. Il peut s'agir de données de la base lexicale Papillon en cours de construction ou de données provenant de dictionnaires existants libres de droits ou qui nous ont été transmis par leurs auteurs. Ces données lexicales sont réparties dans 3 « zones » :

– *Les limbes* contiennent les données lexicales dans leur format propriétaire d'origine. Lorsqu'un dictionnaire nous est fourni, il est disponible dans cette zone en attendant d'être « récupéré ». Chaque dictionnaire est associé à un fichier de méta données contenant toutes les informations disponibles à son propos (son nom, les languages qu'il contient, la date de création, sa taille, ses auteurs, son domaine éventuel, etc.). Les dictionnaires présents dans cette zone peuvent être téléchargés tels-quels, mais les entrées qu'ils contiennent ne peuvent être obtenues individuellement.

– Après récupération, les données lexicales des limbes sont disponibles dans *le purgatoire*. Cette récupération consiste à définir en XML la structure du dictionnaire original (qui n'est pas modifiée). Les données sont ensuite transformées en XML en encodées en UTF8.
Les éléments de la structure XML obtenue (ou de la structure originale si le diction-

naire était déjà disponible en XML) sont ensuite identifiés grâce au mécanisme de pointeurs CDM détaillé § 3.3.2. Cette identification est stockée dans le fichier de méta données du dictionnaire. Elle permet d'accéder individuellement aux entrées de ce dictionnaire par des requêtes émises via l'interface unifiée disponible sur le site Papillon.

– *Le paradis* contient les entrées de la base lexicale multilingue Papillon. Ce dictionnaire est lui aussi accessible par l'interface unifiée disponible sur le site. Par contre, il est le seul dictionnaire qui puisse être modifié par l'interface d'édition.

3.1.3. *Implémentation*

L'application Papillon est organisée en trois couches (figure 4). La première couche prend en charge l'interface vers les utilisateurs. La seconde couche contient l'ensemble des services fournis aux utilisateurs. La dernière couche gère la persistance des données XML manipulées. Cette architecture rend facile l'ajout et la modification des interfaces de l'application vers ses clients (une interface WML est en cours de développement pour permettre l'accès via des téléphones portables). Elle permet aussi de s'abstraire de la manière dont les données sont stockées. Dans l'implémentation actuelle, les données XML sont stockées dans une base relationnelle « open source », via l'API java standard JDBC. Cette application peut être dupliquée sur de nombreux serveurs afin de répartir la charge.

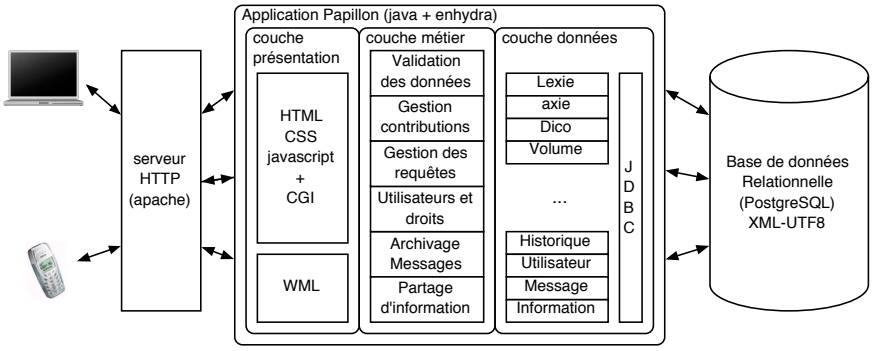


Figure 4. *Architecture du serveur de communauté Papillon.*

3.2. *Représentation des données de la base lexicale multilingue Papillon*

Dans le cadre de travaux antérieurs ([SÉR 94a, SÉR 94b] puis [MAN 01] et [MAN 02]), nous avons défini un cadre de représentation de données lexicales hétérogène. Pour le projet Papillon, nous utilisons ce cadre implémenté en XML (figure 5).

L'implémentation de notre microstructure s'appuie sur 3 niveaux de représentations : un cadre générique de représentation de dictionnaires (Dictionary Markup Lan-

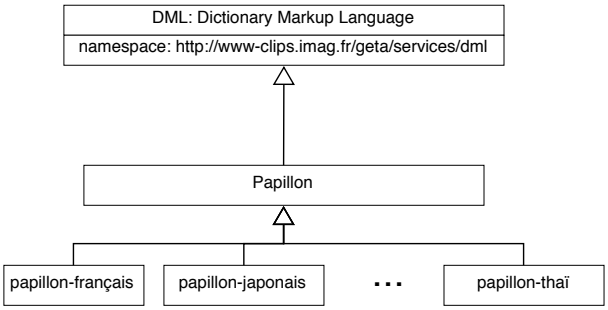


Figure 5. *Définition XML des structures de dictionnaires monolingues de la base Papillon.*

guage), un niveau de représentation de la structure commune à tous les dictionnaires de la base Papillon et un niveau de représentations représentant les spécificités de chaque langue.

L'ensemble des structures est implémenté en XML. Ces trois niveaux sont implémentés à l'aide de schémas XML qui incluent un mécanisme d'héritage simple.

3.2.1. *DML : description de dictionnaires en XML*

Le premier niveau de représentation est un cadre général, qui défini les concepts généraux, qui peuvent être utilisés pour représenter n'importe quel dictionnaire. Ce cadre est générique et permet la représentation aisée de dictionnaires hétérogènes.

Ce cadre est un espace de nom (namespace) XML nommé DML (Dictionary Markup Language, figure 6). Toute donnée d'une base lexicale peut être décrite en utilisant des éléments DML. Ce cadre défini non seulement les objets de base permettant de représenter des structures lexicales complexes (arbre, graphe, automate, etc.), mais aussi des types généraux utiles (booléen, date, langue, etc.) ainsi que les APIs permettant à des clients d'utiliser les données décrites ou à des fournisseurs de rajouter des services.

3.2.2. *Structure commune des dictionnaires monolingues Papillon*

La structure commune des dictionnaires monolingues de la base Papillon est définie par un schéma XML qui utilise l'espace de nom défini dans le DML. Ce schéma XML décrit la structure générale des lexies.

```
<element name="lexie">
  <complexType>
    <sequence>
      <element ref="d:headword" minOccurs="1" maxOccurs="1" />
      <element ref="d:writing" minOccurs="0" maxOccurs="1" />
      <element ref="d:reading" minOccurs="0" maxOccurs="1" />
```

12 2e soumission à *Traitement Automatique des Langues.*

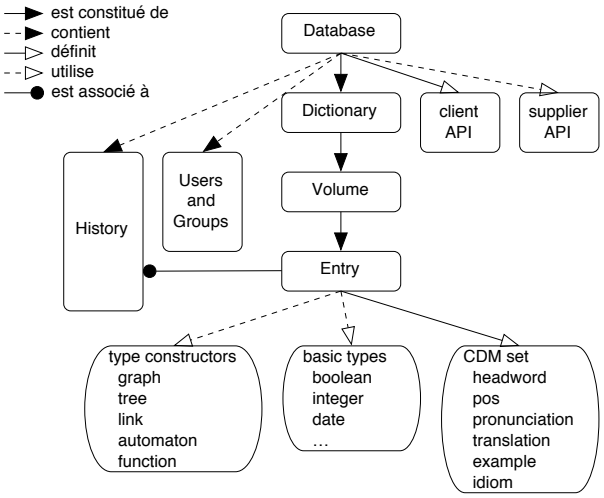


Figure 6. *Les concepts du DML.*

```
      <element ref="d:pronunciation" minOccurs="0" maxOccurs="1" />
      <element ref="d:pos" minOccurs="1" maxOccurs="1" />
      <element ref="d:language-levels" minOccurs="0" maxOccurs="1" />
      <element ref="d:semantic-formula" minOccurs="1" maxOccurs="1" />
      <element ref="d:government-pattern" minOccurs="0" maxOccurs="1" />
      <element ref="d:lexical-functions" minOccurs="0" maxOccurs="1" />
      <element ref="d:examples" minOccurs="0" maxOccurs="1" />
      <element ref="d:full-idioms" minOccurs="0" maxOccurs="1" />
      <element ref="d:more-info" minOccurs="0" maxOccurs="1" />
    </sequence>
    <attribute ref="d:id" use="required" />
  </complexType>
</element>
```

Chaque élément de cette structure est lui aussi défini à ce niveau. Ainsi, à ce niveau, l'élément « pos » (catégorie) est défini comme une chaîne de caractères, tandis que l'élément « lexical-functions » (les fonctions lexicales) est défini comme une liste de « function », qui est un type de base du DML.

```
<element name="pos" type="d:posType" />
<simpleType name="posType">
  <restriction base="string" />
</simpleType>
<element name="lexical-functions">
  <complexType>
    <sequence maxOccurs="unbounded">
```

```
      <element ref="d:function" />
    </sequence>
  </complexType>
</element>
```

3.2.3. *Adaptation de la structure à chaque langue de la base*

Le troisième niveau permet d'adapter légèrement la structure commune définie ci-dessus aux spécificités de chacune des langues de la base lexicale Papillon. Ainsi, chaque langue de la base possède un schéma qui lui est propre et qui redéfinit certains des éléments XML évoqués plus haut. Ainsi, l'élément « pos » (catégorie) du dictionnaire français est redéfini comme suit :

```
<simpleType name="posType">
  <restriction base="d:posType">
    <enumeration value="n.m." />
    <enumeration value="n.m. inv." />
    <enumeration value="n.m. pl." />
    <enumeration value="n.m., f." />
    <enumeration value="n.f." />
    <enumeration value="n.f. pl." />
    ...
  </restriction>
</simpleType>
```

De la même manière, une langue peut redéfinir les valeurs possibles pour les niveaux d'usage et de politesse. Il est souhaitable que chaque langue définisse de manière explicite une liste fermée de valeurs possibles pour ces éléments plutôt que de se fier à la structure générale qui, par défaut, accepte n'importe quelle chaîne de caractère. C'est en effet à partir de ces schémas XML que sont générées les interfaces de saisie.

3.3. *Accès unifié à des dictionnaires existants*

3.3.1. *Interface de consultation*

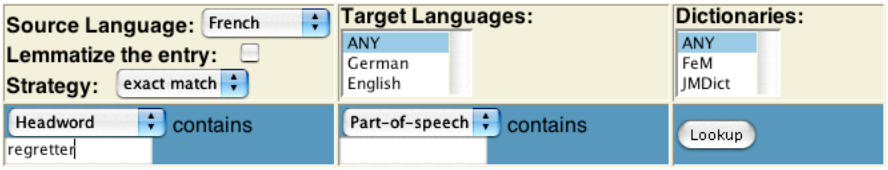


Figure 7. *Interface d'accès unifié aux dictionnaires du purgatoire et du paradis.*

L'interface unifiée permet de faire une recherche sur le lemme, sur la catégorie, sur la prononciation ou sur une traduction d'une entrée de dictionnaire (figure 7). Pour certaine langue, une lemmatisation est disponible pour permettre à l'utilisateur débutant d'entrer une occurrence quelconque. On peut de plus choisir les langues cibles et les dictionnaires dans lesquels se fera la recherche. Le service de lemmatisation est un service externe que nous accédons via Internet.

3.3.2. *Un mécanisme de pointeurs communs : CDM*

Cette interface de requête ne peut fonctionner que s'il existe un moyen d'identifier les éléments sur lesquels portent la recherche (entrée, catégorie, prononciation, traduction) dans des dictionnaires ayant des structures différentes. Cette identification est faite en établissant une correspondance entre un élément particulier d'un dictionnaire et un éléments définis par l'espace de nom DML (cf. § 3.2.1).

Le sous-ensemble du DML avec lesquels une telle correspondance peut être faite est nommé CDM (Common Dictionary Markup). Il est en constante évolution. La figure 8 en donne un exemple d'utilisation.

Element CDM	**équivalent TEI**	**FeM**	**OHD**	**NODE**
<entry>	(entry)	<fem-entry>	<se>	<se>
<headword>	(hom)(orth)	<entry>	<hw>	<hw>
<pronunciation>	(pron)	<french_pron>	<pr><ph>	<pr><ph>
<etymology>	(etym)			<etym>
<syntactic-sense>	(sense level="1")		<sense n=1>	<s1>
<pos>	(pos)(subc)	<french_cat>	<pos>	<ps>
<lexie>	(sense level="2")		<sense n=2>	<s2>
<indicator>	(usg)	<gloss>	<id>	
<label>	(lbl)	<label>		<la>
<example>	(def)	<french_sentence>	<ex>	<ex>
<definition>	(eg)			<df>
<translation>	(trans)(tr)	<english_equ> <malay_equ>		<tr>
<collocate>	(colloc)		<co>	
<link>	(xr)	<cross_ref_entry>	<xr>	<xg> <vg>
<note>	(note)		<ann>	

Figure 8. *Correspondance entre les éléments CDM et des éléments des dictionnaires TEI, FeM (français-anglais-malais), Oxford-Hachette (français-anglais) et Oxford (anglais)*

3.3.3. *Présentation des résultats*

Le résultat d'une requête est un ensemble d'unités codées en XML. Leurs structures sont variées car elles proviennent de dictionnaires différents.

regretter //r(e)gre-te-//

v.tr. regret souffrir du manque de miss se repentir << déplorer << s'excuser <<

regretter ,v.tr.

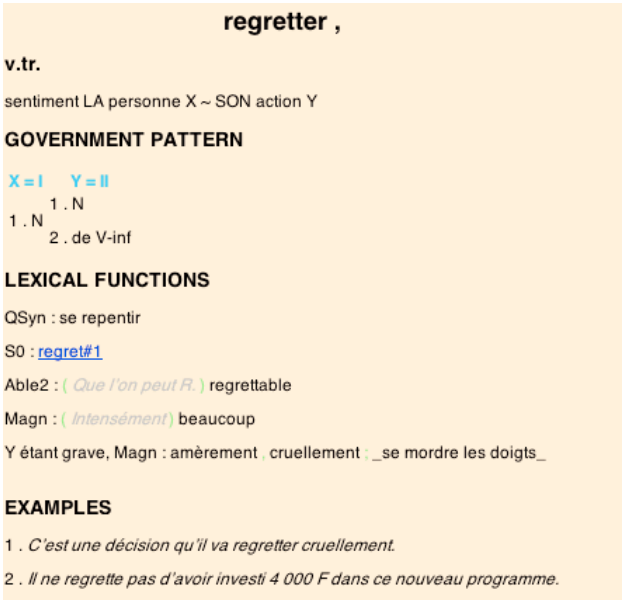
sentiment LA personne X ~ SON action Y QSyn : se repentirIS0 : regret#1 IAble2 : (*Que l'on peut R.*) regrettableIMagn : (*Intensément*) beaucoupIY étant grave, Magn : amèrement ,cruellement ,_se mordre les doigts_I*C'est une décision qu'il va regretter cruellement.Il ne regrette pas d'avoir investi 4 000 F dans ce nouveau programme.*

regretter

ngoại đ ộng từthương tiếc, luyến tiếc *Regretter un ami*thương tiếc một người bạn. hối tiếc; tiếc. *Regretter sa jeunesse*tiếc tuổi xuân *Regretter son argent*tiếc tiền *Regretter son imprévoyance*hối tiếc sự không lo xa của mình; *Regretter d'avoir mal agi*tiếc là đã hành động sai; *Je regrette de vous avoir fait attendre*tôi tiếc là đã để anh phải chờ.

Phản nghĩa Désirer, souhaiter. Se réjouir

Figure 9. *Trois résultats de la requête « regretter ». Le premier résultat provient du dictionnaire français-anglais-malais (FeM), le suivant de la base Papillon (entrée récupérée du DiCo), le dernier provient du dictionnaire français-vietnamien (vietDict).*



regretter ,

v.tr.

sentiment LA personne X ~ SON action Y

GOVERNMENT PATTERN

X = I Y = II

1 . N
1 . N
2 . de V-inf

LEXICAL FUNCTIONS

QSyn : se repentir

S0 : regret#1

Able2 : (*Que l'on peut R.*) regrettable

Magn : (*Intensément*) beaucoup

Y étant grave, Magn : amèrement , cruellement ; _se mordre les doigts_

EXAMPLES

1 . *C'est une décision qu'il va regretter cruellement.*

2 . *Il ne regrette pas d'avoir investi 4 000 F dans ce nouveau programme.*

Figure 10. *Forme inspirée du DEC pour la lexie « regretter.1 » du dictionnaire Papillon.*

Ces structures sont transformées en des « formes » qui dépendent des possibilités de l'interface utilisateur. Ces formes sont obtenues en appliquant des transformations XSL aux résultats de la requête. Ces transformations sont elles-mêmes des données du

serveur et plusieurs transformations peuvent être disponibles pour une même structure. Un utilisateur peut ajouter sa propre forme au serveur.

L'utilisateur peut facilement choisir la forme qu'il souhaite. Celle-ci peut mettre en valeur certaines informations lexicales ou en cacher d'autres. La figure 10 présente la lexie « regretter » (sentiment : personne X regrette son action Y) présentée selon une forme inspirée du DEC.

4. Construction d'une base lexicale initiale

Notre stratégie de construction commence la construction automatique d'une première base lexicale multilingue qui sert de base au travail contributif (l'amorçage). Cette construction se fait en deux étapes. Dans un premier temps, nous construisons les dictionnaires monolingues. Dans un second temps, nous construisons la base d'acceptions qui fera le lien entre les différents dictionnaires monolingues.

Ces deux étapes se font en combinant un certain nombre de ressources lexicales monolingues et bilingues. Un exemple de tels croisements est largement illustré par [QUA 01]. Cependant, l'architecture lexicale choisie nous impose d'établir les liens de traduction au niveau des lexies (acceptions monolingues), ce qui pose le problème de la sélection correcte des différents sens de termes polysémiques.

Pour pouvoir faire cette sélection de manière automatique, nous avons choisi d'utiliser le modèle des vecteurs conceptuels ([CHA 96], [LAF 99]) qui défini une notion de distance que nous interprétons comme une distance sémantique.

4.1. *Le modèle des vecteurs conceptuels*

4.1.1. *Définition et notion de distance*

Le modèle des vecteurs conceptuels a été présenté dans [LAF 02]. On associe à tout segment textuel (mot, syntagme, texte), une association thématique qui prend la forme d'une vecteur de concepts. Le jeux de concepts est prédéfini et constitue un espace générateur sur lequel les sens peuvent se projeter (la figure 11 donne une représentation graphique de deux de ces vecteurs). Par exemple, les sens de ‹*barrage*› peuvent être projetés sur les concepts suivant (les CONCEPT[*intensité*] étant ordonnés par intensité décroissante) : $V_{barrage} =$ (BARRIÈRE[*0.84*], OBSTACLE[*0.83*], ÉLECTRICITÉ[*0.82*], SPORT[*0.77*], FLOT[*0.76*], GUERRE[*0.76*], ...).

Dans ce modèle vectoriel, nous disposons des notions de *similarité* (utilisée habituellement en recherche d'information) et de *distance angulaire* $D_A(V_1, V_2)$. Cette dernière est une vrai mesure de distance (contrairement à la notion de similarité) et elle vérifie les propriétés de réflexivité, symétrie et inégalité triangulaire. Nous l'interprétons comme une évaluation de la *proximité thématique* entre sens de mots.

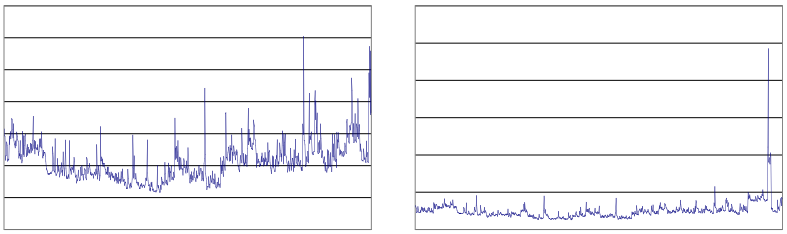


Figure 11. *Représentation graphique des vecteurs des termes* échange *(très polysémique) et* cession.

4.1.2. *Notion de contextualisation faible*

L'opération de contextualisation faible, notée $\Gamma(V_1, V_2)$ (elle aussi définie dans [LAF 02]), nous est aussi très utile dans le travail d'amorçage. En effet, les composantes communes à V_1 et V_2 seront fortement présentes dans $\Gamma(V_1, V_2)$. Cette notion permet d'amplifier les propriétés saillantes d'un vecteur dans un contexte donné.

Pour exploiter cette notion, nous associons à chaque vocable un vecteur égal à la somme normée des vecteurs de ses acceptions. En appliquant l'opération de contextualisation faible à un vecteur de vocable polysémique et à un vecteur contexte (de vocable polysémique ou non), nous obtenons un vecteur qui se rapprochera de l'un des sens du vocable considéré.

Par exemple, le terme *bank* est ambigu et son vecteur est globalement la moyenne entre les vecteurs des sens *river bank* et *money institution*. Si le vecteur de *bank* est contextualisé par celui de *river*, alors les concepts du champs sémantique lié à la finance seront considérablement inhibés.

4.2. *Construction de la base*

4.2.1. *Construction des dictionnaires monolingues*

En compilant les informations extraites de dictionnaires variés, (Hachette, Thésaurus Larousse, dictionnaire de synonymes de l'université de Caen, Wordnet, dictionnaire Oxford...) nous avons construit les lexies des dictionnaires monolingues français et anglais. Ces lexies, sont codées selon le schéma XML Papillon, mais contiennent très peu d'information (mot forme, catégorie et définition en langue naturelle).

Notre premier travail consiste à calculer le vecteur conceptuel associé à chacune de ces lexies. Le jeu de concept (les dimensions de l'espace) est prédéfini à l'aide des concepts présents dans le thésaurus Larousse.

Un indexage manuel de 5000 termes dans chaque langue, nous permet de connaître un premier ensemble de vecteurs. Ensuite, la définition de chaque lexie est analysée avec l'analyseur morphosyntaxique SYGMART. À partir des vecteurs des mots

connus de la définition, et de l'arbre d'analyse produit, nous calculons les vecteurs associés à chaque lexie et à chaque mot forme. Ce processus est itéré jusqu'à stabilité.

Nous disposons ainsi de dictionnaires monolingues vectorisés que nous définissons comme suit :

$$D_a = \{Lex_i\} \quad \text{et} \quad Lex_i = (w_i, \text{cat}_i, \text{def}_i, V_i) \tag{1}$$

Dans un dictionnaire monolingue vectorisé D_a, chaque vocable v correspond à n lexies $Lex_i = (v, \text{cat}_i, \text{def}_i, V_i)(n \geq 1)$. Si $n = 1$ le vocable v est strictement monosémique.

Nous donnons ci dessous un exemple des lexies du vocable *exiger* (dans cet exemple, les définitions sont issues du dictionnaire Hachette).

> **exiger.1** V, #s=1# Réclamer, en vertu d'un droit réel ou que l'on s'arroge. (Exiger le paiement de réparations) - (Exiger que (+subj)) (Il exige qu'on vienne), V_1
> **exiger.2** V, #s=2# Imposer comme obligation. (Allez-y, le devoir l'exige) (Les circonstances exigent que vous refusiez), V_2
> **exiger.3** V, Nécessiter. (Construction qui exige une main-d'oeuvre abondante), V_3

4.2.2. *Construction du dictionnaire interlingue d'acception*

Initialement, nous créons une acception interlingue (axie) pour chaque lexie des dictionnaires monolingues. Chaque acception est associée à un vecteur conceptuel. Initialement, ce vecteur est égal au vecteur de la lexie correspondante. La figure 12 présente l'état de la base lexicale après cette étape.

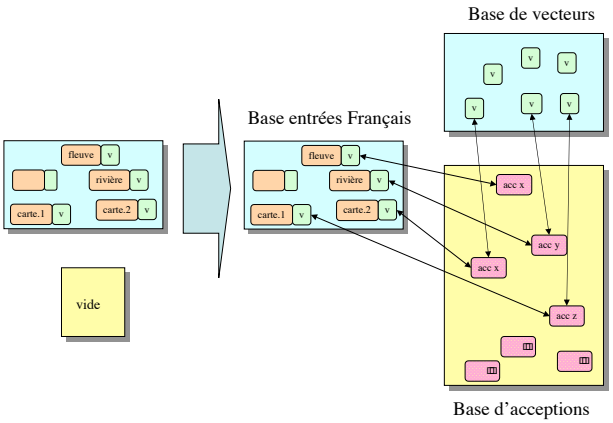


Figure 12. *Chargement initial du dictionnaire interlingue d'acceptions.*

Le travail qu'il reste à faire consiste à réduire cet ensemble d'acceptions à l'aide d'associations bilingues extraites de dictionnaire D_{a-b} (dictionnaires bilingues d'une langue source A vers une langue cible B) que nous définissons comme suit :

$$A(D_{a-b}, w) = \{Sa_i\} \quad \text{et} \quad Sa_i = (w, \text{cat}, \text{glose}^*, \text{equiv}^+)$$

Dans le dictionnaire D_{a-b}, le terme w est associée à n sous-entrée. Chaque sous-entrée contient : une information morphologique (au moins la catégorie morphosyntaxique, Nom, Verbe, Adjectif, Adverbe), zéro ou plus gloses, et au moins un équivalent dans la langue cible. Les gloses sont des termes optionnels qui permettent à l'utilisateur de sélectionner le sens dont il est question si le terme est polysémique. Ce sont ces même gloses qui permettent d'associer via les vecteurs conceptuels une sous-entrée du dictionnaire bilingue à une lexie du dictionnaire monolingue (en cas d'ambiguïté). Un exemple typique d'association bilingue (anglais-français) est :

demand ==
demand.1 VT, g{money, explanation, help}, e{exiger, réclamer})
demand.2 VT, g{higher pay}, e{revendiquer, réclamer})
demand.3 N, g{person}, e{demande})
demand.4 N, g{duty, problem, situation}, e{revendication, réclamation})
demand.5 N, g{for help, for money}, e{demande})

Dans le cas d'associations bilingues entre deux équivalents monosémiques (par exemple *babouin* $\rightarrow$ *baboon*) nous fusionnons les acceptions correspondantes. Un avertissement est émis pour le lexicographe si la distance entre les vecteurs des 2 acceptions est trop importante. Dans ce cas, il est vraisemblable, qu'au moins un des vecteurs conceptuels ne dispose pas d'activation pertinente.

Pour les autres associations bilingues, il nous faut identifier la lexie correspondant à chaque sous-entrée et la lexie correspondant à chaque équivalent.

Appariement lexie-association Pour chaque sous-association Sa_i, nous calculons un vecteur contexte V_C qui est la somme des vecteurs de ses gloses :

Le vecteur de la glose est le vecteur global, toutefois on sélectionne les sens dont les catégories morphosyntaxiques sont compatibles. Le vecteur associé à Sa_i est le calcul de la contextualisation faible (fonction Γ) entre le vecteur issu du dictionnaire monolingue pour w et le vecteur contexte. Le vecteur estimé $V_{\approx}$ de Sa_i est :

$$V_{\approx}(Sa_i) = \Gamma(V(w), V_C(Sa_i)) \tag{2}$$

À chaque sous-association Sa_i il est maintenant possible d'apparier un vecteur issu du dictionnaire monolingue. Il s'agit du vecteur (et donc du sens) qui est le plus proche du vecteur estimé (figure 13).

$$V(Sa_i) = Min(D_A(V(Se_j), V_{\approx}(Sa_i)) \tag{3}$$

Ainsi, nous avons apparié une partie des lexies et des associations bilingues. C'est-à-dire qu'en pratique nous avons pu fournir un vecteur aux associations bilingues ce

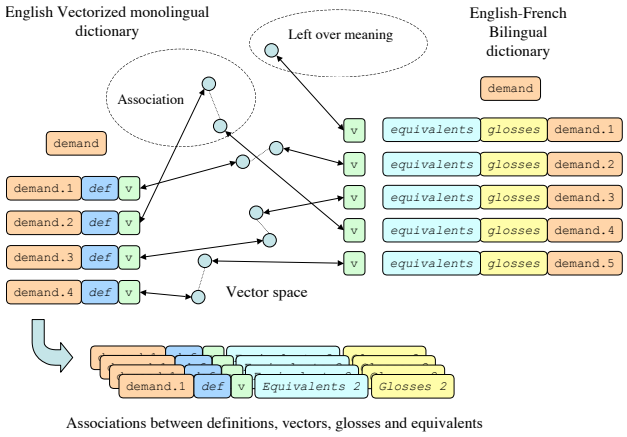


Figure 13. *Les associations sont appariées aux lexies dont le vecteur sémantique est le plus proche.*

qui est la condition pour les relier à des acceptions (dans les cas de polysémie). À la fin de ce processus, certaines des lexies du dictionnaire monolingue vectorisé disposent d'un lien unique vers une entrée du dictionnaire bilingue.

Liaison lexie-acceptions Il s'agit ici d'associer une lexie S_b du langage cible à une acception interlingue Ax_a issue d'une lexie du langage source.

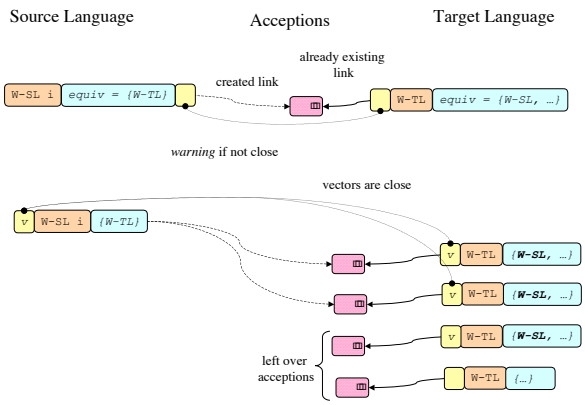


Figure 14. *Liaison Lexie-Acception dans le cas d'un équivalent monosémique (en haut) et d'un équivalent polysémique (en bas).*

On considèrera deux vecteurs conceptuels comme *suffisamment proche* si leur distance thématique est inférieur à un seuil t. Plus ce seuil est faible, plus le niveau de

confiance du lien vers l'acception est fort. En retour, il risque d'être difficile d'automatiquement réaliser l'association. Une valeur de seuil acceptable s'avère être $\pi/4$. Les différentes situations sont les suivantes :

1) **Un sens S vers un seul équivalent monosémique.** Ce cas consiste à sélectionner directement les termes. Les vecteurs conceptuels ne sont pas utilisés ici, si ce n'est pour effectuer une vérification. Si les deux vecteurs conceptuels ne sont pas raisonnablement proches, un message d'alerte est envoyé au lexicographe. Le problème peut aussi bien venir d'une erreur des dictionnaires bilingues, ou qu'un des vecteurs (ou les deux) à des activations inadéquates.

2) **Un sens S vers un équivalent polysémique.** Il faut alors sélectionner le sens équivalent S_b qui pourrait être acceptable (figure 14) Un filtre consiste à sélectionner les équivalents inverses, puis parmi les sens restant (s'il y en a plusieurs) choisir celui dont le vecteur est le plus proche.

3) **Un sens S vers plusieurs équivalent polysémique.** Ce cas est une généralisation des cas précédents.

4) **Cas d'erreur.** L'erreur principale provient de la constitution d'un ensemble vide. Cela peut arriver si les informations dans le dictionnaire bilingue sont inconsistantes. On remarquera que cela arrive relativement souvent en pratique.

4.2.3. *Nettoyage des liens*

L'architecture lexicale choisie impose des contraintes de bonne formation qu'il nous faut prendre en compte dans le peuplement du dictionnaire interlingue d'acceptions.

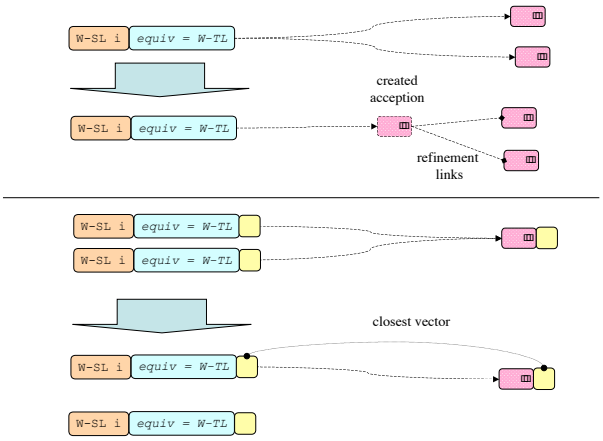


Figure 15. *Nettoyage de liens.* Partie supérieure *: Liens multiples. Une acception intermédiaire est créée ainsi que des liens inter-acceptions.* Partie inférieure *: Sens multiples. On conserve le lien minimisant la distance entre acception interlingue et lexie.*

1) Il y a, au plus, un lien possible d'une lexie vers une acception.

2) Deux lexies ne peuvent pas être liée à la même acception. Si ces sens sont synonymes, cette relation sera explicitée à l'aide d'une fonction lexicale (voir [SCH 01, SCH 02] pour une généralisation de l'approche à plusieurs fonctions lexicales caractéristiques).

Le nettoyage des liens consiste à détruire des liens qui auraient été créés abusivement (au vue des contraintes de bonne formation de la base interlingue) et d'en réévaluer certains. Deux cas peuvent se présenter : une lexie particulière est liée à plusieurs acceptions interlingues, ou une acception interlingue particulière est liée à plus d'une lexie dans une même langue.

1) **Liens multiples.** Pour résoudre ce problème, il est nécessaire de créer une acception intermédiaire. Le sens est alors lié seulement à la nouvelle acception (les liens précédents sont détruits). Des liens de raffinement de sens sont créés entre la nouvelle acception et les précédentes (figure 15).

2) **Sens multiples.** Nous avons à choisir parmi les sens lequel doit être retenu comme liable à l'acception, les autres liens étant détruits. La sélection se fait sur la base du plus proche vecteur.

En toute généralité, les deux situations peuvent survenir en même temps. Le processus de nettoyage est appliqué itérativement avec une priorité en faveur de la création d'acceptions intermédiaires.

4.3. *Premiers résultats et discussion*

Un certain nombre d'aspects doivent être mentionnés leur développement dépassant l'objet de cet article.

1) Vecteurs d'acceptions. Pour chaque acception, nous calculons son propre vecteur conceptuel. Ce vecteur est la moyenne des vecteurs des sens monolingues (dans l'ensemble des langues de la base) associés à l'acception. Ces vecteurs sont stockés comme s'ils constituaient un nouveau dictionnaire monolingue (cf. figure 14). Ils servent essentiellement à confirmer ou infirmer une proposition de liens lors de l'ajout de nouvelles entrées monolingues (à des acceptions déjà existantes).

2) Pondération des liens. Chaque lien créé automatiquement se voit attribuer une valeur de confiance (entre 0 et 1). Cette valeur correspond à la similarité entre le vecteur de l'acception et celui de l'entrée monolingue. Si un lien est confirmé par le lexicographe, le niveau de confiance vaut 1. L'exploitation des fonctions lexicales permet de moduler la valeur de confiance attribuée par le système ([SCH 01]).

3) Le processus de peuplement est effectué itérativement par des agents autonomes. Un agent explore la base d'acceptions et essaye d'évaluer les liens ou d'en créer. Par exemple, une acception pendante (avec un seul lien) doit être reliée à une entrée monolingue pour chacune des autres langues. Dans le cas d'entrée polysémique ou d'équivalent multiple, seule la source monolingue vectorisée nous concerne dans

la mesure où seuls les vecteurs conceptuels sont à la base du processus de décision. Les entrées orphelines doivent également être traité par la recherche d'une acception adéquate. Le processus est globalement convergent surtout dans la mesure où des liens sont fortement confirmés par les contributeurs humains.

Nos expériences de croisement de dictionnaires et de peuplement et liage automatique de la base d'acceptions nous ont permis dans le cas du français-anglais de générer environ 20000 acceptions dont environ 15000 était correctement liés. Le reste consistait en acceptions pendantes (soit du côté anglais soit du côté français). La plus grande difficulté concerne les entrées qui ne sont pas directement lexicalisées dans une langue. Dans ce cas, l'équivalent se réduit à une phrase explicative ou à une paraphrase. Ces traductions ne se retrouvent pas dans les dictionnaires monolingues de la langue cible. Par exemple le terme abêtir se traduit par *to make stupid, to turn into a moron* qui ne constitue pas des entrées du dictionnaire monolingue anglais. Afin de régler ce problème nous avons décidé de générer de telles entrées monolingues qui seront complétées par la suite (en particulier au niveau de leurs fonctions lexicales). Une petite frange d'acceptions (moins de 4%) et de sens (monolingue français ou anglais) sont incorrectement liés ou disposent de liens dont le seuil de confiance est inférieur à $1/2$. Il s'agit en général de termes très polysémiques (verbes support par exemple) qui génèrent beaucoup de forme lexicale voire de locutions dont la forme exacte peut être sujette à des variations. Cela engendre en particulier des amas d'acceptions lié par des raffinements de sens. En toute objectivité, l'approche fournit des résultats dans le rappel est important mais dont la précision est parfois médiocre. En l'occurrence, c'est très exactement la situation souhaitée où le lexicographe humain peut intervenir. Les termes polysémiques dont les champs sémantiques sont relativement distincts (par exemple un terme comme botte) sont correctement traités par les vecteurs conceptuels. Notre approche permet également d'améliorer la qualité des vecteurs conceptuels. Il s'agit ici d'une exploitation du graphe que représente les liens et les acceptions (indépendamment de sa construction). En particulier l'apport des lexicographes sur des informations lexicales permet d'augmenter la pertinence de certains vecteurs ce qui en retour améliore les performances du processus de peuplement.

5. Conclusion

Cet article présente un projet de construction d'une base lexicale multilingue linguistiquement riche. Par l'utilisation d'une stratégie basées sur le modèle « open source », nous souhaitons réduire les coût d'une telle construction en utilisant les compétences naturelles d'internautes volontaires. L'adoption d'une architecture lexicale interlingue qui sépare clairement les informations monolingues des informations interlingues présente de nombreux avantages dans ce cadre. D'une part, elle permet de distinguer les contributions interlingues des contributions monolingues qui requièrent des compétences différentes. D'autre part, elle permet de s'appuyer sur des langues bien dotées pour construire des données interlingues brutes impliquant des langues plus pauvres.

Le choix d'une architecture linguistique monolingue basée sur la composante lexicale de la théorie sens-texte d'Igor Mel'čuk, favorisera une réutilisation future de ces données dans de nombreuses et diverses applications de traitement des langues. De plus, la richesse des données construites rend plus intéressant le travail de contribution, les utilisateurs apportant de nombreuses informations originale, que l'on ne trouve actuellement dans aucun dictionnaire existant.

Le serveur de communauté sur lequel s'appuie le dictionnaire Papillon est en cours de construction, néanmoins, il est déjà fonctionnel et permet notamment un accès unifié à des dictionnaire existants. Ce service permet de rendre le site attractifs à des utilisateurs qui seront peut-être nos futurs contributeurs. Il nous a permis de plus de valider notre choix d'utiliser des outils standards (serveurs d'applications Java, XML pour la représentation des données, XSL pour leur manipulation, etc.). Ce serveur facilite aussi le travail des différents partenaires en proposant des services de partage de documents et d'archivage de liste de diffusion. Notre prochain objectif est d'ouvrir un service qui permettra des contributions en ligne par l'intermédiaire d'interfaces adaptables à l'utilisateur et à son environnement. À terme, nous souhaitons permettre des contributions en ligne (à partir d'un navigateur internet standard) et hors ligne (à partir d'application autonomes spécialisées). Le défi qui suivra consistera à animer une communauté de contributeurs et à trouver différentes motivations à même d'encourager la participation d'utilisateurs aux profils divers.

Une première réponse à ce défi réside dans la stratégie de construction employée, qui impose une phase d'amorçage assez complexe, mais néanmoins nécessaire pour disposer d'un ensemble de données suffisant qui sert à la fois de base de travail (les contributions sont vues comme des modifications de ces données) et de motivation (les données ainsi construites sont accessibles en ligne). Nos travaux préliminaires nous ont permis de produire une base d'acception sur le français et l'anglais par une méthode adaptable à d'autres langues avec un coût raisonnable. Il nous ont permis aussi d'associer un vecteur conceptuel à chaque acception créée. Ainsi, lors de la prise en compte de contributions, nous disposons de critères nous permettant de construire un agent automatique servant à la validation.

6. Bibliographie

[BAR 01] BARRIÈRE C., COPECK T., « Building Domain Knowledge from Specialized Texts », *TIA 2001*, Nancy, 2001.

[BLA 95] BLANC E., « Une maquette de base lexicale multilingue à pivot lexical : PARAX », *Lexicomatique et Dictionnairique, Actes du colloque LTT*, Universités Francophones, Actualités scientifiques, AUPELF-UREF, 1995, p. 43-58.

[BOU 93] BOURRIGAULT D., « Analyse locale pour le repèrage des termes complexes dans les textes », *TAL*, vol. 34, n° 2, 1993, p. 105-118.

[CHA 90] CHAUCHÉ J., « Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance », *TA Informations*, vol. 31, n° 1, 1990, p. 17-24.

[CHA 96] CHAUCHÉ J., SANDFORD E., « Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance », *Actes de MIDDIM-96*, Le Col de Porte, France, Août 1996, p. 56-66.

[CRU 95] CRUSE D. A., TOGIA P., « Towards a cognitive model of antonymy », *Lexicology*, vol. 1, 1995, p. 113-141.

[DEE 90] DEERWESTER S., DUMAIS S., LANDAUER T., FURNAS G., HARSHMAN R., « Indexing by latent semantic analysis », *Journal of the American Society of Information Science*, vol. 416, n° 6, 1990, p. 391-407.

[DES 02] DESPERRIER J.-M., « Analyzis of the results of a collaborative project for the creation of a Japanese-French dictionary », *Papillon'2002 Seminar*, NII, Tokyo, Japan, July 2002, http ://www.papillon-dictionary.org/ConsultInformations.po.

[FEL 95] FELLBAUM C., « Co-occurrence and antonymy », *International Journal of Lexicography*, vol. 8, 1995, p. 281-303.

[FIS 73] FISCHER W. L., *Äquivalenz und Toleranz Strukturen in der Linguistik zur Theory der Synonyma*, Max Hüber Verlag, München, 1973.

[GWE 87] GWEI G., FOXLEY E., « A Flexible Synonym Interface with application examples in CAL and Help Environments », *The Computer Journal*, vol. 30, n° 6, 1987, p. 551-557.

[HAM 01] HAMON T., NAZARENKO A., « La structuration de terminologie : une nécessaire coopération », *TIA 2001*, Nancy, 2001.

[HAT 01] HATHOUT N., « Analogies morpho-synonymiques. Une méthode d'acquisition automatique de liens morphologiques á partir d'un dictionnaire de synonymes », *TALN 2001*, Tours, July 2001, p. 223-232.

[HEA 98] HEARST M., « Automated discovery of Wordnet relations », FELLBAUM C., Ed., *Wordnet An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998, p. 131-151.

[JUS 91] JUSTESON J., KATZ S., « Co-occurrences of antonymous adjectives and their contexts », *Computational Linguistics*, vol. 17, 1991, p. 1-19.

[LAF 99] LAFOURCADE M., SANDFORD E., « Analyse et désambiguïsation lexicale par vecteurs sémantiques », *TALN'99*, Cargèse, juillet 1999, p. 351-356.

[LAF 01a] LAFOURCADE M., « Lexical sorting and lexical transfer by conceptual vectors », *First International Workshop on MultiMedia Annotation (MMA'2001)*, Tokyo, January 2001, page 6.

[LAF 01b] LAFOURCADE M., PRINCE V., « Synonymies et vecteurs conceptuels », *TALN 2001*, Tours, Juillet 2001, p. 233-242.

[LAF 02] LAFOURCADE M., PRINCE V., SCHWAB D., « Vecteurs conceptuels et structuration émergente de terminologies », *TAL*, vol. 43, n° 1, 2002, p. 43-72.

[MAN 01] MANGEOT-LEREBOURS M., « Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue », Thèse de nouveau doctorat, Spécialité Informatique, Université Joseph Fourier Grenoble I, Septembre 2001.

[MAN 02] MANGEOT-LEREBOURS M., « An XML Markup Language Framework for Lexical Databases Environments : the Dictionary Markup Language », *LREC Workshop on International Standards of Terminology and Language Resources Management*, Las Palmas, Islas Canarias, Spain, May 2002, p. 37-44.

[MEL 84] MEL'ČUK I., ARBATCHEWSKY-JUMARIE N., ELTNISKY L., IORDANSKAJA L., LESSARD A., *DEC : Dictionnaire explicatif et combinatoire du français contemporain,*

recherches lexico-sémantiques I, Presses de l'université de Montréal, Montréal(Quebec), Canada, 1984.

[MEL 88] MEL'ČUK I., ARBATCHEWSKY-JUMARIE N., DAGENAIS L., ELNITSKY L., IORDANSKAJA L., LEFEBVRE M.-N., MANTHA S., *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexico-sémantiques II*, Presses de l'université de Montréal, Montréal(Quebec), Canada, 1988.

[MEL 92] MEL'ČUK I., ARBATCHEWSKY-JUMARIE N., IORDANSKAJA L., MANTHA S., *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexico-sémantiques III*, Presses de l'université de Montréal, Montréal(Quebec), Canada, 1992.

[MEL 96] MEL'ČUK I., ARBATCHEWSKY-JUMARIE N., IORDANSKAJA L., MANTHA S., POLGUÈRE A., *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexico-sémantiques IV*, Presses de l'université de Montréal, Montréal(Quebec), Canada, 1996.

[MEL 01] MEL'ČUK I., WANNER L., « Towards a Lexicographic Approach to Lexical Transfer in Machine Translation (Illustrated by the GermanRussian Language Pair) », *Machine Translation*, vol. 16, n° 1, 2001, p. 21-87, Kluwer Academic Publishers.

[PLO 98] PLOUX S., VICTORRI B., « Construction d'espaces sémantiques á l'aide de dictionnaires de synonymes », *TAL*, vol. 39, n° 1, 1998, p. 161-182.

[POL 00] POLGUÈRE A., « Towards a theoretically-motivated general public dictionary of semantic derivations and collocations for French », *Proceeding of EURALEX'2000, Stuttgart*, 2000, p. 517-527.

[POL 02] POLGUÈRE A., « Notions de base en lexicologie », OLST-Département de linguistique et de traduction, Université de Montréal, 2002.

[QUA 01] QUAH C. K., BOND F., YAMAZAKI T., « Design and Construction of a machine-tractable Malay-English Lexicon », *Proceedings of AsiaLex*, Seoul, 2001, p. 200-205.

[RES 95] RESNIK P., « Using Information contents to evaluate semantic similarity in a taxonomy », *IJCAI-95*, 1995.

[RIL 95] RILOFF E., SHEPHERD J., « A corpus-based bootstrapping algorithm for Semi-Automated semantic lexicon construction », *Natural Language Engineering*, vol. 5, n° 2, 1995, p. 147-156.

[SAL 68] SALTON G., *Automatic Information Organisation and Retrieval*, McGraw-Hill, New York, 1968.

[SCH 01] SCHWAB D., « Vecteurs conceptuels et fonctions lexicales : application à l'antonymie », Mémoire de DEA Informatique, LIRMM, Montpellier, 2001.

[SCH 02] SCHWAB D., LAFOURCADE M., PRINCE V., « Amélioration de la représentation sémantique lexicale par les vecteurs conceptuels : le rôle de l'antonymie », *JADT 2002, vol. 2*, 2002, p. 701-712.

[SÉR 94a] SÉRASSET G., « Approche oecuménique au problème du codage des structures linguistiques », BLACHE P., Ed., *TALN-94 : Le traitement automatique du langage naturel en France aujourd'hui*, vol. 1, 7 to 8 April 1994, p. 109-118.

[SÉR 94b] SÉRASSET G., « Sublim : un système universel de bases lexicales multilingues et Nadia : sa spécialisation aux bases lexicales interlingues par acceptions », Thèse nouveau doctorat, Université Joseph Fourier-Grenoble 1, Décembre 1994.

[SPA 86] SPARCK JONES K., *Synonymy and Semantic Classification*, Edinburgh Information Technology Series, Edinburgh University Press, 1986.

[VER 01] VERLINDE S., SELVA T., « DAFA - Dictionnaire d'Apprentissage du Français des Affaires », http ://www.projetdafa.net, 2001.

[YAR 92] YAROWSKY D., « Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora », *COLING'92*, Nantes, 1992, p. 454-460.

Table des matières

Vecteurs conceptuels et structuration émergente de terminologies

Mathieu Lafourcade — Violaine Prince — Didier Schwab

LIRMM (CNRS - Université Montpellier 2)
161, rue Ada - F-34392 Montpellier Cedex 5
{lafourca,prince,schwab}@lirmm.fr

RÉSUMÉ. *Cet article présente les principaux avantages du modèle vectoriel pour la sémantique lexicale. Outre une représentation robuste, ce modèle permet l'émergence de relations entre termes, comme celles de synonymie et d'antonymie relatives. Nous décrivons le formalisme vectoriel utilisé, ainsi que les fonctions de base qui permettent de déterminer la notion de proximité thématique. L'extension de la méthode d'indexation d'un terme issu d'un document de spécialité se base en particulier sur une notion de pliage et de dépliage de vecteurs entre espaces vectoriels. Tout terme défini par d'autres termes qui n'appartiennent pas forcément à la terminologie, va imposer l'union des bases génératrices, et donner lieu à un dépliage du vecteur dans une base plus grande. Nous montrons comment la distribution lexicale, l'antonymie et la synonymie agissent comme des révélateurs de structure et réalisent une mise en relation transversale (non liée aux liens ontologiques) et instantanée dans l'espace vectoriel étendu. Les apports des expériences effectuées permettent de nous focaliser sur l'intérêt de ce type de méthode pour les terminologies.*

ABSTRACT. *This paper presents some advantages of the conceptual vector model for lexical semantics. Besides being a robust representation, this model allows the emergence of relations between terms, as relative synonymy and antonymy. We describe the underlying model, as well as the basic functions which allow to define the notion of thematic proximity. The extension of the indexation mechanism of a term extracted from a document of some speciality domain is mainly based on the notion of vector folding and unfolding between vector spaces. Any term defined thanks to other terms which may not belong to the specialty terminology, will impose the union of the vector space generative families, and lead to vector unfoldings toward a larger base. We show how the lexical distribution, the antonymy, and the synonymy, play as structure spotlight and transversally brigde terms across extended vector spaces. Some experiments focus us on the interest of this kind of approach for terminology.*

MOTS-CLÉS : *vecteurs conceptuels, structuration lexicale, synonymie relative, antonymie relative, extension ontologique.*

KEYWORDS: *conceptual vectors, lexical structuration, relative synonymy, relative antonymy, ontological expansion.*

44 TAL. Volume 43 - n° 1/2002

1. Introduction

La constitution automatique ou semi-automatique de bases terminologiques pour indexer des documents spécialisés dans un domaine donné est une tâche relativement ardue mais qui a déjà donné des fruits : elle a conduit les chercheurs à considérer la structuration de la terminologie comme une réponse au problème de son exploitation [Hamon et Nazarenko 2001]. Si la structuration permet, dans certains cas, de résoudre des questions relatives à la représentation, elle ne résout pas celui de la double appartenance d'un texte : un texte spécialisé comprend aussi bien des termes techniques que des termes généraux. De fait, cela impose l'usage d'un lexique général aussi bien que des lexiques terminologiques, quand il faut indexer ce texte, le classer, et surtout quand on veut l'utiliser comme base d'acquisition lexicale.

En outre, si la structuration permet de dériver des connaissances sur le lexique, elle n'a pas toujours un rôle de premier plan quand il est question d'**augmenter** ces lexiques terminologiques à partir de l'analyse de textes ou de définitions (parmi les travaux qui proposent une acquisition de ce type, on peut penser par exemple à [Barrière and Copeck 2001]). D'une part, ces textes contiennent forcément des termes généraux, ce qui nous ramène au problème précédent, et d'autre part, on s'aperçoit que la structure est un résultat local émergeant du calcul du sens, et non pas forcément une donnée stable fournie *a priori*. La polysémie due aux usages est en grande partie responsable de ce phénomène.

Notre objectif, en nous interressant à la terminologie a été : (**1**) d'explorer les potentialités du modèle que nous étudions (le modèle vectoriel) en termes de propositions de structuration de la terminologie ; (**2**) d'augmenter nos lexiques avec de l'information terminologique appropriée ; (**3**) d'être capable d'indexer et d'analyser thématiquement des textes spécialisés grâce aux structures émergentes ; (**4**) de proposer une application de (2) et (3) dans un domaine donné, ici l'économie, dans la mesure où nous pouvons bénéficier d'une ontologie spécialisée de type thésaurus, que nous savons vectoriser dans ce domaine. Dans cet article, nous montrons comment nous réalisons ces objectifs grâce à des mécanismes de structuration de la terminologie qui *émergent* de l'application de fonctions lexicales comme la synonymie et l'antonymie.

2. Problématique

L'exploitation, la plus automatique possible, d'un lexique terminologique pousse à se positionner sur deux points : le rôle du lexique général (vs. le lexique de spécialité) et la structuration de la terminologie pour indexer des textes susceptibles d'être spécialisés.

2.1. *Les rôles respectifs des lexiques*

On peut difficilement se passer de l'évocation d'un lexique général : l'économie de moyens que l'on pensait réaliser en constituant des lexiques spécialisés (versus un lexique général) est mise en échec. En réalité, le problème qui se pose est celui de la représentation : si on utilise une ontologie[1] pour rechercher des termes par appariement, alors la terminologie est un sous-ensemble de l'ontologie générale. En revanche, si on s'appuie, comme nous le faisons sur le modèle vectoriel, c'est au contraire la structure génératrice la plus petite connue qui sert de *base* au lexique général. Dans notre cas, les termes du lexique général (environ $65\,000$ entrées à ce jour) peuvent, de manière satisfaisante, se décliner en termes de composantes dans un espace vectoriel défini par une ontologie générale limitée à 873 concepts feuilles[2], et fondée sur le thésaurus Larousse [Larousse 2001]. La définition des vecteurs de concepts, qui constituent les briques élémentaires de la construction des vecteurs des différents sens de chaque terme, se fait à partir de l'organisation de ces même concepts en une ontologie.

Si 873 concepts servent de famille génératrice pour la représentation d'un nombre quelconque de termes (actuellement $65\,000$ mots), c'est que la *finesse* de représentation est ici relativement faible. Autrement dit, les termes[3] de spécialité ne peuvent se contenter d'une ontologie aussi peu précise, au risque d'être tous considérés comme des quasi-synonymes. Par exemple, les termes complexes tels que ‹*droit du travail*›, et ‹*économie de marché*› sont très proches dans leur description générale, puisqu'ils ont une composante très forte sur le concept ÉCONOMIE ET DROIT qui appartient à l'ensemble générateur du thésaurus. Le *maillage* du thésaurus général est trop large. Il faut donc proposer pour les termes de spécialité une possibilité de maillage fin. C'est pourquoi nous avons mené une expérience en utilisant une hiérarchie de concepts issue de l'OCDE [OCDE 1991], définissant environ $2\,000$ concepts feuilles sur la thématique économique. L'objectif que nous avons cherché à atteindre est **l'indexation lexicale de textes susceptibles d'être spécialisés**, avec un lexique général et des lexiques de spécialité (qu'on appelera terminologies), sur la base du modèle des vecteurs conceptuels. L'ontologie spécialisée choisie dans le domaine économique, à partir d'une arborescence fournie par des experts, offre un maillage beaucoup plus précis et décrit finement les termes de spécialité, en particulier les termes complexes (groupes nominaux pour la plupart) dont on sait qu'il est nécessaire de les repérer correctement [Bourrigault 1993] lorsque l'on vise une indexation fine.

---

1. Dans le sens communément admis actuellement qui est l'arborescence des notions fondamentales ou concepts d'un domaine. Si tant est que cette arborescence existe, elle est unique, et est acceptée comme un consensus d'expertise.
2. Pour le thésaurus Larousse, un concept est une expression ou un terme qui sert de notion fondamentale et à laquelle on fait référence pour classifier le vocabulaire.
3. Les termes sont des mots comme ‹*capitalisme*› ou des groupes nominaux comme ‹*droit des sociétés*›. Ils servent à exprimer des notions particulières et peuvent servir d'index.

46 TAL. Volume 43 - n° 1/2002

2.2. *La structuration de la terminologie*

De nombreuses recherches proposent d'utiliser des terminologies préstructurées. Compte tenu de l'aspect *associatif* du modèle vectoriel utilisé, nous avons adopté la démarche symétrique. Nous avons plutôt choisi de faire émerger des structures dans la terminologie, qui sont datées et dynamiques. En effet, dans un environnement terminologique très fortement évolutif comme celui que nous avons mis en place, les structures sémantiques sont sujettes à modification. On a alors intérêt à rechercher les *indicateurs de structure* plutôt qu'à réifier des structures prédites (comme des relations *a priori*).

Dans cet article, nous indiquerons les principaux avantages du modèle vectoriel pour la sémantique lexicale (section 3) : outre une représentation robuste, ce modèle permet de faire émerger des relations entre termes, comme les relations de synonymie relative [Lafourcade et Prince 2001] et d'antonymie relative [Schwab 2001]. Nous décrirons ensuite rapidement le formalisme vectoriel utilisé, ainsi que les fonctions de base qui permettent de déterminer la notion de proximité entre termes (section 4). Les relations émergentes sont formalisées dans la section 5. L'extension de la méthode d'indexation de tout terme issu d'un document de spécialité est proposée dans la section 6 : elle se base en particulier sur une notion de pliage et de dépliage de vecteurs entre espaces vectoriels. Tout terme t, défini par d'autres termes qui n'appartiennent pas forcément à la terminologie, va imposer l'union des bases génératrices, et donc donner lieu à un dépliage du vecteur dans une base plus grande. La section 7 montre comment la distribution lexicale, l'antonymie et la synonymie agissent comme des révélateurs de structure et réalisent une mise en relation transversale (non liée aux liens ontologiques) et instantanée dans l'espace vectoriel étendu. Nous concluerons enfin sur les apports de l'expérience effectuée, en nous focalisant sur l'intérêt de ce type de méthode pour enrichir les terminologies.

3. Modèle vectoriel pour la sémantique lexicale

Le modèle vectoriel n'est pas récent, puisqu'il a été au départ introduit par Salton en informatique documentaire [Salton 1968]. Sa réhabilitation dans les recherches en TALN est en revanche relativement récente, car elle a été essentiellement motivée par la mise à disposition des chercheurs de grandes bases de textes grâce au web en particulier, alors que précédemment, ces recherches passaient par des phases ardues de constitution de corpus d'expérience. L'approche que nous avons s'inspire de la version de 1983 du modèle vectoriel de Salton [Salton and MacGill 1983], mais elle en diffère en ce que nous faisons l'hypothèse qu'il existe un jeu de concepts prédéterminé qui peut jouer le rôle d'ensemble générateur et que ce jeu est celui défini par les lexicologues quand ils réalisent un thésaurus [Chauché 1990]. Les concepts de cet ensemble sont par définition interdépendants : la famille considérée n'est pas libre et ne constitue pas une base vectorielle proprement dite. Cette interdépendance est aussi attestée dans un modèle comme LSA [Deerwester *et al.* 1990] qui non seulement la reconnaît mais aussi l'exploite.

Le modèle vectoriel a été appliqué par Salton à l'indexation et à la recherche d'information textuelle en 1988 [Salton 1988]. Si ce dernier utilisait une analyse de surface par mots-clés pour alimenter ses vecteurs, notre démarche s'en distingue nettement : elle se base explicitement pour son calcul sur la géométrie et les variables morpho-syntaxiques des arbres d'analyse structurelle issus du texte. D'une façon générale, les documents sont traités indépendamment les uns des autres, alors que dans LSA, le traitement se fait de manière liée. De plus nous mettons l'accent sur la sélection lexicale en contexte (voir Bourrigault 1993 *op. cit.*) alors que des travaux comme celui de [Resnik 1995] font un usage exclusif de taxonomies.

3.1. *Avantages du modèle vectoriel pour la représentation du sens*

Le modèle de vecteurs conceptuels s'appuie sur la projection dans un modèle mathématique de la notion linguistique de champ sémantique. Tout terme (lexie) et tout concept est projetable sur les vecteurs de la famille génératrice, et est donc représenté par un vecteur *conceptuel*. Mieux encore, on peut calculer le thème de tout segment de texte tel que documents, paragraphes, syntagmes, etc. sous forme de vecteur conceptuel : c'est le *sens* du segment en question [Lafourcade et Sandford 1999]. Cette représentation homogène du sens, quelle que soit la granularité, est très avantageuse pour la classification des textes, l'indexation et la recherche évoluée d'information.

De plus, la représentation vectorielle ne fait aucune hypothèse *a priori* sur les relations conceptuelles. L'ontologie de départ mise à part, on ne se fonde sur aucune relation casuelle pour dériver du sens, et on n'inclut aucune contrainte sémantique. C'est un modèle purement calculatoire qui donne une *image* sémantique instantanée dans un état donné du dictionnaire conceptuel. Ce dernier est en apprentissage permanent, avec augmentation des définitions dès lors qu'une nouvelle source lexicologique électronique est disponible.

3.2. *Méthode d'acquisition lexicale*

Le principe du dictionnaire fondé sur le modèle des vecteurs conceptuels est celui de l'apprentissage de définitions et de concepts à partir de toute source lexicologique. Chaque définition de dictionnaire, expression en langage naturel fournie par des lexicologues, est analysée avec l'analyseur morphosyntaxique SYGMART[4], et un arbre d'analyse est produit. À partir de là ; des pondérations sont calculées et un vecteur conceptuel est produit pour représenter le sens donné par l'analyse de la définition. Ce vecteur entre alors dans le calcul du vecteur conceptuel du terme défini, ce qui fait que tout vecteur conceptuel (sauf ceux correspondant aux concepts de la famille génératrice) est modifié au fur et à mesure de l'analyse des définitions. Ce qui varie, c'est la valeur de la composante, et donc ce que nous appellerons par la suite *intensité*. L'avantage d'un tel système est qu'il peut acquérir non seulement de nouvelles lexies,

4. Développé par Jacques Chauché.

48 TAL. Volume 43 - n° 1/2002

mais aussi de nouveaux sens à des lexies données en fonction d'un réarrangement des intensités que prennent les concepts sollicités.

3.3. *Relations sémantiques induites*

Dans cet espace vectoriel conceptuel, on sait définir une notion de proximité sémantique en calculant une distance angulaire entre vecteurs (section 4.2). Cela signifie que l'on a une représentation de sens *proches*, sans pour autant valoriser correctement cette proximité. Le formalisme développé ci-après amène quelques remarques. On ne sait pas bien encore décliner cette proximité en relation d'hyperonymie ou d'hyponymie, qui sont caractéristiques des ontologies. En revanche, on arrive assez bien à mettre en valeur des relations transverses telles que la synonymie et l'antonymie et qui sont très utiles lorsqu'il s'agit justement de faire émerger une microstructuration. Les paragraphes suivants définissent les propriétés générales de ces fonctions lexicales telles que nous les avons expérimentées dans [Lafourcade et Prince 2001] (*op. cit.*) et [Schwab 2001] (*op. cit.*).

3.3.1. *Synonymie relative*

La synonymie est une *relation d'équivalence* permettant de substituer un terme (ou un segment) à un autre terme (ou segment), sans modifier le sens global de l'énoncé [Sparck Jones 1986]. En tant que relation lexicale, la synonymie n'a malheureusement pas les bonnes propriétés des relations mathématiques d'équivalence, simplement parce qu'à cause de la polysémie, la propriété de transitivité n'est pas vérifiée [Fischer 1973]. Il y a même des cas où la symétrie aussi n'est pas vérifiée (un hyperonyme peut être donné comme synonyme pour son hyponyme, mais pas l'inverse) [Cruse and Togia 1995].

Pour pouvoir néanmoins exploiter les propriétés d'équivalence qui sont fort utiles, nous avons défini une synonymie relative, c'est-à-dire une évaluation de la possibilité de substituer un terme (un segment, un concept...) à un autre, dans le contexte d'un troisième, pratique d'ores et déjà admise dans [Gwei and Foxley 1987]. Nous avons montré [Prince 1991] qu'il s'agissait d'une relation de pseudo-équivalence (en ce qu'elle est pseudo-transitive), et nous en avons proposé une démonstration dans [Lafourcade et Prince 2001] dans le cadre des vecteurs conceptuels, sur le lexique général. Nous en donnons la formalisation dans la prochaine section.

3.3.2. *Antonymie relative*

Habituellement, l'antonymie est définie comme une notion d'incompatibilité entre deux termes. À la lumière de la représentation vectorielle, nous préférons plutôt considérer une notion de *symétrie* qui se définit comme suit : *deux termes sont en relation d'antonymie si on peut exhiber une symétrie de leurs traits sémantiques par rapport à un axe.* La symétrie peut se décliner de différentes manières selon la nature de son support. On distingue, comme supports de symétrie :

– une **propriété** affectant une valeur étalonnable (valeur élevée, valeur faible) : par exemple, *chaud*, *froid* sont des valeurs symétriques de température, sur une échelle implicite.

– l'**application d'une propriété** (applicable/non applicable, présence/absence) : par exemple, *informe* est antonyme de tout ce qui a une forme, *insipide*, *incolore*, *inodore*, *etc.* de tout ce qui pourrait avoir saveur, couleur, odeur, ... [Justeson and Katz 1991].

– l'**existence d'une propriété** ou d'un **élément considéré comme symétrique par l'usage** (e.g. *soleil/lune*), ou par des **propriétés naturelles ou physiques des objets considérés** (e.g. *mâle/femelle, tête/pied* ...). Les antonymes vont alors par *paires* [Fellbaum 1995].

Notre idée est que les constructions d'antonymes sont dépendantes du type de support de symétrie. Il peut alors exister plusieurs types d'antonymes pour un même terme, comme il peut ne pas en exister d'évidents, si la symétrie n'est pas immédiatement décelable. En tant que fonction lexicale, comparée à la synonymie, on peut dire que si la synonymie est la recherche de la ressemblance avec comme test la substitution (*x est synonyme de y si x peut "remplacer" y*), l'antonymie est la recherche de la symétrie avec comme test la recherche du support de la symétrie (*x est antonyme de y s'il existe un support de symétrie t tel que x symétrique de y par rapport à t*). Par exemple, ‹*chaud*› est antonyme de ‹*froid*› car ‹*température*› offre un support de symétrie.

De même que pour la synonymie, l'antonymie s'apprécie toujours en contexte. Par exemple, ‹*frais*› peut être le contraire de ‹*tiède*›, ‹*chaud*›, ‹*racorni*›, ‹*flétri*›, ‹*maladif*›, ‹*rassis*›, ‹*confit*›, ‹*sec*›, ‹*surgelé*›, ‹*pourri*›, ... La prochaine section montre comment nous avons formalisé la représentation du sens en vecteurs conceptuels, les règles de composition des vecteurs, et les fonctions associées aux relations sémantiques de synonymie et d'antonymie décrites ci-dessus.

4. Le modèle des vecteurs conceptuels

4.1. *Principe*

Soit $\mathcal{C}$ un ensemble fini de n concepts. Un vecteur conceptuel V est une combinaison linéaire des éléments c_i de $\mathcal{C}$. Pour un sens A, le vecteur V_A est la description (en extension) des activations des concepts de $\mathcal{C}$. Par exemple, les sens de ‹*ranger*› et de ‹*couper*› peuvent être projetés sur les concepts suivant (les CONCEPT[*intensité*] étant ordonnés par intensité décroissante) :

V_{ranger} = (CHANGEMENT[*0.84*], VARIATION[*0.83*], ÉVOLUTION[*0.82*], ORDRE[*0.77*], SITUATION[*0.76*], STRUCTURE[*0.76*], RANG[*0.76*] ...).

V_{couper} = (JEU[*0.8*], LIQUIDE[*0.8*], CROIX[*0.79*], PARTIE[*0.78*] MÉLANGE[*0.78*] FRACTION[*0.75*] SUPPLICE[*0.75*] BLESSURE[*0.75*] BOISSON[*0.74*] ...).

50 TAL. Volume 43 - n° 1/2002

La description du processus d'apprentissage calculant les valeurs respectives des intensités pour chaque coordonnées d'un vecteur est exposé dans [Lafourcade 2001]. Il est clair, que pour des vecteurs denses (ayant très peu de coordonnées nulles), l'énumération des concepts activés est vite fastidieuse et surtout difficile à évaluer. On préférera en général procéder par sélection de termes thématiquement proches. Par exemple, les termes proches (et ordonnés par distance thématique décroissante) des mots ‹*ranger*› et ‹*couper*› sont :

‹***ranger***› : ‹*trier*›, ‹*cataloguer*›, ‹*sélectionner*›, ‹*classer*›, ‹*distribuer*›, ‹*grouper*›, ‹*ordonner*›, ‹*répartir*›, ‹*aligner*›, ‹*caser*›, ‹*arranger*›, ‹*nettoyer*›, ‹*distribuer*›, ‹*démêler*›, ‹*ajuster*› …	‹***couper***› : ‹*cisailler*›, ‹*émincer*›, ‹*scier*›, ‹*tronçonner*› , ‹*ébarber*›, ‹*entrecouper*›, ‹*baptiser*›, ‹*recouper*›, ‹*sectionner*›, ‹*bêcher*›, ‹*hongrer*›, ‹*essoriller*›, ‹*rogner*›, ‹*égorger*›, ‹*écimer*›, …

En pratique, plus $\mathcal{C}$ est grand, plus fines seront les descriptions de sens offertes par les vecteurs, mais plus leur manipulation informatique peut être lourde, surtout si l'on traite beaucoup de données. on rappelle que dans nos expérimentations sur le lexique général, $\dim(\mathcal{C}) = 873$, ce qui correspond au niveau 4 des concepts définis dans (Larrousse, *op. cit.*) La construction d'un lexique conceptuel (ensemble de triplets *(mot, variables morphologiques, vecteur)*) est réalisée automatiquement à partir de corpora (de définitions, de thésaurii, etc. (Lafourcade *op. cit.*)). Au moment de l'écriture de cet article, le corpus du français représente environ $210\,000$ définitions correspondants à $65\,000$ mots vedettes (pour $31\,000$ mots monosémiques et $34\,000$ mots polysémiques – pour ces derniers le nombre moyen de définitions, certaines éventuellement redondantes, étant de 4.61).

4.2. *Distance angulaire*

Il est souhaitable de pouvoir mesurer la proximité entre les sens représentés par deux vecteurs (et donc celle de leur mot associé). Soit $Sim(X, Y)$ la mesure de *similarité*, utilisée habituellement en recherche d'informations, entre deux vecteurs définie selon la formule (1) ci-dessous (avec "$\cdot$" étant le produit scalaire). On notera que l'on suppose ici que les composantes des vecteurs sont toujours positives ou nulles (ce qui n'est pas nécessairement le cas). Enfin, nous définissons une fonction de *distance angulaire* D_A entre deux vecteurs X et Y selon la formule (2).

$$Sim(X, Y) = \cos(X, Y) = \frac{X \cdot Y}{\|X\| \times \|Y\|} \quad (1)$$

$$D_A(X, Y) = \arccos(Sim(X, Y)) \quad (2)$$

Intuitivement, cette fonction constitue une évaluation de la *proximité thématique* et est en pratique la mesure de l'angle formé par les deux vecteurs. On considérera, en général, que pour une distance $D_A(X, Y) \leq \pi/4$ (soit environ $0,78$ radians ou encore 45 degrés), X et Y sont sémantiquement proches et partagent des concepts. Pour

$D_A(X,Y) \geq \pi/4$, la proximité sémantique de A et B sera considérée comme faible. Aux alentours de $\pi/2$ (soit environ $1,57$ radians ou 90 degrés), les sens sont sans rapport. La synonymie (dans son acception la plus large) est incluse dans la proximité thématique, cependant elle exige, de plus, la concordance des catégories morphosyntaxiques. L'inverse n'est évidemment pas vrai.

La distance angulaire est une vraie distance (contrairement à la mesure de similarité) et elle vérifie les propriétés de réflexivité (3), symétrie (4) et inégalité triangulaire (5) (qui peut jouer un rôle de pseudo-transitivité) :

$$D_A(X,X) = 0 \tag{3}$$

$$D_A(X,Y) = D_A(Y,X) \tag{4}$$

$$D_A(X,Y) + D_A(Y,Z) \geq D_A(X,Z) \tag{5}$$

Par définition, nous posons : $D_A(\vec{\mathbf{0}},\vec{\mathbf{0}}) = 0$ et $D_A(X,\vec{\mathbf{0}}) = \pi/2$ pour tout X avec $\vec{\mathbf{0}}$ dénotant le vecteur nul[5]. On considérera, en toute généralité, l'extension du domaine image de D_A à $[0,\pi]$ afin de comparer des vecteurs ayant des composantes négatives. Cette généralisation ne change pas les propriétés de D_A. On remarquera, de plus, que la distance angulaire est insensible à la norme des vecteurs (α et β étant des scalaires) :

$$D_A(\alpha X, \beta Y) = D_A(X,Y) \quad \text{avec} \quad \alpha\beta > 0 \tag{6}$$

$$D_A(\alpha X, \beta Y) = \pi - D_A(X,Y) \quad \text{avec} \quad \alpha\beta < 0 \tag{7}$$

Par exemple[6] dans le tableau qui suit, nous avons les distances angulaires (en radian) entre les vecteurs de plusieurs termes. Le tableau est symétrique (à cause de la symétrie de D_A) et la diagonale est toujours égale à 0 (à cause de la réflexivité de D_A). On remarquera qu'une valeur prend toute sa signification relativement à une autre. En particulier, il est satisfaisant d'avoir : (a) $d_1 \leq d_3$ et $d_2 \leq d_3$ ce qui correspond bien au fait que ‹*trier*› et ‹*ordonner*› d'une part, et ‹*trier*› et ‹*choisir*› sont "plus synonymes" que ‹*ordonner*› et ‹*choisir*›. On remarquera aussi que d_3 est supérieure à $\pi/4$, ce qui dénote un éloignement sémantique qui commence ; (b) d_4 est la plus petite valeur de $D_A(\text{ranger}, Y)$ car les concepts *CLASSER* et *RÉPARTIR* sont relativement proches, et de plus ‹*ranger*› est par ailleurs polysémique (*CLASSER*, *RASSEMBLER* et *NETTOYER*) et seul *CLASSER* est présent dans le tableau.

$D_A(X,Y)$	*trier*	*ranger*	*choisir*	*ordonner*	*ventiler*	*classer*	*répartir*
trier	0	0,517	0,662 d_1	0,611 d_2	0,551	0,441	0,462
ranger		0	0,829	0,6	0,523	0,409 d_4	0,444
choisir			0	0,848 d_3	0,77	0,796	0,758
ordonner				0	0,595	0,523	0,519
ventiler					0	0,471	0,391
classer						0	0,36
répartir							0

5. Le vecteur n'est sans doute pas représenté par un mot de la langue. Il s'agit d'une idée qui n'active... aucun concept ! C'est l'idée vide.
6. Tous les exemples de cet article sont issus de <http ://www.lirmm.fr/~lafourca>

52 TAL. Volume 43 - n° 1/2002

L'espace vectoriel conceptuel est muni de deux lois de composition interne : la somme (et son opération symétrique, la soustraction) et du produit terme à terme (on ne définit pas ici son opération symétrique) qui sont détaillées dans le prochain paragraphe.

4.3. *Opérateurs*

Somme vectorielle. Soit X et Y deux vecteurs, on définit V comme leur somme normée :

$$V = X \oplus Y \quad | \quad v_i = (x_i + y_i)/\|V\| \tag{8}$$

Cet opérateur est idempotent, autrement dit nous avons $X \oplus X = X$. Le vecteur nul $\vec{\mathbf{0}}$ est l'élément neutre de la somme vectorielle et nous avons $\vec{\mathbf{0}} \oplus \vec{\mathbf{0}} = \vec{\mathbf{0}}$ (par idempotence). De ce qui précède, nous déduisons (sans le démontrer) les propriétés de rapprochement (local et généralisé) :

$$D_A(X \oplus X, Y \oplus X) = D_A(X, Y \oplus X) \leq D_A(X, Y) \tag{9}$$

$$D_A(X \oplus Z, Y \oplus Z) \leq D_A(X, Y) \tag{10}$$

Soustraction vectorielle. Soit X et Y deux vecteurs distincts, on définit V comme leur soustraction normée :

$$V = X \ominus Y \quad | \quad v_i = (x_i - y_i)/\|V\| \tag{11}$$

Cet opérateur n'est pas idempotent et on aura par définition : $V = X \ominus X = \vec{\mathbf{0}}$. On remarquera que, dans le cas général, les valeurs v_i peuvent être négatives et que la fonction de distance a son image sur $[0, \pi]$.

Produit terme à terme normalisé. Soit X et Y deux vecteurs, on définit V comme leur produit terme à terme normalisé :

$$V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{12}$$

Cet opérateur est idempotent ($V = X \otimes X = X$) et $\vec{\mathbf{0}}$ est absorbant ($V = X \otimes \vec{\mathbf{0}} = \vec{\mathbf{0}}$). Cette opérateur n'est pas défini pour des vecteurs à composantes négatives.

Contextualisation faible. Lorsque deux termes sont en présence, pour chacun d'eux certains de leurs sens se trouvent sélectionnés par le contexte que constitue l'autre terme. Ce phénomène de *contextualisation* consiste à augmenter chaque sens de ce qu'il a de commun avec l'autre. Soit X et Y deux vecteurs, on définit $\Gamma(X, Y)$ comme la contextualisation de X par Y comme :

$$\Gamma(X, Y) = X \oplus (X \otimes Y) \tag{13}$$

Cette fonction n'est pas symétrique. L'opérateur Γ est idempotent ($\Gamma(X,X) = X$) et le vecteur nul est un élément neutre ($\Gamma(X,\vec{\mathbf{0}}) = X \oplus \vec{\mathbf{0}} = X$). On remarquera (sans les démontrer) que nous avons les propriétés dites de *rapprochement* suivantes :

$$D_A(\Gamma(X,Y),\Gamma(Y,X)) \leq \{D_A(X,\Gamma(Y,X)), D_A(\Gamma(X,Y),Y)\} \quad (14)$$

$$\{D_A(X,\Gamma(Y,X)), D_A(\Gamma(X,Y),Y)\} \leq D_A(X,Y) \quad (15)$$

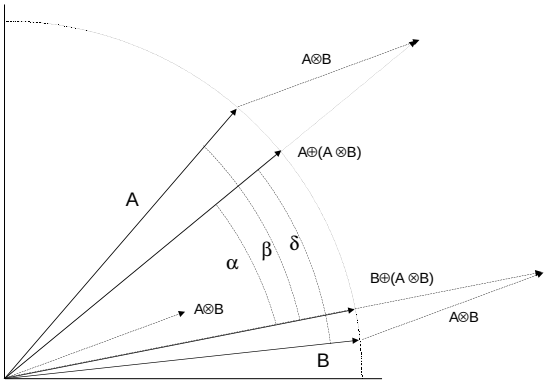


Figure 1. *Représentation géométrique en 2D de la contextualisation faible. L'angle α est la distance $D_A(\Gamma(A,B),\Gamma(B,A))$, β est la distance $D_A(A,\Gamma(B,A))$ et δ est la distance $D_A(\Gamma(A,B),B)$*

La contextualisation $\Gamma(X,Y)$ rapproche le vecteur X de Y proportionnellement à leur intersection. Dans la tableau qui suit, nous avons dans la partie supérieure les valeurs de $D_A(\Gamma(X,Y),\Gamma(Y,X))$.

$b\backslash a$	*trier*	*ranger*	*choisir*	*ordonner*	*ventiler*	*classer*	*répartir*
trier	0	0,269	0,363	0,322	0,288	0,228	0,239
ranger		0	0,474	0,316	0,273	0,211	0,23
choisir			0	0,485	0,434	0,451	0,425
ordonner				0	0,313	0,272	0,27
ventiler					0	0,244	0,201
classer						0	0,185
répartir							0

54 TAL. Volume 43 - n° 1/2002

5. Synonymie et antonymie relatives

5.1. *Fonction de synonymie relative*

Nous définissons la fonction de *synonymie relative* Syn_R entre trois vecteurs A, B et C, ce dernier jouant le rôle de pivot, comme suit :

$$\begin{aligned} Syn_R(A,B,C) &= D_A(\Gamma(A,C),\Gamma(B,C)) \\ &= D_A(A \oplus (A \otimes C), B \oplus (B \otimes C)) \end{aligned} \quad (16)$$

$$Syn_A(A,B) = Syn_R(A,B,A \oplus B) \quad (17)$$

La synonymie absolue Syn_A n'est qu'un cas particulier de la synonymie relative ou A et B constituent leur propre contexte. L'interprétation correspond bien à celle présentée ci-dessus, à savoir que l'on cherche à tester la proximité thématique de deux sens (A et B), chacun augmenté de ce qu'il a de commun avec un tiers (C).

5.1.1. *Propriétés*

Pour rendre compte des trois propriétés théoriques de la relation de synonymie relative (réflexivité, symétrie et pseudo-transitivité), nous les vérifions comme suit :

1. $Syn_R(A,A,C) = 0$ La réflexivité est héritée de celle de la distance angulaire D_A.
2. $Syn_R(A,B,C) = Syn_R(B,A,C)$ La symétrie pour les deux premiers arguments, provient également de celle de la distance angulaire.
3. $Syn_R(A,B,E) + Syn_R(B,C,E) \geq Syn_R(A,C,E)$ C'est un héritage de l'inégalité triangulaire de D_A. Elle représente la pseudo-transivité de la synonymie relative. Elle est en outre plus précise que la vérification de la propriété de transitivité : elle indique que la distance entre A et C/E est au pire égale à la somme des mesures de synonymie de A et B/E d'une part, et B et C/E d'autre part.
4. $Syn_R(A,B,0) = D_A(A \oplus \vec{\mathbf{0}}, B \oplus \vec{\mathbf{0}}) = D_A(A,B)$ Le vecteur nul $\vec{\mathbf{0}}$ ramène la synonymie relative à la distance angulaire.
5. $Syn_R(A,B,C) \leq D_A(A,B)$ Par héritage du rapprochement de D_A, quel que soit le point de vue, la synonymie relative ne peut que rapprocher A et B.

5.1.2. *Exemples*

Dans le tableau qui suit, nous avons dans la partie supérieure le rappel des valeurs de (a) $D_A(X,Y)$ et dans la partie inférieure les valeurs de (b) $Syn_R(X,Y,\textbf{\textit{trier}})$. On voit bien apparaître ici la mise en lumière de la polysémie. Nous avons, par exemple, $Syn_R(\textbf{\textit{classer}},\textbf{\textit{ranger}},\textbf{\textit{trier}})$ valant $0,283$ (soit environ 16°), ce qui indique une forte synonymie relative de ‹*classer*› et ‹*ranger*› par rapport à ‹*trier*›, chose que la distance angulaire correspondante ($0,409$, ou environ 23°) montrait aussi, mais avec

moins d'acuité. À l'inverse, $Syn_R(\textbf{\textit{choisir}}, \textbf{\textit{ordonner}}, \textbf{\textit{trier}})$ vaut $0,636$ (ou 36°), ce qui montre que ‹*choisir*› et ‹*ordonner*› s'*éloignent* l'un de l'autre par rapport à ‹*trier*›, alors qu'ils sont deux synonymes possibles de ‹*trier*›. La synonymie relative apparaît comme un bon indicateur de polysémie : ‹*choisir*› et ‹*ordonner*› relèvent majoritairement des deux "zones" sémantiques différentes. De plus, ‹*ordonner*› est lui-même polysémique.

$b\backslash a$	*trier*	*ranger*	*choisir*	*ordonner*	*ventiler*	*classer*	*répartir*
trier	0	0,517	0,662	0,611	0,551	0,441	0,462
ranger	**0,402**	0	0,829	0,6	0,523	0,409	0,444
choisir	**0,5**	**0,623**	0	0,848	0,77	0,796	0,758
ordonner	**0,478**	**0,43**	**0,636**	0	0,595	0,523	0,519
ventiler	**0,435**	**0,365**	**0,575**	**0,435**	0	0,471	0,391
classer	**0,369**	**0,283**	**0,607**	**0,385**	**0,344**	0	0,36
répartir	**0,376**	**0,309**	**0,57**	**0,383**	**0,272**	**0,268**	0

5.2. *Fonctions d'antonymie relative*

L'identification de plusieurs types d'antonymie (voir la section 2), implique l'existence de plusieurs fonctions d'antonymie. Toutefois, ces fonctions sont toutes basées sur une même méthode que nous explicitons ci-dessous.

5.2.1. *Principes et définitions*

La fonction $Anti_R$ de construction d'un vecteur antonyme V d'un vecteur A selon un vecteur contexte V_c, définie en termes linguistiques en 3.3.2, se note comme suit :

$$V = Anti_R(A, V_c)$$

Comme pour la synonymie, les diverses fonctions $Anti$ dépendent du contexte mais, contrairement à la synonymie, elles ne peuvent pas être indépendantes de l'organisation des concepts. Elles nécessitent d'identifier pour chaque concept et pour chaque contexte un vecteur qui sera considéré comme son opposé. Il faut donc construire une liste de triplets $\langle concept, contexte, vecteur\rangle$ appelé *listes d'antonymes*. Cette liste peut comprendre, par exemple, l'antonyme de EXISTENCE qui serait le vecteur V(INEXISTENCE) quel que soit le contexte. Elle peut contenir aussi l'antonyme d'AMOUR, qui serait, lui, constitué des vecteurs DÉSACCORD, AVERSION et INIMITIÉ. On remarquera, que le concept de HAINE n'existe pas dans l'ontologie utilisée (Larousse). Nous considérons que l'antonyme d'un terme qui ne possède pas d'antonyme(s) avéré(s) est ce terme lui-même (une discussion sur ce point est proposée dans [Schwab 2001] et [Schwab *et al.* 2002]). Il est important de noter que cette liste est différente pour chaque type d'antonymie. Il suffit donc de dresser autant de listes que de types d'antonymie examinés.

Fonction $AntiC$. La fonction $AntiC$ renvoie en fonction d'un concept c_i de $\mathcal{C}$ et d'un vecteur contexte V_c le vecteur considéré comme le vecteur antonyme dans une

56 TAL. Volume 43 - n° 1/2002

liste d'antonymes. Cette fonction se traduit donc par une simple exploration de la liste d'antonymes. On la note comme suit :

$$V = AntiC(c_i, V_c)$$

Par exemple, nous pouvons avoir :

$AntiC(\text{EXISTENCE}, V_c) = V(\text{INEXISTENCE}) \quad \forall V_c$
$AntiC(\text{AMOUR}, V_c) = V(\text{DÉSACCORD}) \oplus V(\text{AVERSION}) \oplus V(\text{INIMITIÉ}) \quad \forall V_c$
$AntiC(\text{DESTRUCTION}, V(\text{TRAVAUX PUBLICS})) = V(\text{CONSTRUCTION})$
$AntiC(\text{DESTRUCTION}, V(\text{ÉCOLOGIE})) = V(\text{PRÉSERVATION})$

5.2.2. *Construction du vecteur antonyme*

Définitions. Nous définissons les fonctions d'antonymie relative et absolue comme :

$$V = Anti_R(A, V_c)$$
$$V = Anti_A(A) = Anti_R(A, A)$$

Construction du vecteur conceptuel antonyme. Le but est, à partir de deux vecteurs conceptuels, un pour l'item lexical dont nous voulons l'antonyme, l'autre pour le contexte, de construire un vecteur opposé. L'idée est d'insister sur les notions saillantes des vecteurs A et V_c. Si ces notions peuvent être opposées, alors l'antonyme doit posséder les idées inverses dans la même proportion. Une fonction d'antonymie est définie comme suit :

$$Anti_R(A, V_c) = \bigoplus_{i=1}^{N} P_i \times AntiC(c_i, V_c)$$

avec comme définition pour le poids P_i :

$$P_i = A_i^{1+\text{CV}(A)} \times max(A_i, V_{c_i})$$

et A_i la ième composante de A :

$$A = \ < A_0, A_1, \ldots, A_{\dim(\mathcal{C})} >$$

Le poids P a été défini empiriquement à la suite d'expérimentations. Clairement, la fonction ne pouvait pas être symétrique, puisque le rôle de *vecteur à opposer* et celui de *vecteur contexte* ne sont pas interchangeables. Nous ne devons pas avoir $Anti_R$(V(‹*chaud*›),V(‹*température*›)) = $Anti_R$(V(‹*température*›),V(‹*chaud*›)). La puissance $1 + CV(V_{\text{item}})$ a donc été introduite pour insister d'avantage sur les idées présentes dans le vecteur que nous voulions opposer. Nous avons aussi remarqué que plus un vecteur était conceptuel (proche du vecteur d'un concept) plus il était intéressant d'augmenter cette puissance. C'est la raison pour laquelle cette puissance comprend le *coefficient de variation* [7] qui est un bon indice de la "conceptualité". Enfin, nous avons

7. Le coefficient de variation est donnée par la formule $\frac{EC(V)}{\mu(V)}$ avec EC(V) l'écart type du vecteur V et $\mu(V)$ la moyenne arithmétique des composantes de V.

introduit la fonction *max* afin de considérer les idées de l'item, même si celles-ci ne sont pas présentes dans le référent. Par exemple, si l'on cherche l'antonyme de ‹*froid*› dans le contexte de ‹*température*›, le poids de ‹*froid*› doit être important même s'il n'est pas présent dans le vecteur représentant ‹*température*›.

Une conséquence importante de notre définition de l'antonymie est que l'antonyme d'un item sans antonyme avéré est l'item lui-même. Celui-ci est alors considéré comme positionné sur l'axe de symétrie ([Schwab 2001] et [Schwab *et al.* 2002] *op. cit.*). Notre formalisation nous a permis de passer d'une fonction d'antonymie discrète (approche *linguistique* classique) à une fonction d'antonymie continuement définie sur l'espace des sens.

5.2.3. *Mesure d'évaluation de l'antonymie*

Il semble pertinent de savoir si deux items lexicaux peuvent être l'antonyme l'un de l'autre afin de posséder un outil comparable à la synonymie relative. Nous avons donc créé une mesure d'évaluation de l'antonymie. Soient les vecteurs A et B. La question est de savoir si on peut dire s'ils sont antonymes dans le contexte C. La distance d'antonymie M_{anti-R} est la mesure de l'angle formé par la somme des vecteurs A et B et la somme de leur opposés $Anti_{c_R}(A, C)$ et $Anti_{c_R}(B, C)$. Soient les mesures d'antonymie relative et absolue :

$$M_{anti-R}(A, B, C) = D_A(A \oplus B, Anti_R(A, C) \oplus Anti_R(B, C)) \quad (18)$$

$$M_{anti-A}(A, B) = D_A(A \oplus B, Anti_A(A) \oplus Anti_A(B)) \quad (19)$$

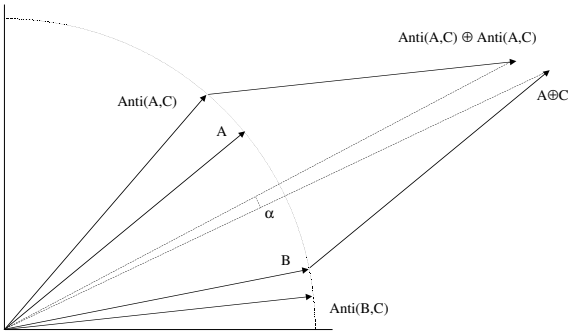


Figure 2. *Représentation géométrique en 2D de la mesure d'évaluation de l'antonymie par l'angle* α

La mesure d'antonymie n'est pas une distance. Ce n'est qu'une pseudo-distance. Elle vérifie les propriétés de réflexivité, symétrie et inégalité triangulaire uniquement dans le sous ensemble des items qui n'ont pas d'antonymes. Dans le cas général, elle

58 TAL. Volume 43 - n° 1/2002

ne vérifie pas la réflexivité. Les composantes des vecteurs conceptuels sont positives et nous avons la propriété : $Dist_{anti} \in [0, \frac{\pi}{2}]$. Plus la mesure est petite, plus les deux items lexicaux sont antonymes dans le contexte. En revanche, ce serait une erreur de considérer que deux antonymes seraient à une distance avoisinant $\pi/2$. Deux items lexicaux à $Mant = \pi/2$ l'un de l'autre n'ont aucune idée en commun[8]. Nous pouvons plutôt voir ici l'illustration que deux antonymes ont certaines idées en commun, celles qui ne sont pas opposables ou celles qui le sont mais dont l'activation est proche. Ils ne s'opposent que par certaines activations de concepts [Cruse and Togia 1995]. Une distance de $\pi/2$ entre deux items lexicaux devrait être plutôt interprétée comme le fait que ces deux items lexicaux n'ont que peu d'idées en commun, une sorte d'anti-synonymie. Ce résultat confirme le fait que l'antonymie n'est pas exactement l'inverse de la synonymie mais lui est très liée. L'antonyme d'un item ‹*m*› n'est pas un mot qui ne partage aucune idée avec ‹*m*› mais un item qui s'oppose à ‹*m*› sur certaines idées !

5.2.4. *Exemples*

Nous avons par exemple :

M_{anti-A}(*EXISTENCE*,*INEXISTENCE*)$= 0,03$
M_{anti-A}(‹*existence*›, ‹*automobile*›) $= 1,06$
M_{anti-A}(‹*existence*›,‹*inexistence*›) $= 0,44$
M_{anti-A}(*AUTOMOBILE*, *AUTOMOBILE*) $= 0,006$
M_{anti-A}(*EXISTENCE*, *AUTOMOBILE*) $= 1,45$
M_{anti-A}(‹*automobile*›, ‹*automobile*›) $= 0,407$

Les exemples ci-dessus illustrent bien ce que nous disions auparavant. Les concepts *EXISTENCE* et *INEXISTENCE* sont très fortement antonymes en antonymie complémentaire. L'effet de la polysémie explique que les items ‹*existence*› et ‹*inexistence*› soient moins antonymes que les concepts. En antonymie complémentaire, *AUTOMOBILE* est son propre antonyme. La mesure de l'antonymie entre *AUTOMOBILE* et *EXISTENCE* est un exemple de notre remarque précédente sur les vecteurs qui ne partagent que peu d'idées. Aux alentours de $\pi/2$, cette mesure se comporte comme la distance angulaire. D'ailleurs, nous avons $D_A(existence, automobile) = 1,464$ (soit un peu moins de $\pi/2$).

5.3. *Vecteurs conceptuels et passage à la terminologie*

Comme on a pu le voir, le modèle des vecteurs conceptuels permet non seulement de travailler sur la composition de sens, mais aussi peut faire émerger des relations sémantiques transverses correspondant aux fonctions lexicales de synonymie et d'antonymie. Ce que nous avons montré a été réalisé sur un lexique général fondé sur une ontologie de même type. Dans la prochaine section, nous allons montrer comment un tel dispositif permet d'exploiter et d'extraire et d'enrichir des terminologies spécifiques et donc de mieux assister le traitement de textes à fort caractère technique.

8. Ce cas de figure est purement théorique, il n'existe dans aucune langue deux items lexicaux qui ne partagent aucune idée.

6. Projection ontologique de vecteurs conceptuels

6.1. *Extensions ontologiques*

6.1.1. *Généralités*

On considérera en toute généralité deux ontologies G (pour générale) et S (pour spécialisée). La première (G) est universelle et est censée engendrer (par définition) tout les mots de la langue et couvre (de façon grossière) tous les champs sémantiques. La seconde (S) ne contient que les termes de sa spécialité et ne couvre (en détail) que les champs sémantiques de son (ou ses) domaines. Parmi les propriétés premières de G et de S : S a de fortes chances d'être *localement* beaucoup plus précise que G, et l'intersection entre G et S ne doit pas être nulle. La première propriété est nécessaire pour rendre S intéressante (on verra dans ce qui suit une formalisation de ces propriétés). Ce qui est présenté ensuite peut s'étendre à n ontologies de spécialités.

Pour demeurer dans le même paradigme que précédemment, on estime que G et S sont des familles génératrices d'espaces vectoriels. Dans la suite, on parlera de G comme de l'espace vectoriel défini par l'ontologie G (idem pour S). Tout vecteur d'un espace E n'est comparable qu'avec un autre vecteur de E : on comparera donc les vecteurs de G (respectivement S) entre eux. Sauf indication contraire, tout vecteur est normé.

Pour traiter un texte technique qui, comme nous l'avons dit, comprend aussi bien des termes techniques que des formulations générales, il importe de considérer l'espace généré par $G \cup S$, que l'on appellera par la suite GS.

6.1.2. *Notion de maillage de la description*

On remarquera que les termes de G sont inclus dans GS, et que plus l'ontologie S est spécialisée, plus le rapport *(nombre de termes* $\times$ *nombre moyen de sens)/ nombre de concepts* est faible. Cela provient du fait que la description est plus précise et donc que la *maille* descriptive est plus serrée. Pour le moment dans nos expériences pour G (Thésaurus Larousse) nous avons $65\,000$ entrées et environ 5 sens en moyenne par entrée. Compte tenu de la dimension de G (873), cela donne $65\,000 \times 5/873 = 372$. Pour les textes techniques nous avons actuellement repéré environ $10\,000$ lexies concernées (en analysant des définitions) et nos premières constatations font état d'environ 2 sens en moyenne par lexie de spécialité. Comme nous l'avons dit précédemment, les textes techniques sont mieux référencés sur GS que sur S seulement. La dimension de GS de l'ordre de 2873 : somme des dimensions de S, $2\,000$ (nombre de concepts dans l'ontologie de l'OCDE considérée), et G soit 873. Ce qui donne pour GS : $10\,000 \times 2/2873 = 6,96$ soit environ 7. On voit bien que la différence de taille de la maille est assez spectaculaire en G et GS car il y a un facteur 53. Evidemment, à la limite (si l'on dispose d'une ontologie, ou d'une union d'ontologies, aussi spécialisée que possible sans synonymie exacte) ce rapport devrait tendre vers 1 (chaque terme est associé à un concept). Cette limite est tout à fait illusoire quand on traite des textes généraux et n'a de sens que pour des textes spécialisés.

60 TAL. Volume 43 - n° 1/2002

6.1.3. *Quantité d'information*

En revanche, le produit des termes (qui représente la quantité d'information à stoker) reste dans les mêmes ordres de grandeur : pour G, on a : $65\,000 \times 5 \times 873 = 283\,725\,000$ (soit au moins 4 fois plus en taille physique, soit environ 1,2 Go) et pour GS (si on ne traite que les textes techniques) $10\,000 \times 2 \times 2873 = 57\,460\,000$ (soit au moins 4 fois plus en taille physique, soit environ 225 Mo).

On remarque que la taille du lexique sémantique (ensemble des sens) spécialisé est presque cinq fois plus petite que celle du lexique sémantique général, ce qui nous ramène à déplacer le problème de la taille de l'ontologie (plus petite traditionnellement si elle est technique, plus grande pour nous) vers celui de la taille du lexique sémantique, en d'autre termes, la *quantité d'information*.

6.1.4. *Commentaire sur l'exhaustivité d'une couverture*

Pour des raisons opérationnelles, il est clair que nous ne souhaitons pas représenter tous les mots de la langue par des vecteurs de l'espace vectoriel GS (le produit serait déraisonablement égal à $910\,000\,000$ soit environ 4 Go). Ce serait non seulement coûteux, mais de plus n'apporterait rien à la finesse d'analyse, la plupart des mots n'appartement pas aux champs sémantiques décrits par la très grande majorité des concepts (ceux-ci étant en grande partie issus de S en propre).

6.1.5. *Apprentissage*

Nous avons analysé, pour obtenir les vecteurs conceptuels correspondants, les termes de spécialité (ici l'économie) à partir de leurs définitions, dont celles issues du DAFA (Dictionnaire d'Apprentissage du Français des Affaires). Notre objectif était, au départ, de construire les lexies à partir de S, autant que faire se peut, et de ne basculer ensuite sur GS que si cela était nécessaire. Nous nous sommes aperçus très vite que, en phase d'apprentissage, et surtout à partir de dictionnaires, nous avions presque systématiquement accès à des termes généraux (hors de S).

Par exemple, la première définition du terme ‹*marché*› est : *Lieu physique ou virtuel d'échanges*. Au mieux, seul le terme ‹*échange*› pourrait se projeter sur S. Ce qui rendrait les termes ‹*échange*› et ‹*marché*› (dans son sens 1) synonymes ! Il est donc nécessaire de tenir compte des vecteurs de ‹*lieu*›, ‹*physique*› et ‹*virtuel*› qui ne sont pourtant définis que dans G. C'est pourquoi nous traitons essentiellement des vecteurs dans GS et que nous avons défini une opération de ‹*dépliage*› (qui déploie un vecteur d'un sous-espace G ou S dans GS) afin d'obtenir un vecteur $D(v)$ de GS à partir de v de G. Ce vecteur ne porte pas plus d'information que v, mais rend possible le calcul dans GS.

On ne sait jamais si, dans une définition, une occurence d'un terme fait référence à un sens général ou spécifique. C'est pourquoi, souvent en pratique l'apprentissage s'amorce avec une combinaison des deux possibilités. Les sens probables émergent par activation des informations mutuelles des occurences des autres termes de la définition.

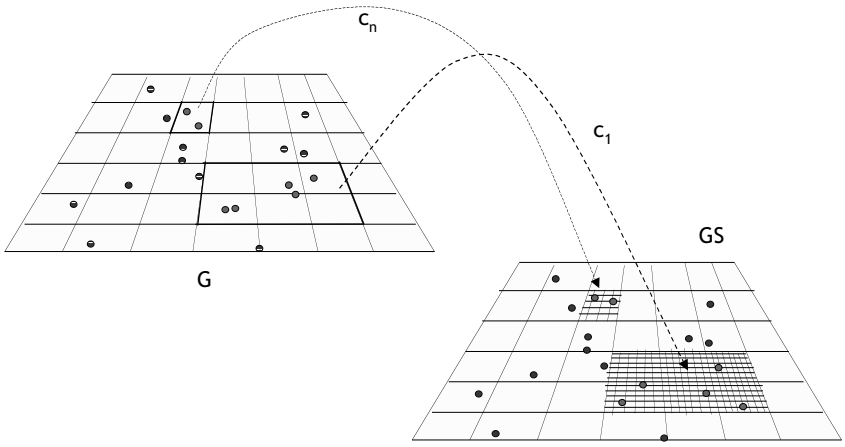


Figure 3. *Affinement du maillage et correspondances entre espaces vectoriels*

6.2. *Dépliage et pliage de vecteurs*

6.2.1. *Correspondances ontologiques*

À un concept c_G de G, on peut associer un ensemble de concepts de S. On appellera une telle association $< C_G, \{C_{S,1}, \ldots, C_{S,n}\} >$ une *correspondance ontologique*. Par exemple, le concept *ÉCONOMIE* de G est associé à toute la sous-arborescence de S contenant ce terme (économie politique, économie de marché, économie dirigé, microéconomie, macroéconomie, etc.)

On se donne, comme contrainte, que l'ensemble des correspondances couvre tout S. C'est-à-dire que l'ensemble des concepts atteint dans S est égal à S. Il s'agit donc d'une surjection. En revanche, ce n'est absolument pas une injection, car il existe des concepts de G qui ne sont pas dans les champs sémantiques de S (qui, on le rappelle sont par définition plongés dans G).

6.2.2. *Dépliage*

La fonction de dépliage D est une projection d'un vecteur v_G de G sur GS : $v_G \rightarrow v_{GS}$, qui permet d'affiner la représentation de ce vecteur si celui-ci est concerné par

les concepts de S. C'est ce que l'on nomme aussi *l'extension ontologique* du vecteur v.

D(v) est un vecteur de $G \cup S$ et se compose comme un vecteur de G suivi d'un vecteur de S, et $dim(D(v)) = dim(G) + dim(S)$. La première partie (nommée Kern) de $D(v)$ est v. La seconde partie (nommée Ext) se calcule comme suit à partir de v (de G) et de la liste des correspondances $\mathcal{C}$:

Procédure déplier (v_G, $\mathcal{C}$) $\rightarrow v_{GS}$

soit $P =< 0, \cdots, 0 >$ % P est un vecteur de taille $\dim(S)$;
pour chaque $< C_G, \{C_{S,1}, \ldots, C_{S,n}\} >$ de $\mathcal{C}$ **faire**
soit $x = v(C_G)$ % x est la valeur de v à la composante C_G
chaque composante $\{C_{S,1}, \ldots, C_{S,n}\}$ de P est incrémenté de x
fin pour
$Ext = p_1 * VC_1 + \cdots + p_{\dim(S)} * VC_{\dim(S)}$
v est normé
retourner v

Fin Procédure déplier

Le vecteur $P =< p_1, \ldots, p_n >$ représente les pondérations pour la somme des $\dim(S)$ vecteurs des concepts VC_i de S. C'est à partir de P que l'on construit Ext. On remarquera dans le vecteur obtenu ne contient jamais de zéro si les vecteurs des concepts invoqués ne contiennent pas de zéros. C'est en effet le cas par construction pour les vecteurs de G. Les vecteurs sont donc très denses.

6.2.3. *Pliage*

La fonction de pliage P est une projection d'un vecteur v_{GS} de GS sur G : $v_{GS} \rightarrow v_G$. Pour plier un vecteur de GS sur G, il suffit de *supprimer* Ext. En pratique, on crée un vecteur qui ne contient que Kern. Cela permet de récupérer sur une plus petite base, les termes peu touchés directement par la spécialité ou d'en avoir aussi une acception plus générale.

$$D(v)_{GS} \quad = \quad < x_1, \ldots, x_{dim(G)}, \ldots, x_{dim(G)+dim(S)} >$$

$$\rightarrow \quad < x_1, \ldots, x_{dim(G)} > \quad = \quad v_G$$

Si les procédures de construction (mais également, dans une moindre mesure, celles du noyau et de l'apprentissage) pour les vecteurs des concepts sont *bonnes* alors les concepts de G qui sont *relié* à ceux de S ont bougé si ceux de S ont été modifiés (et réciproquement). Les procédures présentées assurent cette propriété.

Le pliage est une projection qui perd de l'information, en particulier s'il s'agit de termes à la fois généraux et spécialisés, comme ‹*échange*› ou encore ‹*marché*›, cependant l'activation des concepts de la partie G reflète l'activation des concepts de S.

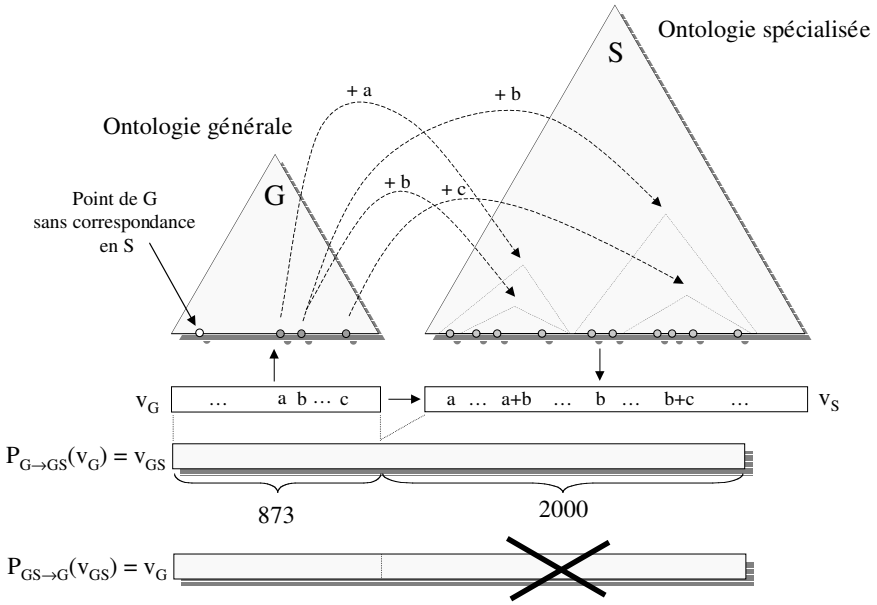


Figure 4. *Correspondances ontologiques. Le pliage est une projection P de GS sur G et le dépliage une projection P de G sur GS*

6.3. *Propriétés*

Une première propriété concerne la composition des fonctions D (de dépliage) et P (de pliage).

$$P(D(v)) = v$$

Déplier puis replier un vecteur équivaut à la fonction identité. Mais dans le cas général, nous n'avons pas l'inverse $D(P(v)) \neq v$ puisqu'il y a perte d'information. Nous avons également une réduction relative de la distance angulaire D_A :

$$D_A(v1, v2) \leq D_A(D(v1), D(v2))$$

Ce phénomène peut se traduire ainsi : *l'extension ontologique augmente la synonymie (hors apprentissage)*. Cela se démontre (et s'expérimente) à partir de la définition de la distance angulaire donnée en section 4.2. Par contre, le *raffinement ontologique* (c'est-à-dire l'analyse d'un terme dans GS au lieu de G) peut soit :

1) Réduire la synonymie (c'est-à-dire augmente la distance sémantique) pour deux termes de spécialité.

64 TAL. Volume 43 - n° 1/2002

Deux termes quasi identiques dans G s'éloignent conceptuellement, ce qui permet de les discriminer davantage. Par exemple : ‹*finances publiques*› et ‹*fiscalité*› sont dans G à $D_A = 0,3$ (environ 17 degrés). Dans S, nous avons $D_A = 1,2$ (environ 69 degrés).

2) Augmenter la synonymie par réduction de la polysémie.

Par exemple, dans G, le terme t1 a deux sens t11 et t12, et t2 a deux sens t21 et t22. On suppose que t11 et t22 sont deux sens synonymes. La distance globale de t1 et t2 peut être (relativement) élevée car t12 et t21 constituent du bruit. En revanche, comme dans S seuls t11 et t22 appartiennent au domaine, nous avons (dans S) t1 et t2 qui sont monosémiques et synonymes.

C'est globalement le cas pour les termes ‹*profit*›, ‹*bénéfice*› et ‹*produit*›.

Dans S, les termes de spécialité sont moins polysémiques, et chacun a une description très fine et séparée des autres. Dans G, ces termes sont souvent fortement polysémiques, les descriptions sont moins fines et moins séparées (elles ont tendance à s'agglutiner en classes d'équivalence lors de l'application de filtres sémantiques basés sur la distance angulaire).

6.4. *Construction de vecteurs ontologiques de* GS

Construction des vecteurs de concepts de S. Les vecteurs de S se construisent comme ceux de G. Pour mémoire, il s'agit de s'appuyer sur l'ontologie pour construire les $\dim(S)$ vecteurs des concepts de S. Cette construction utilise la distance ultramétrique et les activation transverses éventuelles.

Construction des vecteurs de concepts de GS. La question est en fait de savoir comment *ajouter* le vecteurs G (et lequel) à chaque vecteur de concept de S que l'on a produit précédemment. Une solution est d'*inverser* le dépliage (dépliage inverse). On applique la même procédure que *déplier*, mais à partir d'un vecteur de S, on construit un vecteur de G. On peut trivialement inverser une correspondance en une liste de correspondances :

$$\begin{aligned} \mathcal{C} &= < C_G, \{C_{S,1}, \ldots, C_{S,n}\} > \\ \rightarrow \quad (< C_{S,1}, \{C_G\} >, \ldots, < C_{S,n}, \{CG\} >) &= \mathcal{C}' \end{aligned}$$

Le vecteur de S peut être concaténé à gauche de son dépliage inverse qui produit la partie sur G :

$$\text{déplier}(v_S, \mathcal{C}') + v_S \quad \rightarrow \quad v_{GS}$$

Par la suite, la construction des vecteurs du noyau de GS (extensible à tous les termes de S) et l'apprentissage des termes sur GS s'effectue comme pour G.

7. Fonctions de filtres sémantique et changement d'espace ontologique

La densité lexicale permet de mesurer le degré d'appartenance d'un terme (ou d'un de ses sens) à une ontologie donnée. Cette mesure se base sur les variations observées

pour ce terme entre l'espace vectoriel général et l'espace vectoriel spécialisé. Il en est de même pour les deux relations que sont la synonymie et l'antonymie.

7.1. *Distribution et concentration lexicale*

La distribution lexicale $\mathcal{D}_E(\mathrm{t})$ d'un terme t dans l'espace S est la répartition des termes en fonction de leur distance à t. Par exemple la figure 5 représente la distribution lexicale du terme ‹*marché*› de G. Les termes se répartissent en général autour d'un sommet. On observe, systématiquement quel que soit le terme choisi, une *petite bosse* située entre le sommet et $\pi/2$ qui est un point d'accumulation des vecteurs des concepts (qui sont naturellement éloignés des vecteurs des termes).

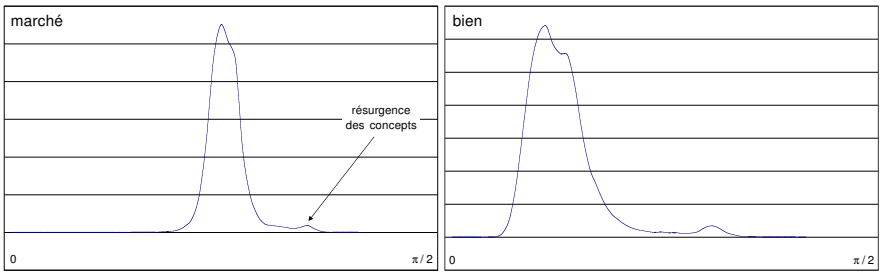


Figure 5. *Distribution lexicale de* ‹marché› *et de* ‹bien› *dans* G

On définit l'*intervalle de proximité thématique*, la fonction $\mathcal{I}_{E,\mathrm{f}}(\mathrm{t})$ avec ($0 \leq f \leq 1$) qui retourne l'intervalle le plus petit dans lequel se trouve la fraction f du lexique de G, *la plus proche* de t (en excluant évidement t lui-même). Par exemple, $\mathcal{I}_{\mathcal{G},0,5}(\mathrm{marché})$ correspond à l'intervalle qui contient la moitiée des mots de G qui sont les plus proche thématiquement du terme ‹*marché*›. Nous avons ici, $\mathcal{I}_{G,0.5}(\mathrm{marché}) = [0,25\,;0,91]$. Il s'agit géométriquement de dire que la moitié des termes de G par rapport à ‹*marché*› se situent entre les deux hypersphères de rayon $0,25$ et $0,91$. Nous avons, $\mathcal{I}_{G,0,5}(\mathrm{bien}) = [0,16\,;0,43]$.

On appelle la *concentration lexicale* $\partial\mathcal{I}$ d'un intervalle $\mathcal{I}$, le pourcentage du lexique couvert divisé par l'écart de cet intervalle. Ici, $\partial\mathcal{I}_{G,0,5}(\mathrm{marché}) = 0,5/(0,91 - 0,25) = 0,76$ et $\partial\mathcal{I}_{G,0,5}(\mathrm{bien}) = 0,5/(0,43 - 0,16) = 1,85$.

Si la concentration lexicale de t est faible alors la courbe est décalée vers $\pi/2$ (il y a peu de termes autour de t). Et inversement, si elle est importante, la courbe est décalée vers 0.

Un terme peut avoir une densité lexicale forte dans plusieurs cas (non exclusifs) :

1) Le terme appartient à un champ sémantique très riche. Par exemple, les noms d'insecte ont une densité lexicale très forte, tout simplement parce qu'ils sont en très grand nombre.

66 TAL. Volume 43 - n° 1/2002

2) Le terme est souvent utilisé dans des définitions comme hyperonyme (il a une forte valeur conceptuelle). C'est le cas de terme comme ‹*insecte*›, ‹*plante*›, ‹*élément*›, ‹*travail*›, ‹*nombre*›... Ces termes sont fortement hyperonymiques, mais des termes très généraux (comme ‹*homme*› ou ‹*former*›) bien qu'ayant une fréquence très élevée ont une concentration lexicale faible. Ils ne sont pas particulièrement porteur de sens, et donc ne participent que peu à la constitution du sens d'un mot.

3) Le terme est très polysémique. Il a tendance à se ramener aux deux cas ci-dessus.

Pour un terme t, nous pouvons nous intéresser, selon l'espace vectoriel considéré, à deux facteurs. D'une part à la variation de la densité lexicale (c'est-à-dire formellement à la taille de l'intervalle), et d'autre part à la variation de positions de cet intervalle.

On remarque que globalement la concentration lexicale *chute* avec l'extension ontologique. C'est-à-dire qu'un terme très polysémique dans G, ne correspond qu'à un nombre de sens très réduit (voire unique) dans S. Il s'agit par exemple du cas de ‹*marché*›, qui prend des sens très spécifiques dans l'ontologie de l'OCDE (et dans les définitions du DAFA). Par exemple, $\partial\mathcal{I}_{S,0,5}(\text{marché}) = 0,68$. De façon, encore plus nette, nous obtenons $\partial\mathcal{I}_{G,0,5}(\text{bien}) = 0,5/(0,43 - 0,16) = 0,8$. Dans S, ‹*bien*› ne correspond qu'au substantif masculin dont la définition est *chose produite pour satisfaire un besoin* et qui correspond directement à un des concepts de S. Ce terme est donc très conceptuel, mais ne constitue pas un terme fortement hyperonymique.

On rappelle que dans nos expériences, tous les termes d'un domaine de spécialité S sont inclus dans l'espace vectoriel général G. Pour un terme t_S issu de S, on peut donc faire la constatation suivante :

$$\partial\mathcal{I}_{G,\mathrm{f}}(\mathrm{t}_S) \geq \partial\mathcal{I}_{S,\mathrm{f}}(\mathrm{t}_S)$$

La densité lexicale dans G est plus forte que dans S. En effet, le maillage étant plus fin dans S, ce terme est mieux discriminé. Le passage du terme t_S de S dans G se fait par pliage.

Si nous avons (dans de rares cas) :

$$\partial\mathcal{I}_{G,\mathrm{f}}(\mathrm{t}_S) \leq \partial\mathcal{I}_{S,\mathrm{f}}(\mathrm{t}_S)$$

cela indique qu'un terme de S dispose d'un certain nombre de termes proches qui s'éloignent dans G. C'est possible, si les termes en question sont fortement polysémiques dans G ou très généraux. Par exemple, dans S le terme de ‹*concentration*› peut être utilisé de façon elliptique pour de nombreux termes associés ayant un sens très précis (non réellement calculables par composition) : *concentration d'entreprise*, *concentration verticale*, *concentration horizontale*, *concentration d'un secteur*, *concentration dans un secteur*... Par contre dans G, le terme de concentration est très général et les plus proches voisins sont globalement plus éloignés que dans S. Ce dernier phénomène est plus rare que le premier. C'est pourquoi globalement la densité lexicale chute.

La distribution et la concentration lexicale fournissent ainsi des filtres permettant de savoir effectivement si un terme *t* peut (ou non) appartenir à une ontologie de spécialité S. Il s'agit des mesures fournissant un degré de confiance. En fixant, *a priori*, une valeur seuil, il est ainsi possible d'extraire automatiquement le vocabulaire spécialisé d'un domaine. Ce vocabulaire se compose des termes de spécialité et les sens des mots généraux qui sont pertinents pour cette spécialité. Par exemple, dans la terminologie pétrolière, le terme ‹*poisson*› est bien sélectionné comme étant un *segment de trépan brisé et logé au fond du puits de forage et que l'on doit aller pêcher*.

La synonymie et l'antonymie permettent, elles, d'affiner cette extraction terminologique en établissant entre les termes des relations de *sens proches* et des *sens en opposition*.

7.2. *Utilisation de la synonymie comme un révélateur de structures*

Il s'agit ici de l'étude dystopique de la synonymie relative, c'est-à-dire de la comparaison de son comportement entre les espace G et GS. La densité lexicale consitue une fonction macroscopique à l'échelle du lexique. À l'inverse, la synonymie relative ici est une fonction microscopique à l'échelle du terme. La synonymie relative constitue ainsi une fonction typique de filtrage sur les points des espace vectoriels. Il s'agit ici d'étudier le comportement de la fonction de synonymie relative $Syn_R(A, B, C)_{\mathcal{E}}$ selon que l'on considère pour espace vectoriel $\mathcal{E}$, l'espace général G ou l'espace augmenté GS.

Considérons tout d'abord le cas plus simple de la synonymie absolue (qui n'est qu'un cas particulier de la synonymie relative). Nous cherchons ici à comparer les valeurs $Syn_A(A, B)_{\mathcal{GS}}$ et $Syn_A(A, B)_{\mathcal{G}}$.

On peut distinguer plusieurs cas selon l'appartenance des termes à S :

1) A, B sont dans S. Dans ce cas si :

$$Syn_A(A, B)_{\mathcal{GS}} \leq Syn_A(A, B)_{\mathcal{G}}$$

alors les termes sont discriminés dans S grâce à l'affinement du maillage. C'est le cas de termes comme ‹*commerçant*›, ‹*marchant*›, ‹*détaillant*›, ‹*grossiste*›, ‹*négociant*›, ‹*fournisseur*›, ‹*revendeur*›, ... Tous ces termes sont quasiment synonymes dans G mais sont très différents dans GS. Par contre si :

$$Syn_A(A, B)_{\mathcal{GS}} \geq Syn_A(A, B)_{\mathcal{G}}$$

les termes se sont rapprochés dans S. Il s'agit d'un cas typique où la polysémie dans GS éloigne deux termes qui ont un sens proche en commun. C'est par exemple, le cas pour ‹*travail*› et ‹*emploi*›, ou encore ‹*traitement*› et ‹*salaire*›.

2) Soit A, soit B est dans S. Dans ce cas, on a toujours une diminution de la synonymie. $Syn_A(A, B)_{\mathcal{GS}} \geq Syn_A(A, B)_{\mathcal{G}}$.

68 TAL. Volume 43 - n° 1/2002

3) Ni A ni B ne sont dans S, les deux termes deviennent bien plus synonymes. On a donc bien $Syn_A(A,B)_{\mathcal{GS}} \leq Syn_A(A,B)_{\mathcal{G}}$.

Pour la synonymie relative $Syn_R(A,B,C)$, la question se ramène à évaluer la situation selon que C est ou non un terme de S. Si C est un terme acceptable (au sens de la concentration lexicale) pour S alors les mesures de synonymie sont plus pertinentes. Inversement, si C n'est pas un terme acceptable pour S, les mesures de synonymie sont dégradées. En particulier, si ni A, ni B, ni C ne sont dans S, cela se ramène bien au troisième cas ci-dessus. Cela signifie que la synonymie relative est un bon indice de structure lorsque le pivot de cette structure (C) est pertinent pour l'ontologie.

7.3. *Utilisation de l'antonymie comme un révélateur de structures*

On ne considère ici que la fonction d'antonymie globale (telle qu'elle est décrite dans [Schwab 2001] et [Schwab *et al.* 2002] *op. cit.*). La relation d'antonymie peut émerger, disparaître ou être conservée quand on passe de G vers GS, d'une part, et de GS vers G, d'autre part. La terminologie procédant par métaphore et composition des termes génériques, la relation antonymique est souvent préservée (par exemple, les *médias froids* et les *médias chauds* de McLuhan).

La plupart du temps, pour le vocabulaire fortement terminologique, l'antonymie utilisée sera du troisième type (*duale*, qui concerne les oppositions culturelles) à cause de l'utilisation intensive de la métaphore dans la création terminologique. On ne peut en effet guère déduire par l'analyse le sens strict des termes qui s'opposent essentiellement à travers l'organisation de l'ontologie (et non forcément en tant que tels). Par exemple (OCDE) :

1) ‹*mortalité*› ↔ ‹*fécondité et planification de la famille*›

2) ‹*zone rurale*›↔ ‹*zone urbaine*›

3) ‹*groupes d'ages*› ↔ ‹*groupes éthniques*›

En revanche, les définitions des termes teminologiques (par exemple issus du DAFA) font largement appel à l'opposition. Ce qui peut alimenter la construction incrémentale de fonctions d'antonymie. Ces fonctions peuvent ensuite jouer un rôle de filtre au même titre que la synonymie, afin de faire émerger des structures cachées dans les agglomérations (ou séparations) de vecteurs. Par exemple : ‹*économie de marché*› = ‹*économie libérale*› ↔ ‹*économie dirigée*›.

L'étude de l'antonymie est similaire à celle de la synonymie. Nous cherchons donc à comparer, dans un premier temps, les valeurs de $M_{anti-A}(A,B)_{\mathcal{GS}}$ et de $M_{anti-A}(A,B)_{\mathcal{G}}$.

1) A, B sont dans S. Dans ce cas si :

$$M_{anti-A}(A,B)_{\mathcal{GS}} \leq M_{anti-A}(A,B)(A,B)_{\mathcal{G}}$$

alors les termes sont plus fortement antonymes dans GS que dans G. Il s'agit encore une fois de l'effet de l'affinement du maillage. C'est le cas de termes comme ‹*économie libérale*› (A) et ‹*économie dirigée*› (B) : $M_{anti-A}(A,B)_{\mathcal{G}} = 0,6$ et $M_{anti-A}(A,B)_{\mathcal{GS}} = 0,3$. Nous avons aussi, le cas de ‹*travail*› (A) et ‹*chômage*› (B) à cause de la polysémie de ‹*travail*›. Nous avons : $M_{anti-A}(A,B)_{\mathcal{G}} = 0,35$ et $M_{anti-A}(A,B)_{\mathcal{GS}} = 0,48$. Par contre si :

$$M_{anti-A}(A,B)_{\mathcal{GS}} \geq M_{anti-A}(A,B)(A,B)_{\mathcal{G}}$$

les termes sont moins antonymes dans GS que dans G. Il s'agit d'un cas ou les concepts sur lesquels s'opposent les termes dans G, soit ne s'opposent plus dans S ou ne sont pas pertinents (et donc ne s'opposent plus). C'est le cas avec ‹*boucher*› (A) et ‹*poisonnier*› (B) car dans G *poisson* et *viande* s'opposent dualement. Nous avons : $M_{anti-A}(A,B)_{\mathcal{G}} = 0,57$ et $M_{anti-A}(A,B)_{\mathcal{GS}} = 0,48$. Les concepts liés à *poisson* et *viande* ne sont pas pertinents dans S et donc leur poids dans GS s'en trouve considérablement amoindri.

2) Soit A, soit B est dans S. Dans cas, la variation dépend de leur opposition potentielle dans S.

3) Ni A ni B ne sont dans S, les deux termes deviennent beaucoup moins antonymes.

8. Conclusion

Les expériences que nous avons menées autour de l'intégration d'une ontologie de spécialité (ici, le domaine économique) à une ontologie générale fondée sur les concepts du thésaurus, et munie du dispositif calculatoire du modèle vectoriel, nous ont permis d'aboutir aux conclusions suivantes.

1. Lorsqu'il faut analyser, classer ou indexer des textes de spécialité, la meilleure solution consiste à utiliser une union entre l'ontologie générale et l'ontologie de spécialité parce que les textes de spécialité ne contiennent pas que des termes techniques. Le passage de l'une à l'autre a été décrit dans la section 6 de l'article à l'aide de procédures et d'algorithmes testés et finalisés.

2. Lorsqu'un apprentissage automatique de concepts spécialisés est réalisé à partir de définitions fournies dans des dictionnaires, cette union d'ontologies s'avère indispensable, puisque tous les mots de la définition peuvent alors contribuer à fournir les éléments pour le calcul du sens.

3. Dans le modèle des vecteurs conceptuels, l'ontologie de spécialité est beaucoup plus fournie que l'ontologie générale, contrairement à l'approche d'*arborescence de connaissances* classique. En revanche, ce que nous avons découvert est que la quantité d'information à stocker est plus petite pour une analyse de textes techniques que pour une analyse de textes généraux. Nous avons ainsi déplacé le problème de la taille depuis l'ontologie vers la quantité d'information.

4. L'ontologie de spécialité permet un maillage plus serré de la représentation du sens, donc une meilleure discrimination sémantique entre des termes qui apparaîtraient proches. Inversement, le calcul du sens et des distances sur cette ontologie permettent de rendre très proches, voire *synonymes*, des sens qui, projetés sur l'ontologie générale, ne le seraient pas. La polysémie, caractéristique principale du lexique général est alors circonscrite au profit des sens spécialisés des termes invoqués.

5. Justement, la notion de synonymie calculée, ainsi que celle d'antonymie (qui permet de traiter d'éventuelles négations) est l'un des grands apports du modèle vectoriel tel que nous le pratiquons (section 5). Dans la majorité des cas, les travaux sur la synonymie partent d'une synonymie prédite ou fournie ([Ploux et Victorri 1998] et [Hathout 2001]). Le modèle vectoriel permet de tester la validité d'une proximité supposée et, dans son raffinement, celui de la synonymie relative, il permet d'explorer les relations qu'entretiennent les termes autour d'un terme dit de *contexte*. Emerge alors une microstructuration, ou plus exactement une microtopologie, qui permet de revisiter l'espace vectoriel lexical avec une notion de *densité lexicale* (section 7) au voisinage d'un vecteur. L'antonymie, à laquelle les définitions de dictionnaires font largement appel pour mettre en contraste une notion par rapport à une autre, aussi bien que la synonymie relative, sont des révélateurs de structure émergeante et dynamique.

Ces conclusions partielles vont dans le sens d'une conclusion plus générale : faire appel à une terminologie de spécialité pour traiter des textes techniques est non seulement faisable dans le modèle vectoriel, mais celui-ci permet d'unir les ontologies, de discriminer des sens, de circonscrire la polysémie, et de faire émerger une microstructuration qui pourra être modifiée au gré de l'apprentissage continu que permet le modèle. Les expériences menées ont permis l'intégration d'une terminologie sous forme d'une ontologie de $2\,000$ concepts feuilles (issue de l'OCDE) et d'analyser des textes définitoires en provenance, entre autres, du dictionnaire des affaires (le DAFA). la construction de l'ontologie terminologique est déjà achevée et les liens émergeants transverses de synonymie et d'antonymie ont été utilisés pour constater des rapprochements entre notions et une amélioration de la discrimination sémantique. Ces tests peuvent être répétés par tout utilisateur qui le souhaite sur un site web[9], où le système est à la disposition de tous.

9. Bibliographie

[Barrière and Copeck 2001] Barrière C., Copeck T., "Building Domain Knowledge from Specialized Texts", *TIA 2001*, Nancy, 2001.

[Bourrigault 1993] Bourrigault D., "Analyse locale pour le repèrage des termes complexes dans les textes", *TAL*, vol. 34, n° 2, p. 105-118, 1993.

[Chauché 1990] Chauché J., "Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance", *TA Information*, vol. 31, n° 1, p. 17-24, 1990.

9. <http ://www.lirmm.fr/~lafourca>

[Cruse and Togia 1995] Cruse D.A., Togia P., “Towards a cognitive model of antonymy”, *Lexicology*, vol. 1, p. 113-141, 1995.

[DAFA 2001] Verlinde S., Selva T., *DAFA - Dictionnaire d’Apprentissage du Français des Affaires*, http ://www.projetdafa.net.

[Deerwester *et al*. 1990] Deerwester S., Dumais S., Landauer T., Furnas G., Harshman R., “Indexing by latent semantic analysis”, *Journal of the American Society of Information Science*, 416(6), p. 391-407, 1990.

[Fellbaum 1995] Fellbaum C., “Co-occurrence and antonymy”, *International Journal of Lexicography*, vol. 8, p. 281-303, 1995.

[Fischer 1973] Fischer W. L., *Äquivalenz und Toleranz Strukturen in der Linguistik zur Theory der Synonyma*, Max Hüber Verlag, München, 1973.

[Gwei and Foxley 1987] Gwei G.M., Foxley E., “A Flexible Synonym Interface with application examples in CAL and Help Environments”, *The Computer Journal*, vol. 30 n˚6, p. 551-557, 1987.

[Hamon et Nazarenko 2001] Hamon T., Nazarenko A., “La structuration de terminologie : une nécessaire coopération”, *TIA 2001*, Nancy, 2001.

[Hathout 2001] Hathout N., “Analogies morpho-synonymiques. Une méthode d’acquisition automatique de liens morphologiques á partir d’un dictionnaire de synonymes”, *TALN 2001*, Tours, vol. 1, p. 223-232, juillet 2001.

[Hearst 1998] Hearst M.A., “Automated discovery of Wordnet relations”, In C. Fellbaum ed. *Wordnet An Electronic Lexical Database*, MIT Press, Cambridge, MA, p. 131-151, 1998.

[Justeson and Katz 1991] Justeson J.S., Katz S., “Co-occurrences of antonymous adjectives and their contexts”, *Computational Linguistics*, vol. 17, p. 1-19, 1991.

[Lafourcade et Sandford 1999] Lafourcade M., Sandford E., “Analyse et désambiguïsation lexicale par vecteurs sémantiques”, *TALN’99*, Cargèse. p. 351-356, juillet 1999.

[Lafourcade 2001] Lafourcade M., “Lexical sorting and lexical transfer by conceptual vectors”, *First International Workshop on MultiMedia Annotation* (MMA’2001), Tokyo, 6 p, January 2001.

[Lafourcade et Prince 2001] Lafourcade M., Prince V., “Synonymies et vecteurs conceptuels”, *TALN 2001*, Tours, p. 233-242, juillet 2001.

[Larousse 2001] Larousse, *Thésaurus Larousse - des idées aux mots - des mots aux idées.* Larousse, 1992.

[OCDE 1991] OCDE, “Macrothesaurus”, http ://info.uibk.ac.at/info/oecd-macroth/, 1991.

[Prince 1991] Prince V., “Notes sur l’évaluation de la réponse dans TEDDI : introduction d’une relation d’équivalence pour la synonymie relative”, *Notes et Documents LIMSI*, 91-20, CNRS, 1991.

[Resnik 1995] Resnik P., “Using Information contents to evaluate semantic similarity in a taxonomy”, *IJCAI-95*, 1995.

[Riloff and Shepherd 1995] Riloff E., Shepherd J., “A corpus-based bootstrapping algorithm for Semi-Automated semantic lexicon construction”, *Natural Language Engineering*, vol. 5, part. 2, p. 147-156, 1995.

[Salton 1968] Salton G., *Automatic Information Organisation and Retrieval*, McGraw-Hill, New York, 1968.

[Salton and MacGill 1983] Salton G., MacGill M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

[Salton 1988] Salton G., *Term-Weighting Approaches in Automatic Text Retrieval, McGraw-Hill computer science series*, McGraw-Hill, vol. 24, 1988.

[Schwab 2001] Schwab D., "Vecteurs conceptuels et fonctions lexicales : application à l'antonymie", Mémoire de DEA Informatique, 2001.

[Schwab *et al.* 2002] Schwab D., Lafourcade M., V. Prince V., "Amélioration de la représentation sémantique lexicale par les vecteurs conceptuels : le rôle de l'antonymie", *JATD 2002*, vol. 2, p. 701-712, 2002.

[Sparck Jones 1986] Sparck Jones K., *Synonymy and Semantic Classification*, Edinburgh Information Technology Serie, 1986.

[Ploux et Victorri 1998] Ploux S., Victorri B., "Construction d'espaces sémantiques á l'aide de dictionnaires de synonymes. " *TAL*, vol. 39, n° 1, p. 161-182, 1998.

[Yarowsky1992] Yarowsky D., "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora", *COLING'92*, Nantes, p. 454-460, 1992.

CHAPITRE 2

Construction de vecteurs d'idées

Comment construire des vecteurs d'idées associés aux éléments du lexique ? La source d'information peut-être un corpus de textes, de définitions de dictionnaire, ou un thésaurus. La méthode de calcul doit extraire l'information pertinente et produire un vecteur selon deux approches possibles : 1) une taille définie implicitement par un ensemble de points d'ancrage, 2) une taille définie arbitrairement et un calcul pseudo aléatoire (dit par émergence).

Articles joints

M. Lafourcade *Conceptual Vector Learning - Comparing Bootstrapping from a Thesaurus or Induction by Emergence.* In proc. LREC'2006, Magazzini del Cotone Conference Center, Genoa, Italia, 24-26 May 2006.

V. Prince, et M. Lafourcade *Mixing Semantic Networks and Conceptual Vectors - Application to Hyperonymy.* IEEE Transactions on Systems, Man, and Cybernetics : Part C. 2006, 11 p.

Encadrement - Didier Schwab, Fabien Jalabert, *groupes de TER : [Barbier et al., 2008], [Lopez & Zouani, 2008]*

Comment construire des vecteurs d'idées associés aux éléments du lexique ? Plusieurs approches sont possibles, variant tant sur l'approche adoptée pour l'algorithme que sur la nature des données exploitées. L'évaluation de la qualité des vecteurs produits peut être réalisée sur une échelle importante via la confrontation indirecte du voisinage des termes et ce que fournissent des êtres humains.

2.1 Intérêt et approches existantes

La *vectorisation* du lexique est une forme de compilation de ce dernier. L'intérêt de construire de tels vecteurs pour des objets lexicaux réside essentiellement dans la facilité à les comparer sous cette forme. Les mesures de comparaison pour des vecteurs (ou des ensembles) qui ont été introduites dans le domaine sont nombreuses, mais somme toute, se résument à un calcul d'information mutuelle, la plus répandue d'entre elles étant la *similarité cosinus* (à savoir le produit scalaire des vecteurs normés). Nous ferons remarquer que la mesure de comparaison utilisée a un impact très réduit sur les performances relativement à, d'une part la source d'information utilisée pour la collecte (spécialistes, corpus, etc.), et d'autre part la réalisation effective de la fonction de vectorisation.

Le modèle vectoriel introduit par Salton ([Salton & McGill, 1983], [Salton, 1991]) a vu de nombreux développements, en particulier sur la manière de construire les vecteurs. Il faut distinguer ici, les vecteurs calculés à partir d'un texte pour représenter celui-ci de ceux construits afin de constituer un lexique en sémantique lexicale. Dans l'approche saltonienne un vecteur est produit pour chaque document d'un corpus, dans le but de pouvoir les comparer. Une requête de recherche de document est transformée en un vecteur et le voisinage de ce vecteur requête dans l'espace des documents en constitue la réponse.

L'approche de l'Analyse Sémantique Latente (en anglais Latent Semantic Analysis, ou encore LSA, [Deerwester *et al.*, 1990b], [Rehder *et al.*, 1998], [Landauer *et al.*, 1998]) est un des modèles de mémoire sémantique ayant rencontré le plus de succès. Il est fondé sur le calcul de cooccurrences à partir d'un corpus. Un calcul de vecteurs singuliers et une réduction de dimension sont effectués de façon à, d'une part appliquer une forme de transitivité, et d'autre part réduire le bruit. Le modèle Hyperspace Analog to Language (HAL) ([Lund *et al.*, 1995], [Lund & Burgess, 1996]) est plus simple et direct que LSA, et en particulier ne procède pas à une réduction de dimension. Cependant, la finalité est identique à celle de LSA, à savoir extraire à partir d'un corpus de grande taille, une forme de réseau de coocurences sous forme vectorielle.

Dans [Véronis & Ide, 1990] est présentée la construction d'un réseau lexical associatif à partir de définitions de dictionnaires. Ce réseau est localement assimilable à des signatures lexicales. Le fait d'utiliser comme source d'information des définitions de dictionnaire situe les travaux présentés ici dans une approche similaire à celles adoptées par [Lesk, 1986].

2.2 Construction par propagation et points d'ancrage

En amont de la construction des vecteurs conceptuels, se pose le problème de l'identification d'un ensemble de concepts assez grand pour exprimer n'importe quel sens d'un mot en s'appuyant sur les seuls éléments de cet ensemble. Du fait de la nature même de la langue naturelle, un tel choix est difficile et toujours critiquable. Dans notre cas, nous avons décidé d'avoir recours à un thésaurus et nous avons démarré avec celui de Larousse (édition 1992, [Larousse, 1992a] et [Pechoin, 1991]). Pour conserver la cohérence avec l'*hypothèse du thésaurus*, nous supposons que cet ensemble constitue un espace générateur pour les mots et leurs sens, de telle sorte qu'il est possible de projeter le sens de n'importe quel mot dans cet espace. Notons toutefois que cet espace n'est sans doute pas libre. Les points d'ancrage à partir desquels le peuplement de l'espace est réalisé, sont les vecteurs des concepts tels qu'énumérés dans [Larousse, 1992a].

Ce thésaurus est très classiquement composé de deux parties. La première décrit une hiérarchie de 873 concepts, et un index contenant des milliers de mots constitue la seconde partie. Les lexicographes de Larousse ont explicitement défini une famille de concepts organisée selon une hiérarchie à quatre niveaux (arbre de profondeur 4). Cet arbre possède 873 feuilles : les concepts identifiés. À chaque entrée de l'index est associée quelques concepts en rapport (parmi les 873 de la liste).

L'étape de démarrage consiste à construire les vecteurs conceptuels des 873 concepts eux-mêmes. Une manière simple de procéder aurait pu être d'associer un vecteur booléen (de dimension 873) à chacun des mot-concepts. Cependant, de notre point de vue ce choix n'aurait pas été judicieux. En effet, l'association d'un vecteur booléen à chaque concept signifierait implicitement que les concepts sont indépendants les uns des autres, ce qui en langage naturel n'est évidemment pas le cas. Nous avons donc choisi de définir chaque vecteur conceptuel des mot-concepts comme étant le résultat de l'expression du concept C lui-même combinée à l'expression d'un voisinage conceptuel pondérée par la structure hiérarchique du thésaurus. Plus le concept D est proche du concept C, dans la hiérarchie du thésaurus, plus son expression est importante dans le vecteur conceptuel de C.

Étant donnés des vecteurs conceptuels des concepts (que nous avons construits), un ensemble ini-

tial de mots a été choisi pour amorcer le processus [1]. Cet ensemble est composé d'environ 2000 termes français parmi les plus communément utilisés (qui possèdent donc une entrée dans l'index du thésaurus). De plus, chacun des 873 concepts était couvert par au moins un des termes de l'ensemble. Pour chaque terme, le vecteur conceptuel a été obtenu comme le résultat d'une combinaison linéaire des vecteurs des concepts liés à ce mot dans l'index.

Pour chaque nouveau mot, plusieurs définitions provenant de différentes sources (principalement des dictionnaires classiques et de synonymes) ont été utilisées. Chaque définition ou description est un texte formé de phrases. Le texte a été analysé en ayant recours à la méthode de *propagation standard*. Le vecteur conceptuel de chaque mot composant le texte de la définition est utilisé pour déterminer le vecteur conceptuel du nouveau mot. Pour les mots figurant dans la définition, et qui n'étaient pas connus, leur vecteur a été remplacé par le vecteur nul.

Le schéma général de la construction d'un ensemble de vecteurs construit par propagation est le suivant :

1. Définir les vecteurs de base (ceux des concepts) ;
2. Sélectionner au hasard un terme t devant être révisé. Soit $E_t = t_1, t_2, \cdots$, l'ensemble des termes présents dans la définition (au sens large) du terme t. Deux cas sont alors possibles :
   (a) t_i a un vecteur associé $V(t_i)$, ce vecteur est alors utilisé ;
   (b) t_i n'a pas de vecteur associé. Le vecteur nul est provisoirement associé à t_i.
3. Calculer le vecteur de t en fonction de chacun des vecteurs des t_i (fonction d'agrégation) ;
4. Recommencer à l'étape 2.

À des fins d'optimisation, le tirage aléatoire du terme peut être amendé via des listes de priorités. En particulier, un terme auquel le vecteur nul a été provisoirement associé, peut être inséré dans une telle liste, et ainsi sélectionné avec une probabilité plus grande. Il est de même pour les termes récemment rencontrés, afin d'avoir une fréquence de révision covariante avec la fréquence d'apparition d'un terme.

De 1999 à 2005, nous avons mené une expérience d'apprentissage de vecteurs conceptuels. En 2005, la base de données des vecteurs conceptuels contenait environ 700 000 vecteurs correspondant à environ 200 000 mots (noms propres et communs, verbes, adjectifs, adverbes, mais également acronymes, locutions, etc.). Des expériences de classification/clusterisation ont été menées de façon à regrouper les définitions de sources différentes correspondant à des acceptions identiques ([Jalabert, 2003], [Jalabert & Lafourcade, 2004b] et [Jalabert & Lafourcade, 2004a]). Les définitions de dictionnaires (à usage humain) posent de sérieux problèmes d'analyse, en particulier au niveau de l'identification du métalangage. Par exemple, les deux définitions suivantes :

> *anthropophage* : ≪ qui mange de l'homme en parlant d'un homme. ≫
>
> *repas d'affaires* : ≪ repas au cours duquel on mange en parlant d'affaires. ≫

ont des structures syntaxiques suffisamment proches et seule leur compréhension fine permet de déterminer dans quel cas le segment *en parlant de* relève du métalangage.

L'algorithme de classification était une construction ascendante d'un arbre avec la fonction de comparaison la fonction de similarité entre vecteurs. Le regroupement dans les bons sous-arbres des définitions (multi-sources) était correct à environ 96%. L'évaluation a été faite sur un double échantillonnage de 500 termes chacun : 1) les 500 termes les plus fréquents du lexique (au sens du nombre d'occurrences d'appels de vecteur), et les 500 termes polysémiques suivants dans la fréquence. Les 38 cas de mauvaise classification étaient dûs à des vecteurs de mauvaise qualité systématiquement mal calculés à cause de la formulation ambigüe de la définition (problème du métalangage).

1. *Amorçage* (bootstrapping) est le terme généralement utilisé pour faire référence à cette étape.

2.3 Construction par émergence

Le principe de construction par émergence permet de s'affranchir d'un ensemble de concepts défini *a priori*. Il est ainsi possible de choisir librement la taille nécessaire pour les vecteurs, ce choix étant un compromis entre finesse de la représentation et coût de calcul/stockage. Une hypothèse forte est que *plus un vecteur sera de grande dimension, meilleure sera la représentation* (toutes choses étant égales par ailleurs). Toutefois, les vecteurs obtenus ne sont pas directement décodables, en ce sens que leurs composantes ne sont pas explicitement associées à des concepts (ou des termes). Il est possible d'évaluer si un vecteur est bien formé par évaluation de son voisinage $\vartheta(V)$.

Le schéma général de la construction d'un ensemble de vecteurs construit par émergence est le suivant :

1. Définir la taille des vecteurs ;
2. Sélectionner au hasard un terme t devant être révisé. Soit $E_t = t_1, t_2, \cdots$, l'ensemble des termes présents dans la définition (au sens large) du terme t. Deux cas sont alors possibles :
   (a) t_i a un vecteur associé $V(t_i)$, ce vecteur est alors utilisé ;
   (b) t_i n'a pas de vecteur associé. Un vecteur est tiré aléatoirement dans l'espace et est associé à t_i.
3. Calculer le vecteur de t en fonction de chacun des vecteurs des t_i (fonction d'agrégation) ;
4. Appliquer une fonction de séparation aux vecteurs de termes voisins de t ;
5. Recommencer à l'étape 2.

La fonction d'*agrégation* que nous avons utilisée dépend de la source d'information. Dans le cas, d'un apprentissage à partir de définitions de dictionnaire nous avons utilisé la même que celle pour les vecteurs conceptuels, à savoir une analyse de texte en remontée-descente. À partir des données du réseau JeuxDeMots, cela a consisté en une simple somme pondérée de vecteurs.

2.4 Évaluation des méthodes de construction de vecteurs

Nous avons mené une expérience d'évaluation de la qualité des vecteurs produits selon différentes méthodes. Choisir un protocole à la fois raisonnable et réaliste, dans le sens ou des utilisateurs accepteraient de s'y soumettre est loin d'être évident. De plus, n'avoir aucun corpus de référence en termes de vecteurs implique nécessairement une évaluation par satisfaction (directe ou indirecte) d'utilisateurs.

Notre choix s'est donc porté sur le comptage du nombre moyen de termes en commun dans le jeu JeuxDeMots (voir chapitre 3). Chaque méthode de construction correspond à un (ou plusieurs) joueurs virtuels - des *bots* (pour robot logiciel). Les propositions de ces bots sont confrontées aux propositions des vrais joueurs. L'hypothèse sous-jascente est que, plus le nombre moyen d'intersections entre les propositions d'un bot et celles des joueurs est élevée, meilleure sera la qualité des vecteurs. L'expérience a été menée sur les 1000 termes les plus fréquents du français ; il ne s'agit donc que de vocabulaire général. Pour chacun de ces termes, le bot produit 15 propositions qui sont les 15 termes les plus proches en termes de voisinage $\vartheta(V)$ du vecteur V du terme cible (dans le cas de la signature SJDM - cf. ci-dessous - il s'agit des 15 termes les plus activés).

Cette expérience a duré d'octobre 2008 à mars 2010, soit durant 18 mois. Chaque bot/méthode était sollicité à tour de rôle de façon à obtenir un nombre de parties jouées relativement équilibré. Les joueurs de JeuxDeMots ignoraient que des bots existaient, mais au vu de certaines propositions *farfelues*, quelques personnes ont parfois émis l'hypothèse que certains joueurs pouvaient en fait être virtuels. Un total d'environ 220 000 parties ont été jouées, soit une moyenne de 30 000 par méthode ou de 30 par terme pour chaque méthode.

Nous avons construit les vecteurs à évaluer selon les configurations suivantes :

- VCT : vecteurs conceptuels (taille 873) construits sur thésaurus/définitions de dictionnaires ;
- VET08 : vecteurs par émergence (taille 873) construits sur thésaurus/définitions de dictionnaires ;
- VET20 : vecteurs par émergence (taille 2000) construits sur thésaurus/définitions de dictionnaires ;
- VEJDM : vecteurs par émergence (taille 2000) construits sur les données de JDM ;
- VLSAC : vecteurs LSA (taille 400) construits sur corpus ;
- VLSAJDM : vecteurs LSA (taille 400) construits sur les données de JDM ;
- SJDM : signatures construites sur les données de JDM.

La méthode de construction pour VCT est celle indiquée au début de ce chapitre, à savoir des vecteurs conceptuels construits sur le thésaurus Larousse combiné à un corpus de définitions de dictionnaires. VET08 et VET20 sont des vecteurs construits par émergence (respectivement de taille 873 et 2000) à partir des mêmes données lexicales que précédemment. VEJDM sont des vecteurs de taille 2000 construits par émergence sur les données JDM (sur la relation *idées associées - association libre*). VLSAC sont des vecteurs construits avec LSA sur un corpus de taille moyenne (10 années du journal Le Monde, 1984-1994) et VLSAJDM sur les données de JDM. Enfin, SJDM sont les signatures lexicales construites directement à partir des données JDM.

Nous avons obtenu les résultats suivants :

	VCT	**VET08**	**VET20**	**VEJDM**	**VLSAC**	**VLSAJDM**	**SJDM**	**joueur**
score (μ)	2.1	2.4	2.7	3.2	1.2	2.7	3.5	3.2
σ	-	-	-	-	-	-	**1.2**	1.5

(note : les valeurs d'écart-type σ n'ont été calculées précisément que pour SJDM et entre les joueurs, mais pour les autres colonnes un calcul par échantillonnage donnait généralement $\sigma > 2$.)

La colonne *joueur* correspond à l'accord moyen entre joueurs (pour le même ensemble de termes). Le score est l'accord moyen entre le bot (proposant les termes issus des vecteurs calculés avec la méthode concernée) et les joueurs. Les différents bots sont confrontés uniquement à des joueurs humains (les bots ne sont jamais confrontés entre eux).

Comment interpréter ces résultats ? Dans un premier temps, considérons seulement le type de source d'information pour la construction des vecteurs : thésaurus + définitions de dictionnaires, corpus, et le réseau JeuxDeMots. La construction avec comme source le réseau JeuxDeMots est systématiquement meilleure, ce qui ne semble pas étonnant car les données n'y sont que peu ambiguës. Si les résultats ne sont comparés que sur le critère des méthodes utilisées, les signatures sont les plus performantes, suivies par les vecteurs construits par émergence. Pour la construction par émergence, plus la taille des vecteurs est grande, meilleurs sont les résultats. Notons aussi que l'écart-type (σ) est plus faible entre le bot SJDM (les signatures construites à partir le réseau lexical de JeuxDeMots) et les joueurs, que pour les joueurs entre eux.

Bien que l'expérimentation soit partielle, il est possible d'en déduire les propositions suivantes :

- La qualité des vecteurs construits est contravariante avec la difficulté d'interprétation de la source d'apprentissage, à savoir dans l'ordre : le réseau JeuxDeMots, les définitions de dictionnaires, un corpus de textes.
- Plus la taille des vecteurs est grande, meilleure est leur qualité, toutes choses étant égales par ailleurs. Cela semble être un résultat en contradiction avec l'argumentaire qui associe et justifie la réduction de dimension de vecteurs à la réduction du bruit. Un tel résultat semble être confirmé par [Gamallo & Bordag, 2011].

– le résultat concernant SJDM laisse penser que le bot répond comme un joueur médian, qui serait hypothétique centroïde parmi les joueurs réels. Les associations de JeuxDeMots sont l'agrégation des réponses produites par des êtres humains, donc il est relativement peu étonnant que les réponses produites sur cette base par le bot correspondent en moyenne à d'avantage de joueurs que les réponses d'un joueur réel particulier. Nous pensons que c'est plutôt une bonne nouvelle, le but du projet JeuxDeMots étant justement de construire une base lexicale correspondant, en moyenne, aux associations mentales des locuteurs.

Conclusion du chapitre 2

L'exploitation de définitions de dictionnaire en vue de la construction de vecteurs est délicate pour plusieurs raisons : (1) la présence de métalangage difficile à identifier, (2) la polysémie des termes de la définition, et (3) le caractère implicite de certaines informations difficiles à reconstituer automatiquement. L'exploitation de corpus de textes semble encore plus hasardeuse. La source d'informations la plus facile à exploiter reste donc un réseau lexical, car l'information y est explicite (à défaut d'être complète, voire pertinente). L'évaluation des méthodes de calculs de vecteurs reste difficile, à la fois à cause de la difficulté à disposer d'une donnée de référence et d'une fonction de comparaison raisonnable. Nous avons opté pour une évaluation indirecte via le jeu JeuxDeMots où est calculé un accord entre les propositions des joueurs et chaque vecteur. L'intérêt de cette méthode est qu'elle peut être effectuée au long cours, sur une part importante du lexique, et face à des évaluateurs implicites (les joueurs) nombreux et divers.

Articles adjoints au chapitre 2

M. Lafourcade *Conceptual Vector Learning - Comparing Bootstrapping from a Thesaurus or Induction by Emergence.* In proc. LREC'2006, Magazzini del Cotone Conference Center, Genoa, Italia, 24-26 May 2006, 4 p.

V. Prince, et M. Lafourcade *Mixing Semantic Networks and Conceptual Vectors - Application to Hyperonymy.* IEEE Transactions on Systems, Man, and Cybernetics : Part C. 2006, 11 p.

Conceptual Vector Learning
Comparing Bootstrapping from a Thesaurus or Induction by Emergence

Mathieu Lafourcade

LIRMM
161, rue ADA – 34392 MONTPELLIER Cedex 5
FRANCE.
lafourca@lirmm.fr

Abstract

In the framework of the Word Sense Disambiguation (WSD) and lexical transfer in Machine Translation (MT), the representation of word meanings is one critical issue. The conceptual vector model aims at representing thematic activations for chunks of text, lexical entries, up to whole documents. Roughly speaking, vectors are supposed to encode *ideas* associated to words or expressions. In this paper, we first expose the conceptual vectors model and the notions of semantic distance and contextualization between terms. Then, we present in details the text analysis process coupled with conceptual vectors, which is used in text classification, thematic analysis and vector learning. The question we focus on is whether a thesaurus is really needed and desirable for bootstrapping the learning. We conducted two experiments with and without a thesaurus and are exposing here some comparative results. Our contribution is that dimension distribution is done more regularly by an emergent procedure. In other words, the resources are more efficiently exploited with an emergent procedure than with a thesaurus terms (*concepts*) as listed in a thesaurus somehow relate to their importance in the language but not to their frequency in usage nor to their power of discrimination or *representativeness*.

1. Introduction

In the framework of the Word Sense Disambiguation (WSD) and lexical transfer in Machine Translation (MT), the representation of word meanings is one critical issue. The conceptual vector model aims at representing thematic activations for chunks of text, lexical entries, locutions up to whole documents. Roughly speaking, vectors are supposed to encode *ideas* associated to words or expressions. The main applications of the model are thematic text analysis and lexical disambiguation [Lafourcade 2001]. Such thematic representation is to be used together with more associative information like lexical networks. Conceptual vectors are more on the verge of improving *recall* than lexical networks which focus more on *precision*.

Practically, we have built a system, with automated learning capabilities, based on conceptual vectors and exploiting monolingual dictionaries (available on the web). So far, from French, the system learned around 145000 lexical entries corresponding to roughly 560000 vectors (the average meaning number for polysemous words being 5.3). We are conducting the same experiment for English. The issue of dimensionality in semantic space has been quite debated (see [Lowe 2000] for a theorization of those subjects), but some questions about the qualitative nature of the produced vector space are still largely untackled.

In this paper, we first expose the conceptual vectors model and the notions of semantic distance and contextualization between terms. Then, we present in details the text analysis process coupled with conceptual vectors, which is used (with very small adjustments) in text classification, thematic analysis and vector learning. The question we focus on is whether a thesaurus is really needed and desirable for bootstrapping the learning. We conducted two experiments with and without a thesaurus and are exposing here some comparative results. Our contribution is that dimension distribution is done more regularly by an emergent procedure. In other words, the resources (the vector components) are more efficiently exploited with an emergent procedure than with a thesaurus (this property seems to be independent of the thesaurus structure or concepts set). Key terms (*concepts*) as listed in a thesaurus somehow relate to their importance in the language (either general or of a specific domain), but not to their frequency in usage nor to their power of discrimination or *representativeness*. Corpora based approaches behave the other way, but do not explicitly point out semantic relations between word meanings.

2. Conceptual Vectors

We represent thematic aspects of textual segments (documents, paragraph, syntagms, etc) by conceptual vectors. Vectors have been used in information retrieval for long [Salton and MacGill 1983] and for meaning representation by the LSI model [Deerwester *et al.* 1990] from latent semantic analysis (LSA) studies in psycholinguistics. In computational linguistics, [Chauché 1990] proposes a formalism for the projection of the linguistic notion of semantic field in a vector space, from which our model is inspired.

From a set of elementary concepts, it is possible to build vectors (conceptual vectors) and to associate them to lexical items. Lexical items are words or expressions, which constitute lexical entries. For instance, *car* or *white ant* are lexical items. The hypothesis, we call *thesaurus hypothesis*, that considers a set of concepts as a generator to language has been long described in [Roget, 1852].

Polysemic words combine different vectors corresponding to different meanings. This vector approach

is based on known mathematical properties, thus it is possible to undertake well founded formal manipulations attached to reasonable linguistic interpretations.

Concepts are defined from a thesaurus (in our prototype applied to French, we have chosen [Larousse 2001] where 873 concepts are identified to be compared with the thousand defined in [Roget, 1852]).

To be consistent with the thesaurus hypothesis, we consider that this set constitutes a generator family for the words and their meanings. This family is probably not free (no proper vector base) and as such, any word would project its meaning on it according to the following principle.

Let be C a finite set of n concepts, a conceptual vector V is a linear combination of elements c_i of C. For a meaning A, a vector V(A) is the description (in extension) of activations of all concepts of C. For example, the different meanings of *door* could be projected on the following concepts (the concept of INTENSITY are ordered by decreasing values):

V(door) = OPENING {0.8}, BARRIER {0.7}, LIMIT {0.65}, PROXIMITY {0.6}, EXTERIOR {0.4}, INTERIOR {0.39}, ...

In practice, the larger C is, the finer the meaning descriptions are. In return, the computing is less easy: for dense vectors (which are those which have very few null coordinates - in practice, by construction, all vectors are dense) the enumeration of activated concepts is long and difficult to evaluate. We prefer to select the thematically closest terms, i.e., the neighborhood. For instance, the closest terms ordered by increasing distance to door are: V$(door) = portal, opening, gate, barrier, …

2.1. Angular Distance

Let us define *Sim(A,B)* as one of the *similarity* measures between two vectors A et B, often used in information retrieval . We can express this function as below with the " . " as the scalar product. We suppose here that vector components are positive or null. Then, we define an angular *distance* D_A between two vectors.

$$Sim(A,B) = \cos(\widehat{A,B}) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

$$D_A(A,B) = \arccos(Sim(A,B))$$

Intuitively, this function constitutes an evaluation of the *thematic proximity* and measures the angle between the two vectors. We would naively consider that, for a distance $D_A(A,B) < pi/4$ (45 degrees) A and B are thematically close and share many concepts. For $D_A(A,B) > pi/4$, the thematic proximity between A and B would be considered as loose. Around pi/2, they have no relation. D_A is a real distance function and it verifies the properties of reflexivity, symmetry and triangular inequality. In the following, we will speak of distance} only when these last properties will be verified, otherwise we will speak of measure. We have, for example, the following angles (values are in radian and degrees).

D_A(V(tit), V(tit))=0 (0)
D_A(V(tit), V(bird))=0.55 (31)
D_A(V(tit), V(sparrow))=0.35 (20)
D_A(V(tit), V(rain))=1.28 (73)
D_A(V(tit), V(insect))=0.57 (32)

The first one has a straightforward interpretation, as a *tit* cannot be closer to anything else than itself. The second and the third are not very surprising since a *tit* is a kind of *sparrow*, which is a kind of *bird*. A *tit* has not much in common with a *train*, which explains a large angle between them.

One can wonder why there is 32 degrees angle between *tit* and insect, which makes them rather close. If we scrutinize the definition of *tit* from which its vector is computed (*Insectivorous passerine bird with colorful feather.*) Perhaps the interpretation of these values seems clearer. In effect, the thematic is by no way an ontological distance.

A less naïve approach is to compare the actual angular distance to the mean distance over the vector space. This is a more practical comparison that is relative to the actual vector population. Anyway, the comparison function by itself has no influence on the conceptual vector construction.

2.2. Conceptual Vector Construction

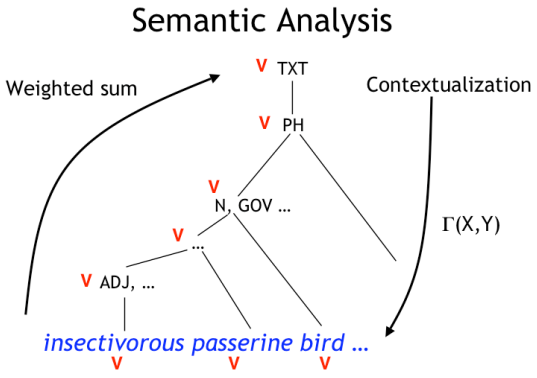
The conceptual vector construction is based on definitions from different sources (dictionaries, synonym lists, manual indexations, etc). Definitions are parsed and the corresponding conceptual vector is computed. This analysis method shapes, from existing conceptual vectors and definitions, new vectors.

It requires a bootstrap with a kernel composed of pre-computed vectors. This reduced set of initial vectors is manually indexed for the most frequent or difficult terms. It constitutes a relevant lexical items basis on which the learning can start and rely. One way to build a coherent learning system is to take care of the semantic relations between items. Then, after some fine and cyclic computation, we obtain a relevant conceptual vector basis. After 2 and half years (after starting in mid 1999), our system counted more than 71000 items for French and more than 288000 vectors, in which more 20000 items are concerned by relations, like antonymy, for example. These items are either defined through negative sentences, or because antonyms are directly in the dictionary. Example of a negative definition: *non-existence: property of what does not exist*. Example of a definition stating antonym: *love: antonyms: disgust, aversion.*

3. Semantic Text Analysis

The text analysis procedure based on conceptual vectors is independent of the underlying vector space. From a morphosyntactic analysis tree of the text, for each term sense (acception) we associate a vector. If the term is not present in the vector database, then the null vector is used instead. Vectors are then propagated upward and downward on the tree. The upward propagation produces merged vectors on the inner nodes of the tree. The downward propagation adjusts the vector of each node according by the context provided by the vectors of the

other nodes. This *weak contextualization* is by itself an exploitation of the mutual-information contained in vectors. When reaching a term node, each acception node is weighted non-linearly according to the context. The process is globally convergent, although in ambiguous text with several possible interpretation some vectors may oscillates between several states. *For* example, this is the case with typical sentences like *L'avocat est véreux* (Eng. *the lawyer is corrupted* or *The avocado is worm-eaten*) where both interpretations are equally reasonable (without further context). When applied to term definitions as found in dictionaries, this analysis leads to vector learning. The overall learning process is continuously iterated, each term definition and acception vector being



automatically revised periodically. The learning process converges globally in less than 10 cycles.

Fig 1 : Semantic analysis with a typical definition of tit as Insectivorous passerine bird with colorful feather

We have undertaken three main experiments. The first one (TH873) is based on the vector space defined in the french thesaurus Larousse, where 873 basic concepts are defined. For bootstrapping the learning process a kernel of roughly 1000 acceptions has been manually indexed on the basis of the thesaurus concepts. The second one (EM873) is done by emergence. No kernel, neither initial concept sets are then needed, but only the dimension of the vectors space is required (the dimension here has been set to 873 for having vector results directly comparable with the TH873 experiment). As no kernel is required, the bootstrapping is induced by randomly generating vectors for unknown terms. These vectors are going to be revised afterward. To keep computed vectors different (and not all converging to a common mean vector), we terminate the computation process of a given vector by an operation, called *amplification* that enhances the *contrast* of the vector. Basically, if the variation coefficient of a vector is extreme (too low or too high), then each component is non-linearly augmented (to a power value over 1) and the vector is then normalized. This process is applied repeatedly until the coefficient variation has a middle value. This process is directly inspired by what is done in photography to augment the contrast of dull pictures. We recognized here, that the most important properties to be achieved when learning vectors is both the coherence between acception vectors (and not their actual component activation) but also the discrimination between them. The third experiment (EM5000) is done by emergence with vector size of 5000.

4. Experiments and results

We found the following comparative results. In TH873 experiment, there is a strong precision induced by the finely crafted concept set issued from the thesaurus, but at the cost of a lack of information sharing. On the other hand, EM873 more evenly distributes the 873 vector components to represent very subtle meaning differences, especially in the vector space region where the lexical density is high. By emergence, the lexical density tends to be more uniform as more components tend to participate (than in TH873). In EM5000, being of a much larger size, vectors describe meanings with much more finesse but at a cost (in space and time). The increase in description is basically logarithmic with the increase in vector size. Globally, from a vector size of 873 to a vector size of 5000, the vector description is increased of (roughly) 33 percent. But, this gain is very significant to discriminate terms than where considered as (quasi) synonymous. For instance, in TH873, the *dragonfly* and *cockroach* have almost identical vectors, although in EM873, they remain quite close although being separable. In the EM5000 experiment, there are quite different and cannot anymore be considered as synonymous. Some assessment with vectors of size 10000 showed that the increase in description quality is negligible (less than 1% percent from EM5000) especially compared to the amount of size occupied.

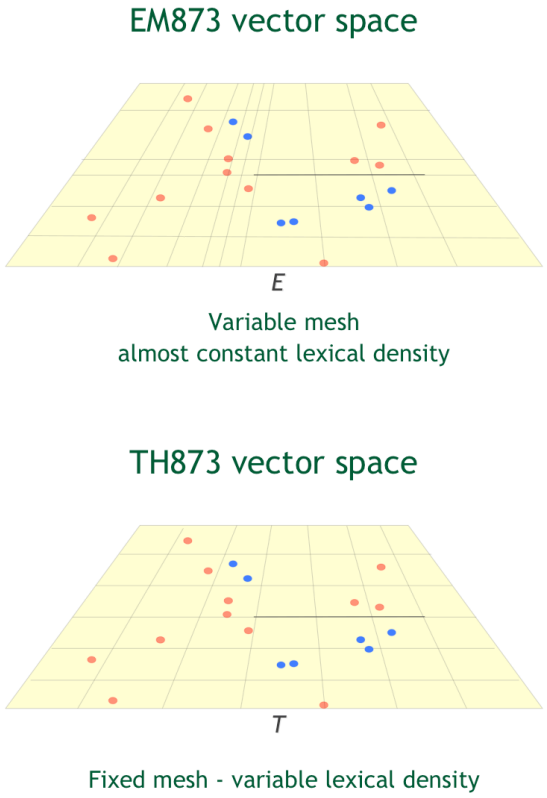
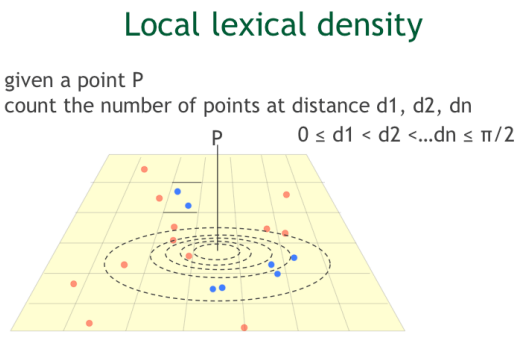
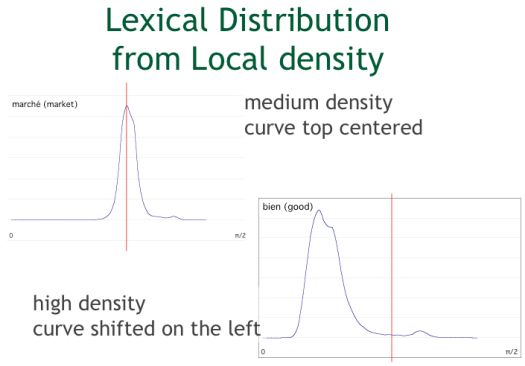


Fig 2 : Schema of a vector space with a fixed set of concepts and with a fixed number of concepts which are computed by emergence



Beside a manual evaluation of vectors, done by enumerating and assessing term neighborhood, some functions can globally assess vectors. For example, the evaluation of the lexical concentration gives clues about in increase of vector representation power (in the full paper those functions are detailed with equations). Our conclusion all in all, is that the higher the dimension the better the description both for separating terms that belong to close semantic fields but also for *relating* terms of different semantic fields but might share some relations that could prove being critical for semantic analysis. However, the best ratio between quality and vector size has to be precisely determined and, of course, may depend on application. Our experiments strongly suggest that a vector size around 5000 seems to be a good trade-off between finesse and space for word sense disambiguation



and indexation of general texts (like those found in newspapers). Results and lexical data (vectors) of some of our experiments are freely accessible at <http://www.lirmm.fr/~lafourcade> .

5. References

[Barrière and Copeck 2001] Barrière C., Copeck T., "Building Domain Knowledge from Specialized Texts", *TIA 2001*, Nancy, 2001.

[Chauché 1990] Chauché J., "Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance", *TA Information*, vol. 31, n_ 1, p. 17-24, 1990.

[Deerwester *et al.* 1990] Deerwester S., Dumais S., Landauer T., Furnas G., Harshman R., "Indexing by latent semantic analysis", *Journal of the American Society of Information Science*, 416(6), p. 391-407, 1990.

[Hearst 1998] Hearst M.A., "Automated discovery of Wordnet relations", In C. Fellbaum ed. *Wordnet An Electronic Lexical Database*, MIT Press, Cambridge, MA, p. 131-151, 1998.

[Lafourcade et al. 2001] Lafourcade M., Prince V. and D. Schwab, "Vecteurs conceptuels et structuration émergente de terminologie", *TAL*, vol 43 - n_ 1, p. 43-72, 2002.

[Lafourcade 2001] Lafourcade M., "Lexical sorting and lexical transfer by conceptual vectors", *First International Workshop on MultiMedia Annotation* (MMA'2001), Tokyo, 6 p, January 2001.

[Larousse 2001] Larousse, *Thesaurus Larousse - des idées aux mots - des mots aux idées*. Larousse, 1992.

[Lowe 2000] Lowe, W., "Towards a theory of semantic space", *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, Lawrence Erlbaum Associates. pp.576-581.

[Lowe 2000] Lowe,W., "Topographic Maps of Semantic Space", *PhD Thesis*, institute for Adaptive and Neural Computation, Division of Informatics, Edinburgh University, 2000.

[Resnik 1995] Resnik P., "Using Information contents to evaluate semantic similarity in a taxonomy", *IJCAI-95*, 1995.

[Riloff and Shepherd 1995] Riloff E., Shepherd J., "A corpus-based bootstrapping algorithm for Semi-Automated semantic lexicon construction", *Natural Language Engineering*, vol. 5, part. 2, p. 147-156, 1995.

[Roget, 1852] *Thesaurus of English Words and Phrases.* Longman, London, 1852.

[Salton and MacGill 1983] Salton G., MacGill M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

[Salton 1988] Salton G., *Term-Weighting Approaches in Automatic Text Retrieval*, *McGraw-Hill computer science series*, McGraw-Hill, vol. 24, 1988.

[Schwab *et al.* 2002] Schwab D., Lafourcade M., V. Prince, "Antonymy and conceptual vectors.", In proc. of *COLING'2002*, Taipei, Taiwan, August 2002.

[Sparck Jones 1986] Sparck Jones K., *Synonymy and Semantic Classification*, Edinburgh Information Technology Serie, 1986.

[Ploux et Victorri 1998] Ploux S., Victorri B., "Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes." *TAL*, vol. 39, n_ 1, p. 161-182, 1998.

[Yarowsky1992] Yarowsky D., "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora", *COLING'92*, Nantes, p. 454-460, 1992.

Mixing Semantic Networks and Conceptual Vectors Application to Hyperonymy

Violaine Prince
LIRMM-CNRS and University Montpellier 2
161 rue Ada, 34392 Montpellier cedex 5
France
prince@lirmm.fr

Mathieu Lafourcade
LIRMM-CNRS and University Montpellier 2
161 rue Ada, 34392 Montpellier cedex 5
France
lafourcade@lirmm.fr

Abstract

In this paper, we focus on lexical semantics, a key issue in Natural Language Processing (NLP) that tends to converge with conceptual Knowledge Representation (KR) and ontologies. When ontological representation is needed, hyperonymy, the closest approximation to the is-a *relation, is at stake. In this paper we describe the principles of our vector model (CVM: Conceptual Vector Model), and show how to account for hyperonymy within the vector-based frame for semantics. We show how hyperonymy diverges from* is-a *and what measures are more accurate for hyperonymy representation. Our demonstration results in initiating a 'cooperation' process between semantic networks and conceptual vectors. Text automatic rewriting or enhancing, ontology mapping with natural language expressions, are examples of applications that can be derived from the functions we define in this paper.* ***Keywords****: knowledge representation, cognitive linguistics, natural language processing.*

1 Introduction

Natural Language Processing by machines (NLP) has long been a keystone for the branch of data processing that deals with Knowledge Representation (KR) and Artificial Intelligence (AI). Since language stands, for human beings, both as a formalism describing knowledge, and their favourite mean of communication, NLP has, for decades, acted as the test for intelligent processing. It is a NLP function that underlies the Turing test, i.e. the ability of mimicking humans in their means of communication. Thus, it is easy to show that NLP is one of the most fundamental topics in Cognitive Informatics.

Since the nineties, with the generalization of the world wide web, a new challenge bursted out, to be tackled by NLP researchers. A huge amount of textual data is now available to users, data they need to browse, understand, summarize and exchange. Therefore, to the problem of intelligence in communication, a new issue has been added to NLP topics: how to deal with important volumes of texts, that human users do not have the time or the power to analyze. New trends, arising from fields such as Information Retrieval (IR) and documents design, are now investigated by NLP techniques.

Within the wide NLP domain, *lexical semantics* are a key issue, since they represent the point of convergence with conceptual KR and ontologies extracted from web semantics. They also browse the area of lexical resources processing, so that many works in both NLP and AI have been devoted to lexical semantic functions, as a way to tackle the problem of word sense representation and discrimination. Among the well established trends in lexical semantics representations, two trends appeared to be conflictual, until now: the WordNet approach [13], [4], born from semantic networks, and KR-oriented, and the "vector approach", originated from the Saltonian representation in Information Retrieval (IR) [19], which has found a set of applications in NLP, especially with web semantics and documents design.

The first is based on logic and the second on vector-space algebra. The first is very efficient for *is-a* relationships (considered as the conceptual relation often embedded in hyperonymy) but is silent, or almost so, about several other interesting lexical functions such as antonymy [1] and thematic association[2]. Synonymy has been tackled by NLP researchers that enhanced the field of textual IR [21], [13], but discrimination between synonymy and hyperonymy has often led them to look for a more flexible notion such as semantic similarity [16].

The vector approach is completely at the opposite. Of-

[1] the opposition semantic relation. Example : 'big' and 'small' are related with antonymy. But so are 'moon' and 'sun' although they share many common traits.

[2] thematic association is often a 'loose' association of words or items belonging to the same topic, whatever the type of the relation.

fering very easily thematic association, it allows several distinct, fine-grained synonymy [8] and antonymy [22] functions to be defined and implemented, but is unable to differentiate or to valid the existence of hyperonymous relations.

In this paper, we show how to account for hyperonymy within the vector-based frame for semantics, relying on a cooperation between semantic networks and conceptual vectors, and how this can be applied to new functions such as word substitution, and semantic approximation, that belong to the field of semantic similarity. We use a semantic network to enhance vector learning, and symmetrically we build customized semantic networks out of hyperonymous relations between vectors. Experiments have been run on French, since our team owns a syntactic parser, and a semantic vectors producer for this language. For the time being, more than $200,000$ terms (words and expressions) are present in our lexical bases, and are regularly processed and tested with every tool we develop [3]. Of course, since methods are generic, they could be easily transposed to any language for which syntactic parsing and semantic vectors are provided[4]. Presenting and discussing our tool for hyperonymy is thus an important issue not only for this lexical base enhancement, but also for all applications that are derivable from semantic associations in texts.

2 Hyperonymy and *is-a* Relations

2.1 Defining Hyperonymy

Hyperonymy is a lexical function that, given a term t, associates to t one or many other terms that are more general, such as those used to define t in *genus* and *differentiae* (in the aristotelian definition). Its symmetrical function is called *hyponymy*. For instance, *bird* is a hyperonym for *sparrow, tit,eagle* and so forth. The latter are co-hyponyms of *bird*.

Hyperonymy, in almost all KR papers, is assimilated to the general argument of the *is-a* relationship (fundamentals are given in [1]). Let us remind that the *is-a* relationship is such as if X is a class of objects, and X' a subclass of X, then $is - a(X', X)$ is true. The rightmost argument X is called the *general* argument whereas X' is said to be the *specific* argument. The problem is that linguistic hyperonymy is not a "pure" *is-a* relation. When the word *horse* is defined, we find: "a herbivorous animal, with four legs, etc...". A good hyperonym for this definition of *horse* is *herbivorous mammal*. *Animal* is another hyperonym, since '*herbivorous mammal is-a mammal* and *mammal is-a animal*' is true. However, thematically, a *horse* is very close to a *herbivore*, whereas *herbivores* do not constitute a class but a set of individuals that may belong to different lines of the taxonomy (birds and insects and reptiles could be herbivorous, but also metaphorically, many other things). Thus, even if, in language, one wants to write that *a horse is a herbivore* eventhough *horse is-a herbivore* is false.

[3] our French lexical base and different tools provided for thematic association are all gathered at the following URL : http://www.lirmm.fr/∼ lafourca.

[4] for English, Roget-based vector representations are definitely adequate.

2.2 Some Specific Linguistic Issues Related with Hyperonymy

Linguistically, a *mammal* is not as good a hyperonym as *herbivorous mammal* for *horse*, because it is too vague. Too many mammals exist, and thus, the more precise the term, the better it is. *Mammal from the equine family* is precise but non informative to the plain user. If IR is stake, one would better be close to the language that is generally used. Thus, *herbivorous mammal* could appear as a trade-off. However, this can 'break' the *is-a* chain, because other relations can be mixed with the general argument. Here *herbivorous* acts as an attribute. But in itself, as a language item, *herbivores* exist as the set name of all animates that share this property. The status of the *attribution* relation is not well defined in all KR-derived models. In fact, attributes are termed as such as the result of the designer decision, and not because of their intrinsic properties.

In short, hyperonymy often appears as a complex function resulting from the composition of *is-a* and *is-attribute* relations, the latter originally present in the semantic networks model, but being abandoned by several formalisms, because of their ambiguous status.

The second linguistic problem is *polysemy*. A word is not a concept, it may address many concepts, and in many different ways with different intensities. A *horse* is:

- an animal
- a power unit for motors
- a mean of transportation.

The three 'points of view' over *horse* are not independent from each other. Historically, the animal has been ridden by humans and served as a mean of transportation. When shifting to mechanical devices, people needed to compare artificial modes of transportation and their original mean. Thus, they used the *horse* as a power unit as *candles* have been used as a mean of comparison for light intensity.

2.3 WordNet and Hyperonymy: How KR Tackles Linguistic Issues

WordNet is a built taxonomy of words, and as such, only captures *is-a* relations. Polysemous words having many definitions, and thus many hyperonyms, are tied with as

many *is-a* relations, which explains why WordNet is a network and not a tree. WordNet discards specific relations, and adresses polysemy only through the modelling of multiple inheritance in *is-a* chains: every step of the chain of classes and subclasses must verify the order relation. As language has not the same density of items everywhere, WordNet appears as a network with a certain amount of *gaps* in some locations and a fine-grained mesh in other places. For instance, the closer to the 'root', the more vague and scarce the words are. This property is important because, unlike local ontologies that are balanced in their densities, WordNet is closer to the core of problems that NLP has to deal with. Vagueness in IR, as well as in indexation, could be a very bad feature.

2.4 Hyperonymy and Word Definition

As shown before, hyperonyms could be extracted, when they are not known, from most dictionary-like definitions. Only general concepts, which tend to play the role of hyperonyms (and *is-a*) superclasses of many others, are not defined through aristotelian definition, but are explained by their hyponyms. This is why, in our CVM (Conceptual Vector Model) model presented in next section, we consider the existence of a "hyperonymy horizon" beyond which definitions become inversed: hyperonyms are more difficult to find and less explicative than hyponyms. The word *action* is almost at the top of the WordNet taxonomy and dictionary definitions tend to explain it with more specific words.

3 The Conceptual Vector Model (CVM)

Vectors have been used in Information Retrieval for long [20] and for meaning representation by the LSI model [3] from latent semantic analysis (LSA) studies in psycholinguistics. In NLP, and in the early nineties, [2] has provided a formalism for the projection of the linguistic notion of *semantic field* in a vector space, from which our model is inspired.

From a set of elementary notions, *concepts*, it is possible to build vectors (conceptual vectors) and to associate them to lexical items.[5] The hypothesis that considers a set of concepts as a generator to language has been long described in the Roget Thesaurus designed by Oxfordian Lexicologists at the end of the 19th century [18] (we call it the *thesaurus hypothesis*) and has been used by researchers in NLP (e.g. [23]) recently. Polysemous words combine different vectors corresponding to different meanings. This vector approach is based on well known mathematical properties: it is thus possible to undertake formal manipulations attached to reasonable linguistic interpretations. Concepts are defined within a thesaurus (in our prototype applied to French, we have chosen [10] where 873 concepts are identified to compare with the 1043 provided by the Roget Thesaurus [18]). To be consistent with the thesaurus hypothesis, we consider that this set constitutes a generator 'family' for words and their meanings. This set is probably not free (no proper vectorial base)[6] and as such, any word would project its meaning on this space according to the following principle.

3.1 Principle

Let be $\mathcal{C}$ a finite set of n concepts, a conceptual vector V is a linear combination of elements c_i of $\mathcal{C}$. For a meaning A, a vector $V(A)$ is the description (in extension) of activations of all concepts of $\mathcal{C}$. For example, the different meanings of ‹*door*› could be projected on the following concepts (the set of pairs (*CONCEPT*[intensity]) are ordered by increasing values): V(‹*door*›) = (*OPENING*[0.3], *BARRIER*[0.31], *LIMIT*[0.32], *PROXIMITY*[0.33], *EXTERIOR*[0.35], *INTERIOR*[0.37], ...)

In practice, the largest $\mathcal{C}$ is, the finer the meaning descriptions are. In return, computer manipulation is less easy. As most vectors are dense (very few null coordinates), the enumeration of activated concepts is long and difficult to evaluate. We generally prefer to select the thematically closest terms, i.e., the *neighbourhood*. For instance, the closest terms ordered by increasing distance of ‹*door*› are: $\mathcal{V}$(‹*door*›)=‹*portal*›, ‹*portiere*›, ‹*opening*›, ‹*gate*›, ‹*barrier*›,...

To handle semantics within this vector frame, we use the common operations on vectors. An interesting measure is the angular distance that accounts for a *similarity measure*. As an example, we present, hereafter, the vector sum, the scalar product and the angular distance equations.

3.1.1 Vectors Sum

Let A and B be two vectors, we define V as their *normed sum*:

$$V = X \oplus Y \quad | \quad v_i = (x_i + y_i)/\|V\| \tag{1}$$

Intuitively, the vector sum of A and B corresponds to the union of semantic properties of A and B. This operator is idempotent as we have $A \oplus A = A$. The null vector $\vec{\mathbf{0}}$ is a neutral element of the vector sum and, by definition, we have $\vec{\mathbf{0}} \oplus \vec{\mathbf{0}} = \vec{\mathbf{0}}$.

[5] Lexical items are words or expressions which constitute lexical entries. For instance, ‹*car*› or ‹*white ant*› are lexical items. In the following we will sometimes use *word* or *term* to speak about a *lexical item*.

[6] Let us remind that a vectorial base is a set of generative and free vectors. Two vectors are said to be free if their vector product is equal to zero. A set of vectors is considered free, if each couple of vectors contained in it, is free.

3.1.2 Vectors Product

The vector product is equivalent to a *normed term to term product*. Let X and Y be two vectors, we define V as *their normed term to term product*:

$$V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{2}$$

This operator is idempotent and $\vec{0}$ is absorbent.

$$V = X \otimes X = X \quad \text{and} \quad V = X \otimes \vec{0} = \vec{0} \tag{3}$$

Also following an intuitive approach, the vector product of A and B represents the intersection of semantic properties of A and B. This is a crucial feature for hyperonymy since a hyperonym and its hyponym could be seen as one 'containing' the properties of the other. But it is also important in synonymy and may give hints about polysemous properties of some conceptual vectors (intersections with many different vectors). A better function for emphasizing intersection is given in the paragraph about contextualization.

3.1.3 Angular Distance

Let us define $Sim(A, B)$ as one of the *similarity* measures between two vectors A and B, often used in Information Retrieval. We can express this function as:

$$Sim(A, B) = \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

with "$\cdot$" as the scalar product. We suppose here that vector components are positive or null. Then, we define an *angular distance* D_A between two vectors A and B as follows:

$$\begin{aligned} D_A(A, B) &= \arccos(Sim(A, B)) \\ \text{with} \qquad Sim(A, B) &= \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|} \end{aligned} \tag{4}$$

This function constitutes an evaluation of the *thematic proximity* as it measures the angle between the two vectors. We would generally consider that, for an angular distance $D_A(A, B) \leq \frac{\pi}{4}$, (i.e. less than 45 degrees), A and B are thematically close and share many concepts. For $D_A(A, B) \geq \frac{\pi}{4}$, the thematic proximity between A and B would be considered as loose. Around $\frac{\pi}{2}$, both vectors are othogonal, and thus tend to diverge very wildly. D_A is a real distance function. It verifies the properties of reflexivity, symmetry and triangular inequality. In the following, we will speak of *distance* only when these last properties will be verified, otherwise we will speak of *measure*.

3.1.4 Contextualization

When two terms are in presence of each other, some of the meanings of each of them are thus selected by the presence of the other, acting as a *context*. This phenomenon is called *contextualization*. It consists in emphasizing common features of every meaning. Let X and Y be two vectors, we define $\gamma(X, Y)$ as the contextualization of X by Y as:

$$\gamma(X, Y) = X \oplus (X \otimes Y) \tag{5}$$

This function is not symmetrical, translating the non symmetry between the role of a context and the role of a contextualized term. As for other mathematical properties: the operator γ is idempotent ($\gamma(X, X) = X$) and the null vector is the neutral element. ($\gamma(X, \vec{0}) = X \oplus \vec{0} = X$). We will notice, without demonstration, that we have thus the following properties of *closeness* and of *farness*:

$$\begin{aligned} & D_A(\gamma(X, Y), \gamma(Y, X)) \\ \leq & \{D_A(X, \gamma(Y, X)), D_A(\gamma(X, Y), Y)\} \\ \leq & D_A(X, Y) \end{aligned} \tag{6}$$

The function $\gamma(X, Y)$ brings the vector X closer to Y proportionally to their intersection. The contextualization is a low-cost meaning of amplifying properties that are salient in a given context. For a polysemous word vector, if the context vector is relevant, one of the possible meanings is *activated* through contextualization. For example, *bank* by itself is ambiguous and its vector is pointing somewhere between those of *river bank* and *money institution*. If the vector of *bank* is contextualized by *river*, then concepts related to finance would considerably dim.

3.2 Implemented Lexical Functions: Synonymy and Antonymy

3.2.1 Synonymy

Two lexical items are in a synonymy relation if there is a semantic equivalence between them.

Synonymy is a pivot relation in NLP, but remains problematic, since semantic equivalence is not translatable into a mathematical equivalence relationship. It does not necessarily verify transitivity [12] and it could be, at least partially, confused with hyperonymy, when equivalence is reduced to semantic similarity [16]. A possible solution in a vector framework is to define a contextual synonymy (also proposed in [6]) represented by a three argument relation, which then supports the properties of an equivalence relationship. The suggested solution is called *relative synonymy* [8]. The functional representation is the following: a *relative synonymy* function Syn_R, is defined between vectors A, B and C, the later playing the role of a pivot, as:

$$\begin{aligned} Syn_R(A, B, C) &= D_A(\gamma(A, C), \gamma(B, C)) \\ &= D_A(A \oplus (A \otimes C), B \oplus (B \otimes C)) \end{aligned} \tag{7}$$

The interpretation corresponds to testing the thematic closeness of two meanings (A and B), each one enhanced with what it has in common with a third (C). The advantage of such a solution is that it circumvents the effects of polysemy in cutting transitivity and symmetry. However, it does not provide a real distinction between a hyperonym of a given meaning of a word, and a true synonym of such a word. This problem is discussed in next section, when introducing more flexible notions such as *word substitution*.

3.2.2 Antonymy

Two lexical items are in antonymy relation if there is a symmetry between their semantic components relatively to an axis.

Three types of symmetry have been defined, inspired from linguistic research [14]. As an example, we expose only the 'complementary' antonymy proposed by [22]: The same method is used for the other types. *Complementary antonyms* are couples like *event/unevent, presence/absence*. Complementary antonymy presents two kinds of symmetry, (i) a value symmetry in a boolean system, as in the examples above, and (ii) a symmetry about the application of a property (*black* is the absence of color, so it is "opposed" to all other colors or color combinations). The functional representation is the following: The function $AntiLex_S$ returns the *n* closest antonyms of A in the context defined by C in reference to R. The partial function $AntiLex_R$ has been defined to take care of the fact that, in most cases, context is enough to determine a symmetry axis. $AntiLex_B$ is defined to yield a symmetry axis rather than a context. In practice, we have $AntiLex_B = AntiLex_R$. The last function is the *absolute antonymy function*. Their associated equations are given hereafter.

$$\begin{aligned} A, C, R, n &\rightarrow AntiLex_S(A, C, R, n) \\ A, X, n \rightarrow AntiLex_R(A, X, n) &= AntiLex_S(A, X, X, n) \\ \text{with} \quad X &= (C|R) \\ A, n \rightarrow AntiLex_A(A, n) &= AntiLex_S(A, A, A, n) \end{aligned} \tag{8}$$

An implementation of these functions in the CVM is detailed and commented in [22]. Contrarily to synonymy, antonymy functions are modelled partially as semantic graphs and partially with conceptual vectors. Some oppositions are primarily of lexical nature, and can potentially be extended continuously in the meaning space.

3.3 Conceptual Vectors Construction

Building conceptual vectors is achieved through processing *definitions* from different sources (dictionaries, synonym lists, manual indexations, etc). Definitions are parsed with an NLP parser called SYGMART (available for French) and the corresponding conceptual vector is computed according to a procedure defined as follows.

After filtering according to various morphosyntactic attributes, we attach to the leaf (terminal node of the conceptual tree) a conceptual vector that is computed from the vectors of its k definitions. The most straighforward way (not the best) to do so is to compute the average vector: $V(w) = V(w.1) \oplus \cdots \oplus V(w.k)$. If the word is unknown (i.e. it is not in the dictionary), the null vector is taken instead.

Vectors are then propagated upward. Consider a tree node N with p dependants $N_i (1 \leq ip)$. The newly computed vector of N is the weighted sum of all vectors of N_i: $V(N) = \alpha_i N_1 \oplus \cdots \oplus \alpha_p N_p$. Weights α depend on the syntactic functions of the node. For instance, a *governor* [7] would be given a higher weight ($\alpha = 2$) than a regular node ($\alpha = 1$).The vectors computed for *a boat sail* and for *a sail boat* would not be identical. Once the vector of the tree root is computed a downward propagation is performed. A node vector is contextualized by its parent: $V'(N_i) = V(N_i) \oplus \gamma(N_i, N)$. This is done iteratively until reaching a leaf. This analysis method shapes, from existing conceptual vectors and definitions, new vectors. It requires a bootstrap with a kernel composed of pre-computed vectors, manually indexed for the most frequent or difficult terms and already defined in [10] . One way to build a coherent learning system is to take care of the semantic relations beetween items, and among them, synonymy, antonymy and the most important, hyperonymy. A relevant conceptual vector basis is obtained after some iterations in the learning process. At the moment of writing this article, our system counts more than $71,000$ items for French and more than $288,000$ vectors (because vectors may represent expressions and/or concepts). 2000 vectors are concerned with antonymy, and almost all of them are concerned with synonymy and hyperonymy. The computed functions have allowed to enhance the representation of almost all vectors.

3.4 Importance of Hyperonymy in CVM

A framework for hyperonymy is very useful for enhancing vector construction, since most vectors are built by parsing hyperonymous definitions provided by on-line sources on the Web. In fact, all lexical functions appear to be a great help for such as task. Symmetrically, relations between vectors are crucial for a data driven approach : trying to extract semantic relations in corpora ([23]) and thus building a domain ontology, or trying to organize information in corpora by relying upon *is-a* hierarchies ([11], [17]).

[7] the 'leader' in a syntactic group. For instance, subjects and verbs in a sentence are governors, whereas complements are definitely not. In a noun phrase, one of the nouns is a governor, and the other is a subordinate. Example : in the noun phrase 'grammar school' , 'school' is governor.

4 Computing Hyperonymy

As our approach is both data driven and hierarchy-based, we first try to define the impact of hyperonymy by measuring distances in corpora. These distances help to define *word substitution* and *semantic approximation* (with a taxonomical aspect). The theoretical model, both within semantic networks and vector space, is the *inclusion model*: a subclass includes the properties of its superclass. We show in this section how inclusion is dealt with and what results we have obtained.

4.1 Co-occurrence Model

Corpora are seen by researchers in NLP as set of real instantiations of linguistic phenomena, when compared to intentionally built toy sentences. The co-occurrence of items, either words or expressions, especially when it is repeated through a rich set of documents, is a good measure of a semantic relationship between these items [5]. This semantic relationship is sometimes assumed to be one of synonymy, closeness, but without a strict and rigorous linguistic definition. The Church's formula tends, however, to consider co-occurring items in a given string of words, and to rely on the frequency of this co-appearance to draw probabilities of relationship. What we suggest here, is to consider documents (and not pairs of items) as the unit measure, and a single co-occurrence in a document is as meaningful as repeated associations of the same items.

Thus, we define two measures of co-occurrence between a term w and an *hyperonym candidate* h:

$$M_T(w,h) = \frac{|H \cap W|}{|W|} \quad \text{and} \quad M_S(w,h) = \frac{|H \cap W|}{|H|} \tag{9}$$

W (resp. H) represents the set of documents in a given corpus that contains the term w (resp. h). $|W|$, respectively $|H|$, is the number of documents considered where w, respectively h appears. $|H \cap W|$ represents the set of documents that contains both terms h and w. M_T tends to determine the ratio of h and w co-occurrence as a pair, when compared to w. So if w is the reference element, and W is the relevant set of documents about w, then M_T tends to show how much of w meaning is available when using h, knowing that w and h do (or not) co-occur in texts. M_T is reminiscent of a *recall* measure in Information Retrieval. [8]

M_S on the contrary, relates the same numerator, with the number of documents containing h. So if h is a possible, but polysemous, hyperonym of w, or if h was scarcely related to w then $|H \cap W|$ would be small when compared to $|H|$, and M_S would define thus the relevance of replacing w by h, without bringing in irrelevant meanings or ideas. M_S is thus our realization of a *precision* measure. [9]

M_T and M_S are in an inverse relationship, but are neither symmetrical nor complementary. It is more a question of a trend.

4.1.1 Hyperonymy, Word Substitution, Taxonomy Evaluation

If we add the hypothesis that *h is possible hyperonym*, that is, we have good reasons to think that w *is-a* h is true, then the measure M_S evaluates to which extend w can be replaced by h and is thus a *word substitution measure*. Similarily, M_T is a taxonomy evaluation, the way one can approximate *horse* by *mammal* without being too vague.

We have run experiments by accessing Google *(www.google.com)* and the number of hits returned for each request. This number of hits corresponds to the cardinal of the considered set of documents. For example, we have the following result for the term *airplane*:

aircraft /$M_T = 0.2659 \quad M_S = 0.025$
plane /$M_T = 0.1237 \quad M_S = 0.1741$
flying plane /$M_T = 0.5317 \quad M_S = 0.0007$
aircraft heavier than air /$M_T = 0.5238 \quad M_S = 0.00004$

The best M_S value (when *airplane* is the reference) is for *plane*, however, it is small, probably because of the embedded polysemy in the term (it also means a flat world, a two dimensional mathematical space, ...). In the general context of documents accessed by Google, people tend to use *plane* instead of *airplane*, when they exactly know what type of item they are talking about. However it has the worst value in the taxonomical evaluation: among the relevant hyperonym candidates, any other is more relevant than *plane*.

On the other side, *aircraft heavier than air* as well as *flying plane* have the best M_T or recall value. In fact, they are very good definitions or explanations of what is an *airplane*, eventhough people tend not to use them much as substitutes. This might appear strange, at least for *flying plane*: we interpret this absence of substitution frequency as the result of an economy principle that underlies most cognitive actions. If one undergoes the replacement of something by something else, one hopes at least to gain some cognitive effort. A shorter form as a substitution candidate is a good heuristic.

[8] Recall is the number of relevant items retrieved among the relevant records/documents present in the set of records/documents.

[9] Precision is the number of relevant items retrieved among the most exhaustive set of records/documents, where some are relevant and the others, not.

As a larger example, we have run the test for the term *horse*. We have found several meanings for *horses*:

- (a) the animal,
- (b) the class of horses or specie,
- (c) horse riding,
- (d) the representation of a horse,
- (e) the wooden horse,
- (f) the manlike women,
- (g) the power unit
- (h) an unreliable person
- ...

The results of requests and co-occurrence measures are :

mammal / $M_T = 0.81$ $M_S = 0.0005$ (a)
animal / $M_T = 0.0986$ $M_S = 0.1523$ (a)
domestic animal / $M_T = 0.133$ $M_S = 0.0035$ (a)
kind of mammal / $M_T = 0.0481$ $M_S = 0.00002$ (a)
specie / $M_T = 0.1376$ $M_S = 0.0857$ (b)
horses / $M_T = 0.4673$ $M_S = 0.2954$ (b)
equitation / $M_T = 0.3498$ $M_S = 0.0991$ (c)
representation / $M_T = 0.0399$ $M_S = 0.0505$ (d)
toy / $M_T = 0.1363$ $M_S = 0.0184$ (e)
child toy / $M_T = 0.2387$ $M_S = 0.0004$ (e)
wooden horse / $M_T = 0.2025$ $M_S = 0.0012$ (e)
woman / $M_T = 0.0363$ $M_S = 0.4012$ (f)
manlike woman / $M_T = 0.5692$ $M_S = 0.00003$ (f) unit / $M_T = 0.033$ $M_S = 0.0647$ (g)
arbitrary unit / $M_T = 0.067$ $M_S = 0.00004$ (g)
power unit / $M_T = 0.1042$ $M_S = 0.0003$ (g)

mammal is the most precise for the taxonomy (hyperonym used in definition) but *animal* is a better susbsitution term, eventhough it might not be a very good substitute (M_S around 15%). *specie* is too vague, when compared to *horses*. *child toy* has a best rendering of the meaning in item *(e)* than *toy* but is not as good as a substitute.

As we have noticed before, short terms are better substitutes, as representatives of the economy principle in linguistics. Taken out of their context, they might appear, from a taxonomic point of view, quite vague or ambiguous. However, since they are never isolated, their role as substitutes is not overburdened by polysemy.

Let us finally notice that if M_T values might sometimes come close to 0.8, this is never the case with M_S. Ratios for substitution continue to be very small. We have run the same experiments of many other words, and we have noticed the same difference in scale between the two measures.

4.1.2 Building and Upgrading a Local Possible *is-a* Hierarchy

A good M_T measure for a possible hyperonym helps to create a local *is-a* hierarchy by testing values from the most particular item up to the most general one. For instance, for *horse*, we can extract, directly from the text, the knowledge as a *horse is-a* a *mammal* is better than a *horse is-a* an *animal* on the taxonomical line. Since we can calculate and show that a *mammal is-a* an *animal* is true, then it is easy to create the following *is-a* line :
horse $\leq$ *mammal* $\leq$ *animal*
where $\leq$ represents an *is-a* relationship.
However, these different lines have to be merged, and moreover, sometimes, new meanings (unknown or not encountered before) have to be added to the existing structure, transforming it from a tree-like hierarchy into a plain graph. This graph plays the role of an *extracted semantic network*, at least one that has emerged from raw texts, vector forms and nothing else. Figure 1 shows a portion of the semantic network for *horse*.

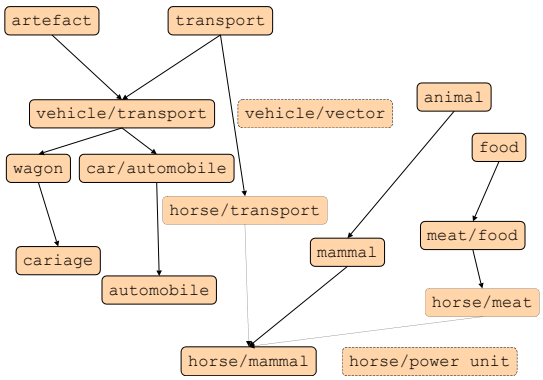


Figure 1. Hyperonym insertion in the built semantic network. Adding found hyperonyms can lead to the identification either of: (1) new salient properties in already existing meanings or (2)new meanings altogether. Thematic distance is used as a meaning selector.

About new meanings, in fact, two at least are lacking in the list of item given before.

- (i) a transportation mean (*we travelled on horseback*)
- (j) a type of food (horsemeat)

In this case, we do create the new meanings (*horse/ transportation mean* and *horse/meat*) and link them to their hyperonyms. The problem is that, starting from vectorized definitions, there is no way to catch these new meanings as they are not (yet) identified. Thus, to overcome this problem, we link each of these new meanings as hyperonym to

its closest already existing counterpart. In the above example, we have:

- *horse/ transportation mean* is closer to *horse/mammal* than to *horse/power unit*. This relation can be checked on their respective vector, and (sometimes) by pattern matching on some part of (encyclopedic) definition.
- *horse/meat* is closer to *horse/mammal* than to *horse/power unit*.

4.1.3 Conclusion about the Co-occurrence Model

These two measures, M_T and M_S, are particularly useful in semantic analysis. In fact, building a lexical network on the basis of M_T and M_S allows to recognize loose substitution hyperonyms (low M_T and high M_S). For example, during analysis, we can detect that the text thematic coherence is much stronger when we (re)subsitute *aircraft* to *plane*. Candidates for substitution are determined by the network structure strengthened by the angular distance between the candidate and the context. It is an iterated process that is globally converging ([9]). Thus, for textual analysis, we process in the reverse way of the text author, who has replaced precise terms with more or less vague hyperonyms, motivated by stylistic considerations (for example, deleting repetitions).

4.2 Inclusion Model

Inclusion, as a general idea, is what appears as common to both semantic networks in KR, and vector modelling in NLP when dealing hyperonymy. It is derived from a set theory approach, and suggests the following ;

If A is an hyperonym of B, then the properties of A are included in the properties B.
In KR, this means that A and B are in a super/subclass relationship (classical *is-a*). However, another definition also appears :
A is an hyperonym of B, if B has the same properties than A, and if B properties are instances of A properties
Examples:
'to cut' is a hyperonym of *'to saw'*. The latter provides the value of the action instrument (here the *saw*).
horses as the generic value of the specie, is a hyperonym of *horse* the individual (element (b) in the list of meanings for *horse*).

In KR this assets a set-member relationship (classical $member - of$), where the properties of A are instantiated by values belonging to the description of B.

As seen here, in fact, if KR tends to consider *is-a* hyperonyms only, unfortunately, NLP, at least in corpora, tends to consider also the $member - of$ relationship as a clue to a hyperonymy-hyponymy relationship. In fact, this is one of the cases where hyperonymy and hyponymy are symmetrical. In usage, if *to cut* acts as a good explanation of *to saw* the other way round is not true.

Thus, only in a restricted approach, the *is-a* and $member - of$ hyper/hyponymies are symmetrical. This symmetry, relevant to the Inclusion Model, dissapears in the Co-Occurrence Model (M_S is not equal to $1 - M_T$).

However, inclusion does exist, and could bring useful properties.

4.2.1 The Inclusion Measure

In a vector space approach, inclusion can be mesured through vector intersection and distance:

$$\begin{aligned} & H(A,B) \Rightarrow \\ & D_A(V(A), \gamma(V(A)\, V(B))) \\ & \leq D_A(V(B), \gamma(V(A), V(B))) \end{aligned} \tag{10}$$

For example, we have the following measure between *horse/mammal* and *mammal*:

$D_A(V(\text{horse}), \gamma(V(\text{horse})\, V(\text{mammal}))) = 0.41$
$D_A(V(\text{mammal}), \gamma(V(\text{horse})\, V(\text{mammal}))) = 0.25$

From this result, we deduce that *mammal* properties are included in *horse* Morover, if we know that *horse* and *mammal* are in a hyperonymic relation (either through a very good M_T value, or otherwise), then *mammal* is the hyperonym. The relationship between Inclusion and Co-occurrence Models is obvious : high M_T values for candidates provide an assumption about a good hyperonymic relationship, which in turn is cheched and thus validated (or invalidated) by the inclusion measure defined above.

4.2.2 Limits of the Inclusion Measure Scope

The model, restricted to the sole inclusion measure, operates very well for vectors that has been computed from hyperonymic definitions. But for very general terms, where definitions tends to be hyponymic (a collection of examples), the inclusion vector is reversed. More precisely, this is called the *horizon limit*. The *horizon* is constituted by leaves (terminal concepts) of the taxonomy on which the vector space is defined.

When the definition leads to a new vector, vectors of the terms present in this definition are mixed. Thus, the vector is flat compared with the main involved concept(s). We have a formal measure for *flatness* which is the *variation coefficient* V_C:

$$V_C(X) = \frac{s(X)}{\mu(X)} \quad \text{with} \quad s^2(X) = \frac{\sum(x_i - \mu(X))^2}{n} \tag{11}$$

V_C is the ratio between the standard deviation s of the vector component, and the mean μ. This a unitless value. By definition, V_C is only defined for non null vectors. If $V_C(A) = 0$ then the vector A is flat, that is, all components have the same value. At the maximum value of V_C (around 29 when $n = 873$), we have a boolean vector (only one component is activated with 1 while all others are zeros).

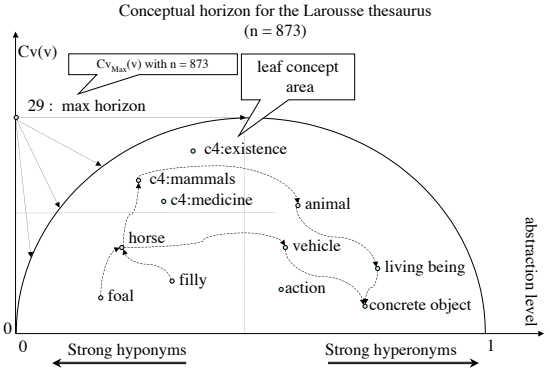


Figure 2. Graphical representation of the conceptual horizon. The horizon stands at the highest level of the variation coefficient which is the lowest level of the thesaurus hierarchy. On the left side, we have terms that are strictly specialization (by mixing) of concepts. On the right side, we have generalization of concepts, which similarly by vector mixing tend to lower the variation coefficient of vectors.

Over the horizon, we do have:

$$H(A,B) \Rightarrow D_A(V(A), \gamma(V(A)\,V(B))) \geq D_A(V(B), \gamma(V(A), V(B))) \tag{12}$$

4.2.3 The Conceptuality of a Vector : Beneath or Beyond the Concepts Hill ?

A very important issue is to be tackled: How is it possible to assess on which side of the *concepts hill* [10] a given vector stand? By itself, the variation coefficient just evaluates the general shape of the vector and its *conceptuality* relatively to the concept set. We have two ways to solve this problem:

1. the first is focusing on a lexical approach mixing lexical functions and information to vectors. The Co-occurence Model is a possible answer and, more generally, semantic graphs [11] as well. The Co-occurrence Model might be consolidated with an inclusion measure.

2. A second approach is to include, as a dimension of the vector space, every concept of the hierarchy and not only the leaves. This solution is only partial, because it cannot address the adjoining problem of polysemy when working on the lexical item level and not on the acception (conceptual) level.

We have undertaken the first approach, on a restricted scale (see discussion). The second one has been until now discarded, but before rejecting it completely, we would like to evaluate its true usefulness.

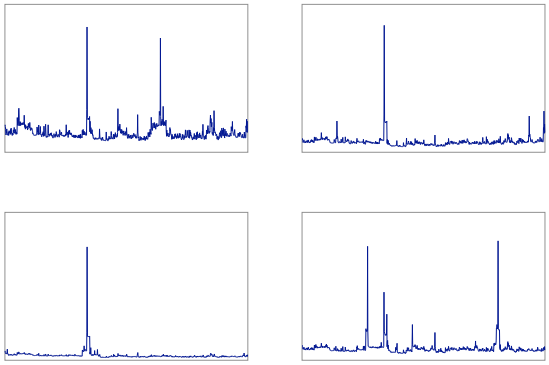


Figure 3. Graphical representations of the vectors of the terms ‹*poulain*› (in English *'foal'*), ‹*cheval*› (*'horse'*), ‹*Mammifères*› (*'mammals'*) and ‹*animal*›. The variation coefficient increases from left to right and top to bottom until the third vector (‹*Mammifères*›) (*'mammals'*) and then begins to decrease for the fourth one (‹*animal*›). Concepts are represented horizontally and their activation values vertically. If a not null vector is flat, then all concepts are equally activated. In this case the variation coefficient V_C is null.

4.3 Discussion

The experiments we have conducted (another example is given in the annex) on a collection of a few hundred nouns (and compound nouns), revealed the problem of the conceptual horizon. This horizon stands at the lowest level of the concepts hierarchy (in the hierarchy we use [10] for French language, which corresponds to the depth 4. For the Roget,

[10] the graphical representation in the preceding figure shows a reversed parabol as a representation of the concept horizon, thus the metaphor of the 'hill' looks relevant.

[11] among them, conceptual graphs or UNL based graphs are possible representations

this might go to depth 6 sometimes). Because of the nature of vector composition, the inclusion model should be inverted when terms stand beyond this horizon.

Detecting the conceptual horizon crossing is done through lexical models. More precisely, it can be achieved through the Co-occurence Model but also when identifying hyponyms. The detailed presentation of hyponyms identification is beyond the scope of this paper, but it is enough to say that more abstract terms (corresponding to large taxonomic classes) contain a large number of hyponyms. According to the Co-Occurence Model hyperonymy and hyponymy functions are not strictly symmetrical, both in their usage and behavior in corpora. In fact, if, in a semantic network in KR, hyperonymy and hyponymy are strictly symmetrical, language tends to assign different roles to hyperonyms and to hyponyms. For instance, if hyperonyms could be **good explanations through definitions**, hyponyms are **the best possible explanations through examples**. And very obviously, examples do not have the same relationship to assertion than definitions, and 'the best possible' is not even symmetrical to 'good'... However, both hyperonyms and hyponyms (of a given item) often co-appear in texts, and thus can be used together to strengthen the built network.

An application of our model, still under development, is a *paraphrase tool*, useful for stylistic goals. From a given text, the system produces a new text where terms are substituted by hyperonyms (or quasi synonyms). Initial results show that the most natural paraphrases are those which maximize the substitution value but not the taxonomic relevance. Such a tool could be used not only to globally assess the practical validity of our approach but also as a partial preprocess to Machine Translation.

5 Conclusion

In this paper we have tried to show how to account for hyperonymy within the vector-based frame for semantics, relying on a cooperation between semantic networks and conceptual vectors. After having assessed the importance of lexical functions such as synonymy and antonymy for lexical choice and conceptual vectors construction and usage, we have focused on hyperonymy, more difficult to discriminate in a numeric approach such as ours.

As our method is both data driven and hierarchy-based, we first tried to define the impact of hyperonymy by measuring distances in corpora. These distances help to define word substitution and semantic relevance (with a taxonomical aspect). The theoretical model, both within semantic networks and vector space, being the *inclusion model* we showed how inclusion has been dealt with and what results we have obtained.

Although being satisfactory, these results tend to reflect the multifaceted properties of hyperonymy: by being more complex than an *is-a* relation, hyperonymy needs to be constrained by the task to perform. If text correction or explanation are at stake, then *word substitution* is a good usage to apply hyperonymic properties. If taxonomy building is the goal, then *semantic relevance* is a better candidate. So, the same way other lexical functions such as synonymy and antonymy have been restricted by adding a notion of *relativity* when confronted to text bases, also hyperonymy appears not to be absolute, as the *is-a* relation is not either. It seems better to split it into its functions and to define it according to processing goals. Regarding applications, specific terminological database building as well as domain based ontologies for web browsing are achievable with semantic relevance. User-helping tools as linguistic assistance fit into the field of word substitution.

In a way, lexical functions, sometimes as theoretical as hyperonymy may appear to the non specialist, may have a great impact on NLP based tools for everyday assistance to computers users.

References

[1] Brachman R. J. and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[2] Chauché J. Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. *TA Information*, 31(1): 17–24.1990

[3] Deerwester S., S. Dumais, T. Landauer, G. Furnas, and R. Harshman, Indexing by latent semantic analysis. *Journal of the American Society of Information science*,416(6): 391–407,1990.

[4] Fellbaum C. (ed). WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachussetts,1998.

[5] W. Gale and K. W. Church. *Identifying Word Correspondences in Parallel Texts.* Proceedings of the DARPA SNL Workshop. Asilomar, CA. 1991

[6] Gwei G. M. and E. Foxley. A Flexible Synonym Interface with Application examples in CAL and Help environments. *The Computer Journal* 30 (6): 551–557,1987.

[7] Hearst M. A. *Automated discovery of WordNet relations*, In C. Fellbaum ed. *WordNet : An Electronic Lexical Database* MIT Press, Cambridge, MA, 131–151, 1998.

[8] Lafourcade M. and V. Prince. *Relative Synonymy and Conceptual Vectors NLPRS01*, 127-134, 2001.

[9] Lafourcade M. *Conceptual Vectors and Fuzzy Templates for Discriminating Hyperonymy - is-a - and Meronymy - part-of - relations.* In proc. of OOIS 2003 Workshop MASPEGHI, P.Valtchev, M. Huchard, H. Astudillo (eds.), Montral, Canada October 6th 2003, ISBN 2-89522-035-2, pp. 19-29.

[10] Larousse, *Thésaurus Larousse - des idées aux mots - des mots aux idées.* Larousse, 1992.

[11] Lee J. H., M. H. Kim and Y. J. Lee. Information Retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*, 49(2), 188–207,1993.

[12] Lewis C. I. *The modes of meaning.* in Linsky ed, "Semantics and the philosophy of language". Urbana. NY, 1952.

[13] Miller G. A. and C. Fellbaum. Semantic Networks in English. in Beth Levin and Steven Pinker (eds.) *Lexical and Conceptual Semantics* , 197–229. Elsevier, Amsterdam, 1991.

[14] Palmer F. R. *Semantics: A New Introduction* . Cambridge University Press, 1976.

[13] Resnik P. *Using Information Contents to Evaluate Semantic Similarity in a Taxonomy*, *IJCAI-95*, 1995.

[16] Resnik P. Semantic similarity in a taxonomy:An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11, 95–130,1999.

[17] Resnik P. Disambiguating noun groupings with respect to WordNet senses. in S. armstrong, K. Church, P. Isabelle, E.Tzoukermann, S. Manzi and D. Yarouwsky (eds.) *Natural Language Processing using Large Corpora*, Kluwer Academic, Dordrecht, 1999.

[18] Roget P. M. *Thesaurus of English Words and Phrases* Longman, London, 1852.

[19] Salton G. *Automatic Information Organisation and Retrieval*, McGraw-Hill, New York, 1968.

[20] Salton G. and MacGill M.J.. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[21] Sparck Jones K. *Synonymy and Semantic Classification*. Edinburgh Information Technology Serie, 1986.

[22] Schwab D., M. Lafourcade and V. Prince. Antonymy and Conceptual Vectors. *COLING'02*, vol 2/2, 904-910 , 2002.

[23] Yarowsky D. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *COLING'92*, 454–460, 1992.

6 Annex

Measuring M_T and M_S for the French term *peinture* :

art / $M_T = 0.133$ $M_S = 0.6913$ (a)
art de peindre / $M_T = 0.649$ $M_S = 0.0016$(a)
ouvrage / $M_T = 0.2248$ $M_S = 0.0955$ (b)
ouvrage d'un artiste / $M_T = 1.0$ $M_S = 0.00001$(b)
matière / $M_T = 0.2543$ $M_S = 0.1644$ (c)
produit / $M_T = 0.2301$ $M_S = 0.1755$ (c)
produit à base de pigments / $M_T = 1.0$ $M_S = 0.00004$ (c)
produit à base de pigments en suspension / $M_T = 1.0$ $M_S = 0.00004$ (c)
produit à base de pigments en suspension dans un liquide / $M_T = 1.0$ $M_S = 0.00004$ (c) couche / $M_T = 0.1443$ $M_S = 0.0876$ (d)
couche de couleur / $M_T = 0.4939$ $M_S = 0.0004$ (d)
description / $M_T = 0.2049$ $M_S = 0.1216$ (e)

The term *peinture* could be: (a) the *art*, (b) *painting*, (c) the *coloring matter*, (d) the *color layer*, and (e) a *description*. We can see that very precise terms are not good substitutes (see different cases for (c)). And inversely best substitutes are often more general and possibly polysemous terms.

CHAPITRE 3

Acquisition de réseaux lexicaux

Comment construire efficacement un réseau lexical ? La réponse que nous proposons ici est une alternative à l'indexation manuelle et l'extraction automatique à partir de corpus. L'approche contributive volontaire non rémunérée souffre de ne pas rencontrer de succès si la tâche n'est pas valorisée. Le projet Papillon semble avoir souffert de ce problème, sans doute parce que la granularité des tâches était trop faible, et que l'apport de chaque contributeur n'était pas mis en évidence. Une approche permettant de s'affranchir de cette difficulté consiste à proposer des jeux lexicaux où l'activité des participants a comme effet de bord la construction de la ressource visée. La valorisation de l'effort se traduit alors sous forme d'affichage de la performance du joueur via différents moyens (classement, événements, etc.) mais aussi et peut-être en premier lieu, du plaisir à jouer. Le projet JeuxDeMots part de l'hypothèse que cette approche est efficace aussi bien en temps, en qualité et en coût. Les résultats obtenus via l'expérience menée depuis septembre 2007 semblent l'avoir démontré, au moins au niveau quantitatif. La ressource, un réseau lexico-sémantique pour le français, est ainsi créée par émergence d'une activité dont les motivations sont autres (le plaisir du jeu, la confrontation aux autres) que l'existence de la ressource en soi, de la part de non spécialistes. Ce processus de construction incrémentale permet aussi d'effectuer des opérations de raffinement sur les usages de termes et de calculs de vecteurs d'idées sur les objets du réseau.

Articles joints - M. Lafourcade, A. Joubert *Word usage identification from a crowd-sourced lexical network built with online games.* in LRE (Language Resources and Evaluation), 22 p. to appear.

Les tâches requérant un processus de désambiguïsation, peuvent tirer profit de la connaissance de relations lexicales ou fonctionnelles entre termes. Ces relations, présentes généralement dans des thésaurus ou des ontologies, peuvent être mises en évidence de façon manuelle ; il est possible de citer ici, par exemple, le thésaurus Roget 1852 [Roget, 1852b], l'un des plus anciens, le thésaurus Larousse [Pechoin, 1991] pour la langue française ou l'un des plus célèbres réseaux lexicaux, WordNet ([Fellbaum, 1988], [Miller *et al.*, 1990], et [Harabagiu *et al.*, 1999]). De telles relations peuvent aussi être déterminées automatiquement à partir de corpus de textes, par exemple (Robertson et Spark Jones 1976), (Wettler et Rapp 1992) ou (Lapata et Keller 2005), dans lesquels sont effectuées des études statistiques sur les distributions de mots. De nombreux travaux ont également porté sur la détection de collocations tels ceux de (Spence et Owens 1990), [Smadja, 1993] ou plus récemment (Ferret 2002). La méthode Latent Semantic Analysis (LSA), présentée par (Dumais 1994) ou (Lan-

dauer et Dumais 1997), s'appuie également sur des ensembles de textes ; elle permet de calculer une proximité sémantique entre mots et ainsi de produire des nuages de termes appartenant à un même champ sémantique. LSA s'appuie sur le contexte dans lequel les mots sont utilisés : deux mots sont ainsi sémantiquement proches si les contextes dans lesquels ils sont rencontrés dans le corpus sont similaires. (Wandmacher et al. 2008) propose une réflexion sur la qualité des associations de termes dans LSA, si on prend comme référence celles de locuteurs. En outre, certaines applications de TAL requièrent des informations de différentes natures, comme la synonymie ou l'antonymie, mais également des relations d'hyperonymie/hyponymie, holonymie/méronymie, etc. L'établissement de telles relations, s'il est effectué manuellement par un ensemble d'experts, nécessite des ressources (en durée et en personnel) qui peuvent être prohibitives, alors que leur extraction automatique sur un corpus de textes semble beaucoup trop dépendante du domaine des textes choisis.

La méthode développée ici s'appuie sur un système contributif, dans lequel ce sont les utilisateurs qui font évoluer la base, au travers d'une interface présentée sous la forme d'un jeu en ligne. Cette méthode s'apparente à celle utilisée par [Zock & Quint, 2004] pour l'apprentissage de la langue japonaise. De plus, le prototype introduit ici permet l'acquisition d'informations lexicales évolutives, contrairement à la plupart des méthodes classiques où les informations lexicales sont généralement statiques, même si actuellement bon nombre de bases de connaissance évoluent incrémentalement au rythme des mises à jour. Par évolutif, nous entendons que l'échelle de temps est faible et que certaines données peuvent évoluer très rapidement (en fonction de l'actualité, notamment).

3.1 Acquisition lexicale par des jeux : l'exemple de JeuxDeMots

Concevoir un jeu en ligne permettant de construire par effet de bord un réseau lexical se heurte à plusieurs difficultés, dont les principales sont : (1) le jeu doit en premier lieu présenter un intérêt pour les joueurs ; (2) la mécanique du jeu doit permettre un filtrage assurant une qualité raisonnable des données récoltées ; (3) il est impératif d'identifier aussi clairement les biais qu'il présente afin d'en corriger les effets.

3.1.1 Principes généraux de JeuxDeMots

Un système, qu'il soit fondé sur des mécanismes d'extraction automatique ou une contribution volontaire de la part d'utilisateurs, est confronté en général à un niveau de bruit assez important pour les données recueillies. C'est pour cela que la validation des relations proposées indirectement par un joueur est effectuée (tout aussi indirectement) par d'autres joueurs [Joubert & Lafourcade, 2008b]. Par *indirectement*, nous entendons ici qu'il ne sont pas consciemment dans une approche contributive, mais que leur activité de jeu génère ces relations. Ce principe de *covalidation* est celui à la base des systèmes à contributions (que ce soit Wikipedia, Wiktionary pour les plus connus ou encore le projet Papillon), toutefois dans notre cas il est travesti sous la forme de jeux et les utilisateurs (les joueurs, donc) ne sont pas sollicités à contribuer mais invités à jouer. De plus, une hypothèse (longtemps implicite) sur laquelle nous nous appuyons est :

Les joueurs contribuent mieux et plus s'ils sont placés en situation de proposer des associations qu'ils pensent être valides pour autrui.

En pratique, les validations sont faites par concordance des propositions entre paires de joueurs. Ce processus de validation rappelle celui utilisé par [vonAhn & Dabbish., 2004] pour l'indexation d'images ou plus récemment par [Lieberman *et al.*, 2007] pour la collecte de connaissances dites *de bon sens*. À notre connaissance, il n'avait pas été mis en œuvre dans le domaine des réseaux lexicaux-sémantiques.

Scénario typique

Une partie se déroule entre deux joueurs, de façon asynchrone — c'est-à-dire que les deux joueurs n'ont pas besoin d'être physiquement présents dans le jeu au même moment. Le scénario utilisateur typique est le suivant.

Le joueur nommé *kaput* débute une partie. Une consigne concernant un type de compétence (associations libres, synonymes, contraires, domaines, etc.) est affichée, ainsi qu'un terme *M* sélectionné dans une base de mots. Ce joueur a alors un temps limité (de l'ordre de la minute) pour répondre en donnant des propositions correspondant, selon lui, à la consigne appliquée au mot *M*. Le nombre de propositions qu'il peut faire est borné à une dizaine par défaut. Il peut néanmoins contrôler légèrement la partie en rachetant du temps ou en augmentant le nombre maximum de propositions possibles via des crédits. Ces crédits sont gagnés lors des parties, et offrent aux joueurs des possibilités accrues de contrôle du jeu, en favorisant son appropriation.

Le joueur valide chaque proposition une à une via un bouton (ou en appuyant sur la touche entrée) (figure 3.1). En cas de remords, le joueur peut revenir en arrière et supprimer tout ou partie de ses propositions. Cela est en pratique particulièrement utile en cas de faute d'orthographe signalée par le jeu (garder des termes mal orthographiés est contre-productif du point de vue des chances de score). Si le terme est repéré comme polysémique dans la base, le joueur a la possibilité de sélectionner un des usages disponibles.

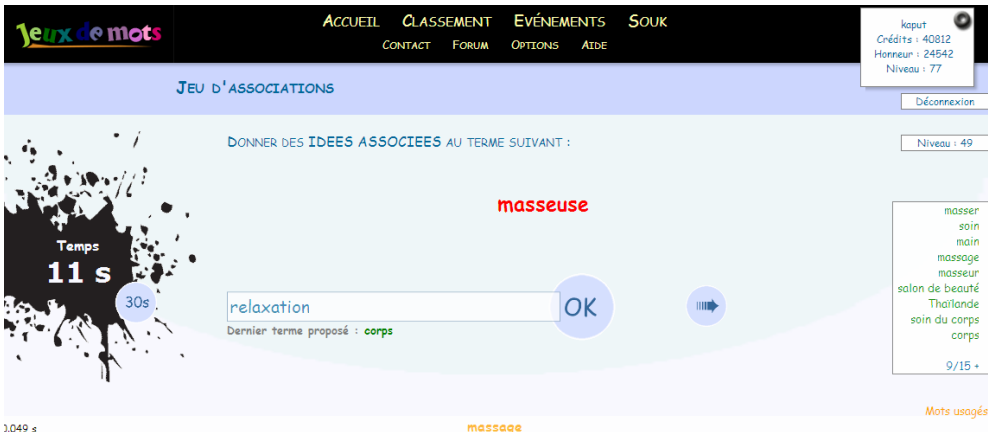


FIGURE 3.1 – Cours d'une partie de JeuxDeMots. Le joueur *kaput* doit donner des idées qu'il associe au terme *masseuse*. Il a déjà proposé 9 termes, qui sont rappelés à droite, et peut en proposer jusqu'à 15. Il lui reste 11 secondes avant que la partie ne se finisse.

Une fois le temps écoulé ou bien le nombre maximum de propositions atteint, le résultat de la partie est affiché. Le joueur gagne d'une part un certain nombre de points d'honneur qui, en quelque sorte, caractérisent son niveau de performance, et d'autre part des crédits (monnaie du jeu lui permettant plus de contrôle sur son environnement comme gagner du temps, faire plus de propositions, etc.). Le classement des joueurs est fait sur la base des points d'honneur, que ces derniers cherchent donc *a priori* à maximiser.

Ce même mot et cette même consigne sont proposés par la suite un certain nombre de fois à d'autres joueurs via un mécanisme de jetons et de priorités. L'auteur de la partie, si elle est jouée, sera notifié.

Pourquoi borner le nombre de propositions ? Il a été observé (dans une version préliminaire du jeu) que laisser ouvert le nombre de propositions avait pour effet d'encourager le joueur à maximiser

FIGURE 3.2 – Résultat de la partie de JeuxDeMots. Le joueur *kaput* a eu trois mots en commun avec le joueur *zora* et a gagné des points et des crédits.

le nombre de propositions à fournir dégradant la qualité de celles-ci. Il est par ailleurs observé que le nombre moyen de propositions pouvant raisonnablement être fournies durant une minute est de l'ordre de la dizaine. Lors d'une partie, un joueur très inspiré peut acheter à l'aide des crédits des possibilités de propositions supplémentaires. En pratique, le nombre de propositions dépasse rarement la vingtaine.

Pourquoi limiter le temps d'une partie ? Il s'agit plutôt ici d'un élément ludique, la contrainte du temps étant en général considérée comme un élément favorisant l'aspect ludique en augmentant l'excitation ressentie durant une partie. C'est un effet addictif. Toutefois, il est possible de s'affranchir partiellement de cette contrainte et d'acheter plus de temps avec les crédits. Une autre motivation, au moins aussi importante, est le désir d'obtenir une part de réponses spontanées. Le facteur temps a tendance à induire pour certains joueurs des réponses proposées dans l'urgence.

Se libérer partiellement des contraintes présentes dans le jeu représente pour le joueur un investissement, et un *pari* quant au succès de la partie qu'il est en train de jouer. Une forme d'auto-régulation se met naturellement en place chez les joueurs, en fonction des couples mot/consigne et de leur intérêt.

Modèle d'interaction

Le modèle d'interaction est schématisé par la figure 3.4. L'essence même de JeuxDeMots est donc de collecter des données via l'accord entre les joueurs. Cet agrément n'est pas négocié (donc sans rapport de force direct) dans la mesure où un joueur ne sait pas avant l'affichage du résultat contre qui il joue. Dans le cas contraire, il serait trop facile de tricher en s'entendant sur les réponses, et les données collectées n'auraient sans doute aucune valeur. À l'issue d'une partie, chaque joueur remporte strictement les même gains, faisant d'une séquence de jeu une interaction coopérative indirecte avec les autres joueurs. La compétition se fait globalement, en maximisant le nombre de parties jouées, et pour chaque partie l'investissement consenti.

Si un joueur est confronté à une partie d'un autre joueur, c'est que nécessairement des parties sont en attente dans la base du jeu. La question se pose de savoir comment se fait l'amorçage. Nous avons mis en place trois stratégies :

– proposer des *parties en création*, et donc sans affichage de résultats. Nous cherchons à minimiser ce cas de figure dans la mesure ou il n'est pas très satisfaisant pour le joueur ;

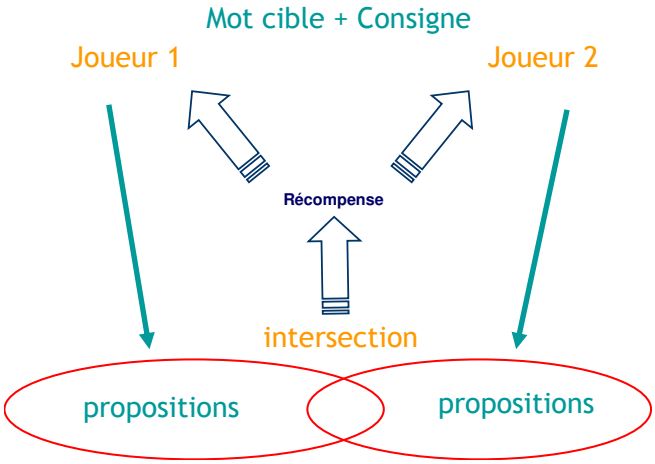


FIGURE 3.3 – Modèle d'interaction entre les joueurs et le système.

– permettre et encourager les joueurs à s'offrir sous forme de cadeaux des parties à jouer sur des mots de leur choix. Les centres d'intérêt étant partagés, la probabilité qu'une partie existe en stock sur le couple mot/consigne est plus élevée que pour un terme sélectionné aléatoirement ;

– mettre en place des robots qui sont des pseudo joueurs qui construisent des parties à l'attention des joueurs. Ce mécanisme est utilisé pour palier un manque de partie en stock, et n'est déclenché que dans le cas où aucune partie de joueur n'est disponible ;

De façon occasionnelle, un joueur joue une partie en création qui sera ultérieurement proposée à d'autres joueurs. Le joueur est averti à la fin de la partie que cette dernière était en création et qu'il sera prévenu par courrier électronique lorsqu'un autre joueur l'aura jouée. Si le nombre de parties en attente est trop faible, le joueur se verra proposer des parties à la création afin de regarnir le stock. En pratique, cela n'arrive que rarement, le niveau moyen de partie en stock étant de l'ordre de 30000 (pour un nombre moyen d'environ 2000 parties jouées par jour). Nous insistons sur le fait que le joueur ne sait qu'à l'issue d'une partie si celle-ci était en création ou en attente. Ce point est important car s'il le sait à l'avance, ce dernier pourrait systématiquement passer ce type de parties pour ne jouer que celles en attente (et donc avoir des points à la fin). Une telle pratique épuiserait vite le stock de parties en attente.

Les joueurs peuvent s'offrir des mots, c'est-à-dire inviter un autre joueur à jouer une partie sur un mot déterminé (par celui qui fait le cadeau). Les termes offerts doivent être déjà présents dans le réseau pour être l'objet de cadeau (si ce n'était pas le cas, le vandalisme de la base serait aisé). Le joueur gagne quelques points d'honneur à offrir des mots variés à autrui. De plus, si une partie d'un joueur est un succès (rapporte beaucoup de points) et était à l'origine un cadeau, un pourcentage de ces points revient au généreux donateur (c'est une forme de retour sur investissement). L'observation des joueurs montre qu'ils s'offrent des cadeaux sur des thématiques les intéressant, ce qui ainsi accélère et consolide l'indexation de termes appartenant à des champs sémantiques de leur choix. Ce comportement émergent rend inutile l'élaboration d'algorithmes sophistiqués d'échantillonnage ou de sélection de termes critiques à indexer.

Il y a dans ce qui précède un implicite fort : *les termes critiques sont ceux qui intéressent les joueurs.* Qu'en est-il alors d'un groupe de personnes souhaitant indexer à l'aide de JeuxDeMots un lexique sur un domaine spécialisé ? La réponse la plus simple est qu'ils peuvent jouer thématiquement et s'offrir mutuellement les mots relevant du lexique qui les intéresse. Ce lexique s'enrichira progres-

sivement à mesure des parties jouées et des propositions communes correspondant à des termes de spécialités (qui entreront alors dans le réseau lexical).

Enfin, il existe dans le système un certain nombre de robots (des *bots*) qui créent des parties à compléter par d'autres joueurs. Les raisons motivant l'adjonction de robots au jeu se situent à plusieurs niveaux. D'abord, il s'agit d'un moyen relativement aisé pour évaluer au long cours la qualité de vecteurs produits à l'aide d'algorithmes ou de ressources différents (voir chapitre 2). Ensuite, les joueurs apprécient assez peu d'être confrontés à des parties en ouverture, les obligeant à différer la confrontation de leurs réponses avec d'autres joueurs. Il est à noter que certaines méthodes de production de vecteurs sont tellement mauvaises que quelques joueurs en sont venus à deviner qu'il ne pouvait pas s'agir d'individus réels. Un principe important est que jamais deux bots ne sont confrontés, évitant ainsi de polluer la base lexicale avec des données calculées de qualité incertaine.

Calcul du score

Le gain d'une partie repose sur deux critères apparemment contradictoires. Pour toute réponse commune dans les propositions des deux joueurs, chacun d'eux gagne un certain nombre de points. Précisément, pour le couple terme/relation (A,R), si le terme B a été proposé par les deux joueurs, alors le nombre de points gagnés par les deux joueurs est de 1000 - poids(R). C'est-à-dire que plus la relation est récente, plus elle a de valeur, cela revient à *payer la primauté*. Cette fonction est décroissante, au fur et à mesure de son renforcement, une relation rapporte de moins en moins de points. Ainsi, cette mécanique pousse le joueur à tenter de deviner ce qu'un autre pourrait avoir en tête mais sans pour autant ne proposer que des réponses évidentes.

Ce qui rend le jeu excitant (aux dires des joueurs) est clairement le suspense lié à l'attente du nombre et de la nature des termes en commun. Le plaisir du résultat est covariant avec le nombre de mots en commun et surtout avec leur originalité. La fonction de score que nous avons choisie est la traduction de cette attente, et elle l'induit. La frustration liée à une partie ratée est également pour beaucoup, un facteur d'addiction (le joueur rejoue), à condition que cette frustration reste occasionnelle.

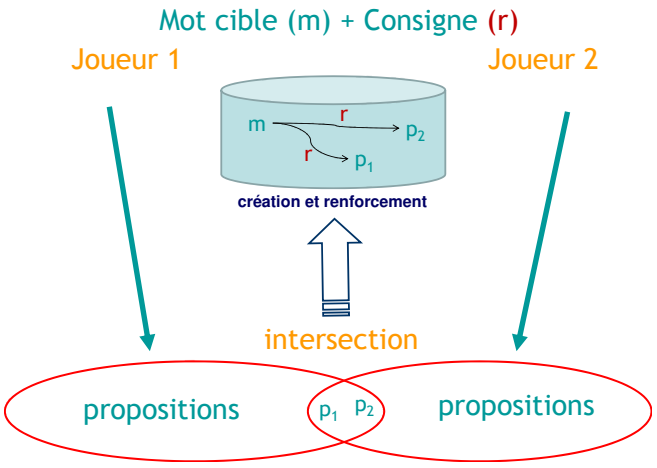


FIGURE 3.4 – Modèle d'interaction entre la partie et le réseau lexical.

Création et renforcement des relations

À l'issue d'une partie non nulle, le réseau lexical se retrouve modifié. Chaque terme commun aux deux joueurs renforce la relation lexicale entre le terme cible et la proposition, ou la crée si

elle n'existait pas. Dans la partie donnée en exemple (figure 3.2), les trois relations suivantes sont ajoutées au réseau ou ont été renforcées :

– *masseuse* $\xrightarrow{assoc}$ *masseur*
– *masseuse* $\xrightarrow{assoc}$ *corps*
– *masseuse* $\xrightarrow{assoc}$ *massage*

Plus précisément, dans le tableau 3.5, la colonne de gauche contient les relations sortantes pour le terme *masseuse* avant la partie, la colonne de droite contient les relations après la partie.

masseuse $\xrightarrow{assoc:340}$ *massage*	***masseuse*** $\xrightarrow{assoc:350}$ ***massage***
masseuse $\xrightarrow{assoc:170}$ *masseur*	***masseuse*** $\xrightarrow{assoc:180}$ ***masseur***
masseuse $\xrightarrow{assoc:150}$ *détente*	*masseuse* $\xrightarrow{assoc:150}$ *détente*
masseuse $\xrightarrow{assoc:130}$ *kiné*	*masseuse* $\xrightarrow{assoc:130}$ *kiné*
masseuse $\xrightarrow{assoc:130}$ *huile*	*masseuse* $\xrightarrow{assoc:130}$ *huile*
masseuse $\xrightarrow{assoc:120}$ *kinésithérapeute*	*masseuse* $\xrightarrow{assoc:120}$ *kinésithérapeute*
masseuse $\xrightarrow{assoc:90}$ *relaxation*	*masseuse* $\xrightarrow{assoc:90}$ *relaxation*
masseuse $\xrightarrow{assoc:80}$ *dos*	*masseuse* $\xrightarrow{assoc:80}$ *dos*
masseuse $\xrightarrow{assoc:80}$ *femme*	*masseuse* $\xrightarrow{assoc:80}$ *femme*
masseuse $\xrightarrow{assoc:80}$ *institut*	*masseuse* $\xrightarrow{assoc:80}$ *institut*
masseuse $\xrightarrow{assoc:70}$ *repos*	*masseuse* $\xrightarrow{assoc:70}$ *repos*
masseuse $\xrightarrow{assoc:60}$ *main*	*masseuse* $\xrightarrow{assoc:60}$ *main*
masseuse $\xrightarrow{assoc:50}$ *masser*	*masseuse* $\xrightarrow{assoc:60}$ *masser*
masseuse $\xrightarrow{assoc:50}$ *muscle*	*masseuse* $\xrightarrow{assoc:50}$ *muscle*
masseuse $\xrightarrow{assoc:50}$ *plaisir*	*masseuse* $\xrightarrow{assoc:50}$ *plaisir*
masseuse $\xrightarrow{assoc:50}$ *thaïlandaise*	*masseuse* $\xrightarrow{assoc:50}$ *thaïlandaise*
masseuse $\xrightarrow{assoc:50}$ *Thaïlande*	*masseuse* $\xrightarrow{assoc:50}$ *Thaïlande*
masseuse $\xrightarrow{assoc:50}$ *bien-être*	*masseuse* $\xrightarrow{assoc:50}$ *bien-être*
	masseuse $\xrightarrow{assoc:50}$ ***corps***
masseuse $\xrightarrow{assoc:50}$ *douleur*	*masseuse* $\xrightarrow{assoc:50}$ *douleur*
masseuse $\xrightarrow{assoc:50}$ *mains*	*masseuse* $\xrightarrow{assoc:50}$ *mains*
masseuse $\xrightarrow{assoc:50}$ *palper*	*masseuse* $\xrightarrow{assoc:50}$ *palper*
masseuse $\xrightarrow{assoc:50}$ *salon*	*masseuse* $\xrightarrow{assoc:50}$ *salon*
masseuse $\xrightarrow{assoc:50}$ *sauna*	*masseuse* $\xrightarrow{assoc:50}$ *sauna*
masseuse $\xrightarrow{assoc:50}$ *sexe*	*masseuse* $\xrightarrow{assoc:50}$ *sexe*

FIGURE 3.5 – État du réseau lexical avant (à gauche) et après (à droite) une partie jouée pour le terme *masseuse*.

Une nouvelle relation a été introduite, deux autres renforcées. Cet exemple est caractéristique dans la mesure où une partie moyenne a tendance à renforcer deux relations et à en créer une nouvelle.

Relations usagées

À partir d'un certain seuil pour la valeur du poids, la relation est dite usagée (ou encore, taboue). Elle est alors indiquée comme telle en même temps que la consigne et le mot M (c'est une solution donnée, exemple : aile → oiseau ou aile → avion). Jouer un mot tabou continue à rapporter des points, mais beaucoup moins : cette proposition devient moins intéressante pour les joueurs qui sont

donc invités à en trouver d'autres. Dans le paramétrage du jeu, une relation devient taboue quand elle a été proposée par 25 couples de joueurs.

L'intérêt de ce mécanisme est, avec le temps, d'augmenter la rappel. Ainsi les associations évidentes perdent de la valeur, incitant les joueurs à plus d'originalité. Lorsque le nombre de mots usagés augmente, une partie de ce couple mot/relation devient de plus en plus difficile à rentabiliser, et les joueurs ont tendance à passer (par lassitude, par crainte de ne pas rentrer dans leur investissement, etc.). Plus une partie est passée, moins elle a de chance d'être proposée automatiquement par le système. Un mécanisme de consolidation automatique, sollicité à intervalle régulier, effectue une consolidation du réseau par croisements entre parties de couples mot/relation n'ayant plus aucune chance d'être sélectionnés, puis les supprime.

FIGURE 3.6 – Partie de JeuxDeMots où de nombreux termes sont usagés/tabou relativement à la relation en jeu (idées associées). Les termes à éviter sont en orange en bas l'écran. Les termes proposés comme éléments d'inspiration sont en vert.

3.1.2 Le réseau obtenu

La structure du réseau lexical que nous cherchons à obtenir s'appuie sur les notions de nœuds et de relations entre nœuds, selon un modèle initialement présenté à la fin des années 1960 par (Collins et Quillian 1969), développé dans (Sowa 1992), utilisé dans les petits mondes par (Gaume 2006) et (Gaume et al. 2007), et plus récemment explicité par (Polguère, 2006). Chaque nœud du réseau est constitué d'une unité lexicale (terme ou expression) regroupant toutes ses lexies. Les relations entre nœuds sont typées. Certaines de ces relations correspondent à des fonctions lexicales, telles qu'explicitées par (Mel'cuk 1988), (Mel'cuk et al. 1995) et (Polguère 2003). Nous aurions souhaité que notre réseau comporte toutes les fonctions lexicales définies dans (Mel'cuk 1988), mais, compte tenu du principe de notre logiciel JeuxDeMots explicité en section 2.2, cela n'est pas raisonnablement possible. En effet, certaines de ces fonctions lexicales sont trop spécialisées ; par exemple, (Mel'cuk 1988) fait la distinction entre les fonctions Conversif, Antonyme et Contrastif. Il considère également des raffinements, avec des fonctions lexicales caractérisées de « plus larges » ou « plus étroites ». JeuxDeMots s'adressant à des utilisateurs qui sont de *simples internautes*, et non pas nécessairement des experts en linguistique, de telles fonctions auraient pu être mal interprétées par eux. De plus, certaines de ces fonctions sont trop peu lexicalisées, c'est-à-dire que trop peu de termes possèdent des occurrences de telles relations ; c'est par exemple le cas des fonctions de *Métaphore* ou de *Fonctionnement avec difficulté* et de l'ensemble des fonctions lexicales non standard ou composées.

JeuxDeMots possède une liste prédéterminée de types de relation ; les joueurs ne peuvent pas introduire de nouveaux types. Ces types de relation sont de plusieurs catégories :

- relations lexicales : synonymie, antonymie, locution, famille. Il s'agit de relations portant sur les termes, leurs dérivés, leurs substituts ou quasi-synonymes, les contraires (quels que soient les modes d'opposition définis par D. Schwab dans son travail de Master 2 et de Doctorat).
- relations ontologiques : générique (hyperonymie), spécifique (hyponymie), partie (méronymie), tout (holonymie), etc. Il s'agit de relations portant sur des connaissances liées à des objets du monde.
- relations associatives : association libre, sentiment associé, signification. Il s'agit plutôt de connaissance subjectives et globales.
- relations prédicatives : agent, patient, instrument, etc. Il s'agit de relations associées à un verbe/prédicat et aux valeurs de ses arguments (au sens très large).
- relations de typicalité : lieux typiques, moments typiques, caractéristiques typiques, etc.

Liste des relations

Ce qui suit est une liste partielle des relations (jouables) dans JeuxDeMots. Certaines autres relations peuvent être, soit présentes dans le réseau sans être jouables (comme la catégorie du discours, le niveau de langue, etc.) et donc renseignées par d'autres moyens, soit jouables mais seulement à travers des parties offertes aux joueurs (par exemple, produit/producteur, etc.). En effet, dans ce dernier cas, il semble difficile d'être capable de déterminer automatiquement quels termes pourraient entretenir cette relation.

assoc	association libre - relation associative	*chat* $\xrightarrow{assoc}$ *chien*
syn	synonyme - terme ayant un sens identique ou proche - relation lexicale	*chat* $\xrightarrow{syn}$ *matou* *voiture* $\xrightarrow{syn}$ *automobile*
anto	antonyme - terme ayant un sens contraire - relation lexicale	*chaud* $\xrightarrow{anto}$ *froid*
is-a/hyper	générique - terme associé à un générique de la cible - relation ontologique	*chat* $\xrightarrow{is-a}$ *félin* *chat* $\xrightarrow{is-a}$ *animal* *chat* $\xrightarrow{is-a}$ *animal de compagnie*
hypo	spécifique - terme associé à un spécifique de la cible - relation ontologique	*chat* $\xrightarrow{hypo}$ *chat persan*
partie	méronyme - terme désignant une partie de la cible - relation ontologique	*chat* $\xrightarrow{partie}$ *queue*
tout	holonyme - terme désignant un tout dont fait partie la cible - relation ontologique	*moustache* $\xrightarrow{tout}$ *chat* *racine* $\xrightarrow{tout}$ *arbre*

Il s'agit de relations qui sont souvent qualifiées dans la littérature de *standard*. Elles sont souvent présentes dans les ressources existantes (comme dans WordNet ou le WOLF, par exemple). Nous noterons que, souvent dans ces ressources, la relation d'hyperonymie est monovaluée, approche que nous ne suivrons pas, à la fois à cause de notre méthode d'acquisition, mais aussi car différents hyperonymes peuvent traduire différentes facettes d'un même objet (un *chat* est à la fois un *félin* et un *animal de compagnie*).

locution	locution contenant le mot cible - relation lexicale	*chat* $\xrightarrow{locution}$ *chat perché*
famille	terme de la même famille, termes dérivés et/ou apparentés étymologiquement - relation lexicale	*chat* $\xrightarrow{famille}$ *chatière*

Les relations *locution* et *famille* relèvent strictement du lexique.

obj-dom	chose→domaine - domaine auquel peut appartenir la cible - relation ontologique	*chat* $\xrightarrow{obj-dom}$ *zoologie* ; *tennis* $\xrightarrow{assoc}$ *sport*
dom-obj	domaine→chose - terme typique pouvant appartenir au domaine - relation ontologique	*zoologie* $\xrightarrow{dom-obj}$ *taxonomie*

Les relations *chose→domaine* et *domaine→chose* sont strictement thématiques et concernent la définition du champ lexical. Elles ne sont que partiellement redondantes avec l'association libre, les joueurs ayant tendance à fournir des termes qui sont pour eux des noms de domaine (la consigne induit cela). Par ce biais, il est possible de collecter des noms de domaines plus ou moins spécialisés ainsi que leurs termes associés.

obj-loc	lieu typique où peut se trouver la cible - relation ontologique	*chat* $\xrightarrow{obj-loc}$ *maison*
loc-obj	objet typique que l'on peut trouver dans ce lieu - relation ontologique	*canapé* $\xrightarrow{loc-obj}$ *chat*

obj-car	chose→caractéristique - objet typique que l'on peut trouver dans ce lieu - relation ontologique	*chat* $\xrightarrow{obj-car}$ *agile*
car-obj	caractéristique→chose - terme typique associé à la caractéristique - relation ontologique	*brûlant* $\xrightarrow{car-obj}$ *feu* *rapide* $\xrightarrow{car-obj}$ *TGV*

magn	terme typique désignant l'intensification de la cible - relation lexicale	*averse* $\xrightarrow{magn}$ *déluge*
antimagn	terme typique désignant l'amoindrissement de la cible - relation lexicale	*forêt* $\xrightarrow{antimagn}$ *bois*

pred-man	action→manière - manière typique liée à l'action - relation prédicative	*miauler* $\xrightarrow{pred-man}$ *bruyamment*

La manière dont peut être réalisée une action (le prédicat) demande, en général, un syntagme adverbial. Cette relation est particulièrement intéressante dans certain cas de désambiguïsation des verbes, par exemple *craquer nerveusement* et *craquer bruyamment*.

pred-agt	action→agent - agent typique pouvant réaliser l'action - relation prédicative	*vacciner* $\xrightarrow{pred-agt}$ *médecin*
agt-pred	inverse de la précédente - relation prédicative	*médecin* $\xrightarrow{agt-pred}$ *soigner*
pred-pat	patient typique subissant l'action - relation prédicative	*vacciner* $\xrightarrow{pred-pat}$ *enfant*
pat-pred	inverse de la précédente - relation prédicative	*enfant* $\xrightarrow{pat-pred}$ *gronder*
pred-instr	instrument typique permettant de réaliser l'action - relation prédicative	*vacciner* $\xrightarrow{pred-instr}$ *seringue*
instr-pred	inverse de la précédente l'action - relation prédicative	*pistolet* $\xrightarrow{instr-pred}$ *tirer*

Il s'agit ici des rôles sémantiques.

affect	sentiment que l'on peut avoir face à la cible - relation associative	*chat* $\xrightarrow{sentiment}$ *affection*

sens	sens/signification - terme typique (gloses) pouvant désigner un sens possible de la cible - relation associative	*bois* $\xrightarrow{sens}$ *matière* *bois* $\xrightarrow{sens}$ *lieu* *bois* $\xrightarrow{sens}$ *cornes*

Pourquoi avoir choisi ces relations-là et pas d'autres ? En effet, un bon nombre des fonctions lexicales de I. Mel'čuk ne sont pas présentes. Nous pouvons identifier deux raisons à cela. La première est que certaines relations sont sémantiquement compliquées et qu'il est difficile avec une consigne courte de faire comprendre de quoi il s'agit aux joueurs. La seconde raison est qu'il peut être problématique de sélectionner automatiquement un terme pertinent pour certaines relations. C'est particulièrement vrai pour les relations non standard de I. Mel'čuk. Par exemple, la relation *x* $\xrightarrow{produit:50}$ *y* où *x* peut produire *y* (comme par exemple *vache* $\xrightarrow{produit:50}$ *lait*) serait particulièrement utile, cependant il semble difficile d'établir une procédure automatique ne proposant que des *x* pertinents ou disposant de suffisamment de *y* pour que le jeu soit amusant.

Analyse quantitative

En 40 mois, nous avons obtenu :

1 155 967 relations lexicales pour 232 007 termes ;

141 355 termes ont au moins une relation sortante ;

75 654 termes ont au moins une relation entrante .

Concernant les relations, nous avons la répartition suivante (le détail est disponible à : `http://jeuxdemots.org/jdm-about.php`) :

- 619 471 relations pour les associations libres
- 151 881 relations pour les synonymes ;
- 39 813 relations pour les hyperonymes ;
- 11 930 relations pour les antonymes ;
- 11 073 relations pour les agents typiques ;

Environ 1,5 million de parties ont été jouées en 40 mois. À raison d'une moyenne d'une minute par partie, cela équivaut à environ 25 000 heures de jeu cumulé, pour environ 2 500 joueurs inscrits (soit 10 heures par joueur). La somme des poids des relations est de 60 millions, ce qui donne un poids moyen de 60 par relation. Le nombre de *touches* (événements où une relation a été créée ou incrémentée) est de l'ordre de 20 millions. Le nombre de relations ayant un poids supérieur à 300 (seuil pour commencer à être usagée) dépasse les 6 000.

Sans préjuger la qualité de ces relations, à quelles conditions aurions-nous pu obtenir ces chiffres de façon manuelle ? À supposer qu'un lexicographe introduise une relation par minute (ce qui est sans doute une hypothèse optimiste), il lui faudrait globalement 20 000 h (à comparer aux 25 000 h de jeu cumulées ci-dessus). L'approche de JeuxDeMots est donc théoriquement moins efficace (25% en plus) en temps global, mais ce temps est *distribué* sur 2 500 joueurs. De plus, le coût d'indexation est nul, or il faudrait bien payer le ou les lexicographes.

Analyse qualitative

L'évaluation de la qualité du réseau lexical a été entreprise de plusieurs manières. Une expérience via un outil de *mot sur le bout de la langue* est présentée dans le chapitre 5. En toute généralité, évaluer une telle ressource est un problème en soi. En effet, nous pouvons buter sur les difficultés suivantes :

- Il parait difficile d'évaluer manuellement la ressource sur toutes les entrées vu la taille de la ressource. Il est toujours possible de faire un échantillonnage pour les termes *intéressants* mais ce critère reste délicat à définir. S'il s'agit de termes du vocabulaire courant, alors nous serons confrontés à un grand nombre de relations (par exemple, *chat* a 350 relations sortantes, et 400 relations entrantes).

– N'évaluer que les relations les plus fortes serait faire fi de la longue traine (les relations de poids moyen ou faible), qui constituent la richesse d'un tel réseau. L'évaluation se ferait donc essentiellement en précision et non en rappel.
– La comparaison avec une ressource étalon se heurte à l'inexistence d'une telle ressource. De plus, la mesure ne porterait que sur la partie commune du réseau et de l'étalon, mais pas sur les relations présentes dans la ressource et absentes de l'étalon.

3.1.3 Le joueur comme sujet et le système comme scrutateur

Le système *observe* l'activité des joueurs afin d'orienter la construction du réseau lexical. Cette observation a deux conséquences sur le jeu, d'une part sur le mode de sélection des parties à jouer, et d'autre part, sur le choix des cadeaux que le système offre aux joueurs. Dans le cadre de JeuxDeMots, chaque relation prend la forme d'une consigne. Par exemple, pour la relation d'association libre, nous demanderons à l'utilisateur "Quelles idées associez-vous au terme *x*". Pour les antonymes, la consigne sera "Quels sont pour vous les contraires de *x*".

Deux problèmes se posent concernant (1) le choix des relations pour lesquelles nous souhaitons collecter des données, et (2) la manière de libeller la consigne. Le choix des relations est fortement contraint par l'aspect ludique. En effet, certaines relations bien qu'intéressantes sont trop peu productives pour être raisonnablement demandées. Par exemple, la relation "produit/est produit par", avec :

vache $\rightarrow$ *produit:x*$\rightarrow$ lait

usine $\rightarrow$ *produit:x*$\rightarrow$ bien de consommation

artiste $\rightarrow$ *produit:x*$\rightarrow$ œuvre d'art

associe en général trop peu de termes à un mot donné pour que le jeu soit réellement intéressant. De plus, il est *a priori* très difficile de sélectionner automatiquement des termes à la fois productifs et dignes d'intérêt pour cette relation.

Potentiel de relation

Comment JeuxDeMots sélectionne-t-il les couples relation/terme lors de la création d'une partie ? Le réseau lexical contient une relation dite *potentiel de relation* que nous nommerons *PR* vers un nœud spécial de potentiel de relation que nous nommerons par la suite POT_x avec *x* un type de relation. Par exemple, pour le terme *chat* nous avons :

– $chat \xrightarrow{\text{PR:434}} POT_{partie}$
– $chat \xrightarrow{\text{PR:357}} POT_{assoc}$
– $chat \xrightarrow{\text{PR:352}} POT_{is\text{-}a}$
– $chat \xrightarrow{\text{PR:246}} POT_{agt\text{-}pred}$

Si une relation est absente pour un terme, alors nous considérons selon ce modèle que son potentiel pour cette relation est nul. Un mode de sélection des couples mot/relation est donc fondé sur un tirage pseudo aléatoire en fonction des relations PR présentes dans le réseau.

À l'issue d'une partie, les relations PR du terme cible sont créées ou ajustées en fonction du score. Une *contamination* par propagation sur le réseau pour les termes associés en accord pour la partie est réalisée à l'issue de celle-ci. Une partie passée ou sans réponse (ce qui revient au même), ou une partie à score faible, fait baisser le potentiel de la relation considérée pour le terme cible. La relation est supprimée si son activation devient nulle.

Le réseau contient à l'heure actuelle environ 127 000 relations PR pour environ 65 000 termes distincts (en tout et pour tout, relations utilitaires comprises, le réseau compte environ 1,6 million de relations).

Limites de JeuxDeMots

L'approche de JeuxDeMots comporte certaines limites. Premièrement, il s'agit d'un vocabulaire actif, forcé par la consigne. Le vocabulaire est dit actif au sens où il fait partie du discours de tous les jours du joueur. Une grande partie du vocabulaire passif (connu mais non régulièrement usité) reste donc non proposée. De plus, il est forcé par la consigne, car le joueur donne ses réponses en fonction de ce qui lui est demandé et non ce qui lui vient en premier à l'esprit à part dans le cas de la consigne *association libre*. Deuxièmement, certains joueurs ont recours à des ressources externes telles que Wikipédia ou des dictionnaires de synonymes. Cela introduit un biais car il ne s'agit pas du vocabulaire actif, mais cela permet aussi d'introduire dans le réseau du vocabulaire plus soutenu. Ceci compense partiellement la limite précédente. Troisièmement, les associations sont fortement liées à l'actualité. Le réseau évolue ainsi avec le temps et est représentatif des relations sémantiques à un instant t. Par exemple, actuellement, le terme *Amérique* sera très fortement en association avec *Obama* ce qui n'était pas forcément le cas il y a quelque temps. Enfin, les connaissances sont issues d'un échantillon non représentatif de la population. En effet, toutes les tranches d'âge ne sont pas représentées, ni toutes les classes sociales.

3.1.4 Calcul de vecteurs via un réseau lexical

Nous effectuons ici un rapide retour sur le calcul de vecteurs d'idées à partir d'un réseau lexical (partie 3.6). En effet, nous avons entrepris un tel calcul (de façon incrémentale et permanente) à partir du réseau lexical de JeuxDeMots. À l'issue de chaque partie, les termes concernés par la modification des relations sont repérés comme devant avoir leur vecteur mis à jour, tâche qui est effectuée soit à la demande soit lors des moments de faible activité du système.

Le calcul d'un vecteur d'idées peut se faire à partir du réseau lexical de JeuxDeMots. Le calcul ne doit pas nécessiter d'amorçage spécifique avec des données externes, les seules données disponibles étant le réseau lui-même. Dans le cadre de nos expériences, le déclenchement du calcul d'un vecteur est effectué en fonction de l'activité du jeu. Plus précisément, lorsqu'un terme voit ses relations modifiées alors son vecteur est révisé. D'une façon générale, considérons un terme T relié à n autre terme T_i via une relation pondérée R_i. Le *vecteur agrégé* du terme T est la somme pondérée et normalisée des n vecteurs de T_i. En toute généralité, l'opération d'agrégation entre deux vecteurs n'est pas nécessairement la somme vectorielle. Nous pensons en particulier à l'opération de contextualisation faible comme possible forme d'agrégation.

Vecteurs conceptuels

Nous rappelons que pour un vecteur conceptuel, l'espace générateur est défini *a priori*, c'est-à-dire qu'à chaque composante est associé un concept clairement défini. L'ensemble fini des concepts est la base (génératrice). Chaque concept est donc un terme choisi dans le réseau. Ce terme est éventuellement désambiguïsé, par exemple, on préfèrera le terme *fruit>botanique* au terme *fruit*. Nous calculons un vecteur en ne tenant compte que des relations en rapport avec la base. Cette opération effectue implicitement une projection des associations dans le réseau sur les termes de la base (c'est-à-dire les concepts choisis).

Vecteurs pseudo-aléatoires

Si un terme est sollicité pour intervenir dans le calcul d'un vecteur celui-ci fournit son vecteur s'il existe. Dans le cas où le terme ne dispose pas de vecteur alors ce dernier est tiré aléatoirement dans l'espace. Cette approche est en fait légèrement simpliste dans la mesure où plus le nombre de dimensions est important moins nous nous écartons statistiquement d'un vecteur moyen. Il est donc

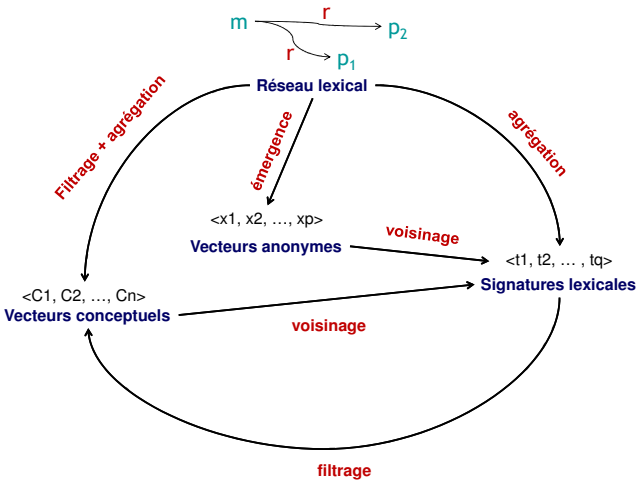


FIGURE 3.7 – Mode de construction de vecteurs d'idées à partir d'un réseau lexical

nécessaire de sélectionner un vecteur aléatoire certes, mais dont le coefficient de variation soit assez fort. Donc, la fonction de génération d'un vecteur aléatoire n'est pas équiprobable sur l'ensemble de l'espace.

Vecteurs étiquetés

Jusqu'à présent, nous n'avons présenté que des vecteurs génériques, c'est-à-dire en relation avec l'ensemble des termes reliés à un item lexical donné indépendamment du type de relation. Or, il semble intéressant de pouvoir adjoindre à un vecteur une étiquette correspondant à une relation particulière, et de disposer de plusieurs vecteurs spécifiques pour chaque terme. Ainsi, à un terme est associé un ensemble de vecteurs, chacun correspondant à une relation (pertinente pour ce terme). Nous maintenons tout de même un vecteur général pour le terme, dont la relation artificielle est notée ω. Nous parlerons donc de vecteurs ω, *hyper*, *syn*, *agent*, etc. en fonction de la relation R considérée. Le calcul d'un vecteur pour un terme T pour une relation R est la somme pondérée et normalisée des vecteurs des termes, ce que nous avons appelé le vecteur aggloméré. Le principe de calcul des vecteurs est similaire à celui présenté dans [Bouklit & Lafourcade, 2006] sur les doubles signatures de document web en récursion croisée. Le vecteur constituant le cas de base est la somme de deux vecteurs ω entrant et sortant. Le vecteur ω entrant (resp. sortant) n'est rien d'autre que la signature lexicale entrante (resp. sortante), c'est-à-dire l'ensemble pondéré des termes entrants (resp. sortants) de T. Le processus de calcul est globalement convergent.

3.2 PtiClic : un jeu de consolidation

▷ *Ce qui suit est une synthèse de [Lafourcade & Zampa, 2009a], [Zampa & Lafourcade, 2009a], [Zampa & Lafourcade, 2009b], et [Lafourcade & Zampa, 2009b]*

3.2.1 Problématique et objectifs

Les termes proposés par les joueurs lors d'une partie de JeuxDeMots sont essentiellement issus du vocabulaire actif. Cette remarque peut être partiellement tempérée par l'observation de joueurs utilisant des ressources externes comme aide (dictionnaires en ligne, Wikipédia, etc.). De plus, une majorité des parties (environ une sur deux) concerne la relation la plus simple, à savoir *idées associées*, et ceci a pour effet que plus de la moitié des relations (642409 sur 1,1 million) sont

thématiques. Il est souhaitable d'être capable de basculer une partie de ces associations vers des relations plus précises (par exemple, hyperonymes et hyponymes, contraires, etc.).

Ainsi, PtiClic peut être vu comme un jeu *dual* de JeuxDeMots : un terme cible est proposé ainsi qu'un nuage de mots, et la tâche consiste à réattribuer chacun des termes du nuage à des relations précises qu'il entretient avec le terme cible (voir figure 3.8). Fonctionnellement, il s'agit alors d'un raffinement des associations déjà existantes. Si la ressource permettant la construction du nuage est externe (il ne s'agit pas du réseau) alors, il est envisageable d'acquérir de nouvelles relations entre termes.

3.2.2 Scénario typique

La consigne donnée au joueur est la suivante :

> *Tu prends chaque mot et tu le déposes gentiment sur une des zones visibles, en fonction de son rapport au mot cible (celui qui se trouve au milieu du nuage de mots). Certains mots n'ont rien à voir et ne doivent pas être déposés. Pour d'autres il y a plusieurs possibilités, en choisir une bonne suffit. Quand tu as fini, clique sur le bouton en bas.*

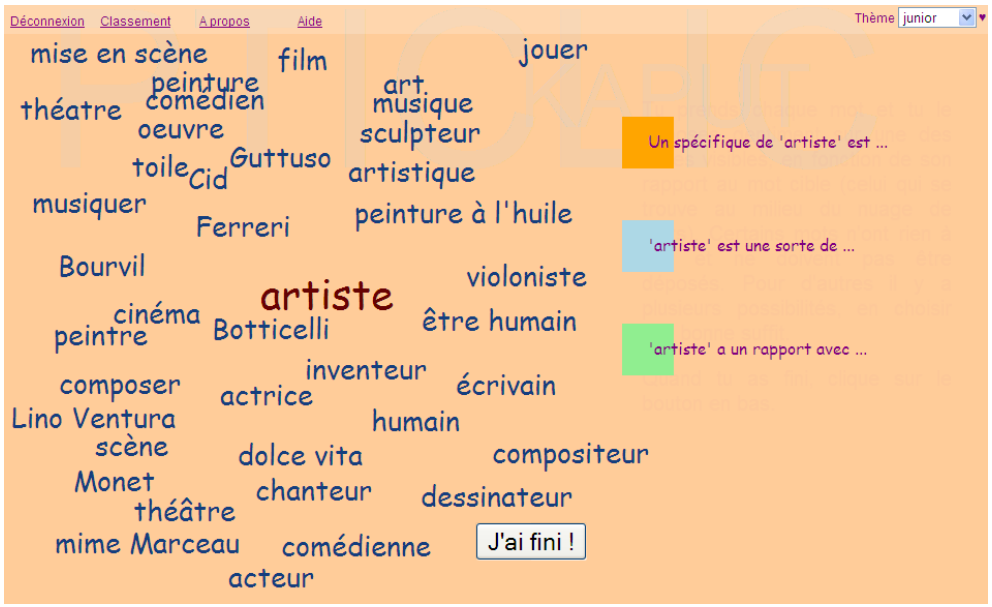


FIGURE 3.8 – Exemple de partie de PtiClic. Le mot cible est au centre et entretient ou non certaines relations avec chacun des termes du nuage de mots. Certains de ces derniers doivent être déplacés et lâchés sur le carré correspondant à la relation pertinente selon le joueur.

Comme nous l'avons déjà montré, le joueur, doit placer les mots-cibles qui conviennent dans les catégories proposées. Lorsque deux joueurs ont eu la même partie, le résultat est affiché (voir figure 3.9) et les points gagnés sont calculés par comparaison entre les accords, différences et oublis. Dans la copie d'écran ci-dessous les mots en vert correspondent à l'accord entre les joueurs, ils apportent chacun 1 point. Les mots en gris sont ceux qui ont été mis par le joueur 1 mais pas par le joueur 2. Chacun fait baisser le score de 0,5. Enfin les mots en rouge sont ceux mis par le joueur 2 mais pas par le joueur 1. Tout comme les mots manquants, chacun fait reculer le score de 0,5.

Tout comme dans JeuxDeMots, la relation n'est validée dans la base qu'après accord entre paires d'utilisateurs. PtiClic est ainsi composante de JDM agissant sur le même réseau lexical. Contrairement à ce dernier, c'est un jeu en monde clos pour les utilisateurs (le joueur sélectionne parmi

FIGURE 3.9 – PtiClic : résultat de la partie précédente.

des propositions mais ne peut en faire). Ce choix de conception permet d'obtenir des associations sur des termes relevant du vocabulaire passif sélectionnés par LSA, termes qui n'auraient pas spontanément été proposés par les joueurs. La sélection des termes se fait par énumération des termes les plus similaires (selon LSA) du terme cible. L'ajout de PtiClic dans JeuxDeMots permet de réduire le bruit (des termes mal orthographiés ou des confusions de sens) ainsi que le silence (de nouveaux termes sont introduits grâce à LSA). PtiClic permet ainsi de consolider les relations de la base et de densifier le réseau lexical.

3.2.3 Construction d'une partie

La construction du nuage de mots se fonde sur les associations entre termes fournis par LSA. De par la nature même de LSA, il s'agit d'associations relevant largement de la cooccurrence soit directe, soit indirecte par transitivité et filtrage du bruit (par réduction de la dimension de la matrice de cooccurrences). Une sélection des termes les plus proches (selon LSA) est réalisée avec un mode de préférence pour les termes qui soit n'existent pas encore dans le réseau, soit ne sont pas reliés au terme cible. Ce mode de sélection favorise donc à la fois la consolidation du réseau et la découverte de relations avec des termes issus du corpus (celui d'apprentissage de LSA), ce vocabulaire étant en général d'un registre de langue soutenu et/ou technique.

3.2.4 Injection dans le réseau JeuxDeMots

À l'issue de la partie donnée en exemple (3.8 et 3.9), les relations suivantes ont été introduites dans le réseau :

- *artiste* $\xrightarrow{hypo}$ *Léonard de Vinci*
- *artiste* $\xrightarrow{hypo}$ *Bourvil*
- *artiste* $\xrightarrow{hypo}$ *Monet*
- *artiste* $\xrightarrow{hypo}$ *Botticelli*
- *artiste* $\xrightarrow{hypo}$ *Lino Ventura*
- *artiste* $\xrightarrow{assoc}$ *Livre de peinture*
- *artiste* $\xrightarrow{assoc}$ *toile*
- *artiste* $\xrightarrow{assoc}$ *scène*

Ainsi, l'activité liée à PtiClic a un effet indirect sur l'activité liée à JeuxDeMots (et inversement). Les réponses données par certains des robots dans JeuxDeMots (cf. chapitre 2) évoluent en fonction de l'état du réseau et ont tendance à contenir une part plus importante de vocabulaire soutenu, ou de spécialité. Les joueurs en retour finissent par ajuster légèrement leur comportement, en augmentant également la qualité des réponses qu'ils fournissent. Il y a un *bouclage* (espérons vertueux) entre les joueurs et l'association JeuxDeMots-PtiClic. Nous pouvons même considérer que cette boucle est double (partiellement à l'insu des joueurs), car elle concerne d'une part les joueurs entre eux, et d'autre part, les joueurs et le système.

3.3 Identification d'usages de termes

▷ *Ce qui suit est une synthèse des travaux présentés dans [Jalabert, 2003], [Jalabert & Lafourcade, 2003], [Jalabert & Lafourcade, 2004b], [Jalabert & Lafourcade, 2004a], [Joubert & Lafourcade, 2008a] et [Lafourcade & Joubert, 2009]*

La question qui se pose ici est à nouveau de savoir si à partir du réseau lexical obtenu, il est possible de déterminer les différents sens d'un terme. Il s'agit en fait, plutôt que d'élaborer un inventaire d'acceptions (au sens classique de la lexicographie), d'essayer de déterminer de façon hiérarchique des usages pour chaque terme. Les travaux avec F. Jalabert [Jalabert & Lafourcade, 2004a] ont précédé la création de JeuxDeMots, et ont été entrepris à partir de définitions de dictionnaires et de vecteurs conceptuels.

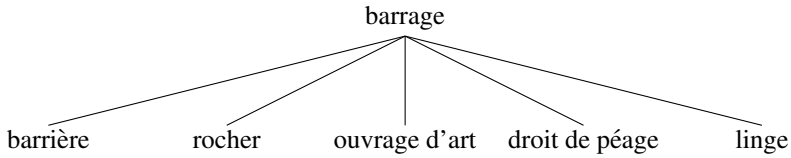
3.3.1 Cliques et usages de sens

Une première hypothèse est :

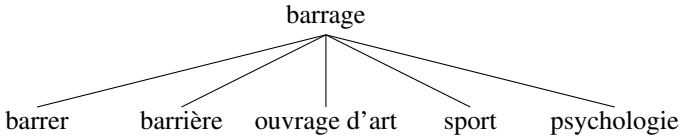
> Si un terme T est polysémique alors les termes qui lui sont reliés forment plusieurs groupes distincts, chacun de ces groupes constituant un *sens* de T.

Toutefois, le découpage en sens, de façon similaire à un dictionnaire d'usage, semble ici illusoire. En effet, ce découpage est une tâche lexicographique de projection des usages d'un terme, qui se fonde en partie sur des informations étymologiques, de fréquence, etc. Ce travail n'est pas fait, même indirectement, par les joueurs de JeuxDeMots, et il semblerait peu réaliste de le leur demander. Nous sommes donc amenés à faire la distinction entre les notions de sens d'usage et de sens. La notion de sens d'usage (appelée plus communément usage) serait beaucoup plus fidèle aux locuteurs que celle de sens qui, comme l'a montré [Veronis, 2001], est relativement pauvre lorsque nous nous référons aux dictionnaires traditionnels ou à des ressources comme WordNet.

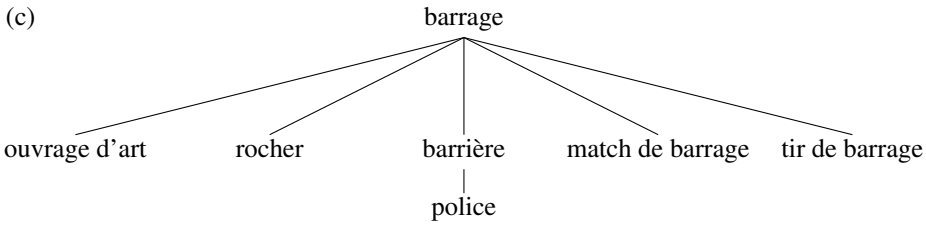
1. (a)



2. (b)



3. (c)



Ci-dessus se trouvent quelques exemples de découpage de sens pour le terme *barrage* (exemple que l'on reprend de [Veronis, 2001]. Le découpage (a) est celui de Wiktionnaire [1] et inclut deux acceptions vieillies (*droit de péage* et *linge*). Le découpage (b) est celui d'un dictionnaire en ligne [2] et il inclut un sens particulier en *psychologie*. Enfin, le découpage (c) est celui existant dans le réseau lexical de JeuxDeMots.

1. http://fr.wiktionary.org/wiki/barrage
2. http://www.le-dictionnaire.com/definition.php?mot=barrage

Par ailleurs, dans le cadre d'une analyse automatique, l'usage semble une information plus pertinente que le sens. En effet, dans le cadre de l'analyse sémantique, nous cherchons à identifier ce que l'auteur a voulu dire, même si souvent sens et usages se confondent.

L'affinement de la première hypothèse nous en donne une seconde :

> Si un terme T est polysémique alors les termes qui lui sont reliés forment plusieurs groupes distincts, chacun de ces groupes constituant un *sens d'usage* de T.

Les objets que nous allons donc identifier ne recouvrent que partiellement ceux trouvés dans les dictionnaires. Cependant, plusieurs observations semblent démontrer qu'ils sont plus facilement interprétables par des non-linguistes. La page de calcul est disponible à `http://www.lirmm.fr/jeuxdemots/rezomut.php`.

La troisième et dernière hypothèse est :

> Si un terme T est polysémique alors les termes qui lui sont reliés forment plusieurs groupes, chacun de ces groupes constituant un *sens d'usage* de T.

C'est cette dernière hypothèse que nous retenons dans la suite. Les usages ne sont clairement pas distincts et ont tendance à se recouper, voire à s'inclure tout ou partie. Un exemple représentatif est le terme *voiture*, qui désigne (au moins) (1) une automobile, (2) un wagon de voyageurs, (3) tout moyen de transport semblables par ses attributs (une caisse montée sur des roues, comme une calèche, par exemple) et peut-être (4) de façon absolue l'automobile (comme industrie, phénomène de société, etc.). Ces deux derniers usages incluent au moins partiellement le premier, qui est suffisamment fort pour ne pas pouvoir être considéré comme un raffinement.

3.3.2 Organisation d'usages de sens en arbre

Les cliques (ou quasi-cliques) trouvées pour un terme sont réorganisées en un arbre. Les cliques sont fusionnées à l'aide d'un algorithme de classification ascendant par sélection des cliques les plus proches. La fonction de similarité utilisée correspond à une adaptation de l'indice de Jaccard (qui est bien adapté à ce type d'ensembles pondérés).

Exemple pour le terme *miel*

L'ensemble des termes mutuellement reliés à *miel* est : *abeille, ours, agréable, nectar, sucre, bon, ruche, abeille (insecte)*. Une matrice de poids (non symétrique) est calculée, avec des poids correspondant à la somme des activations des relations entre chaque couple de termes. La matrice obtenue pour *miel* est :

	miel	abeille	ours	agréable	nectar	sucre	bon	ruche	abeille (insecte)
miel	1	85	70	50	110	70	50	150	50
abeille	270	1						180	107
ours	110		1					25	
agréable	35			1			110		
nectar	120	110			1	110		35	
sucre	80				25	1			
bon	25			75			1		
ruche	140	140	25		25			1	25
abeille(insecte)	75							45	1

Par convention, la diagonale de la matrice est à 1. Le problème de recherche de sous-cliques maximales strictes est NP-complet, toutefois en pratique les matrices sont rarement de grande taille (au maximum de l'ordre d'une centaine de termes). De plus, une heuristique polynomiale exploitant les poids de la matrice est utilisée, mais dans certains cas, celle-ci regroupe des cliques en quasi-cliques.

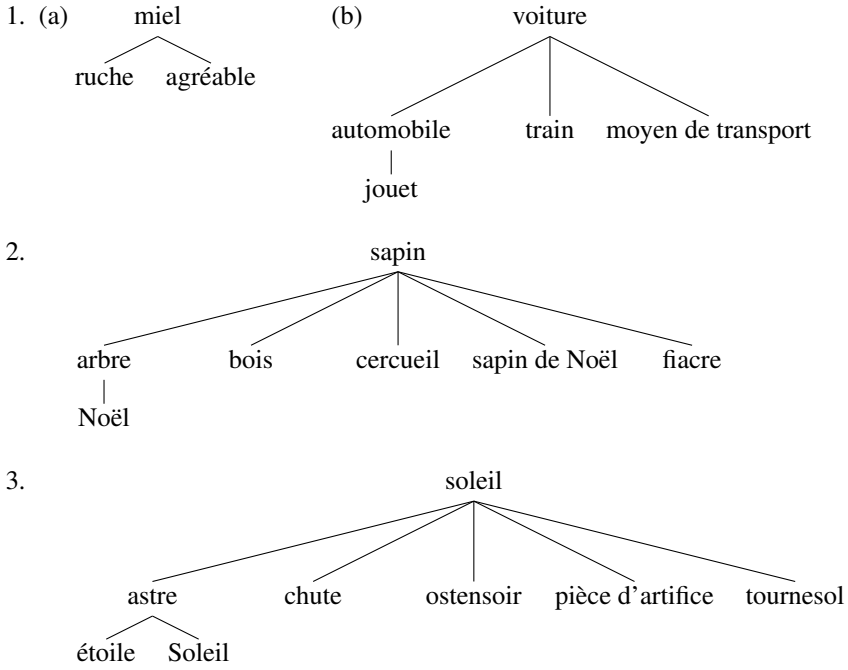
Les (quasi-)cliques trouvées sont les suivantes :
– 1 : 'miel' 'abeille' 'ours' 'nectar' 'ruche' 'abeille (insecte)'
– 2 : 'miel' 'nectar' 'sucre' 'ruche'
– 3 : 'miel' 'agréable' 'bon'

Les cliques 1 et 2 sont fusionnées (similarité de 0,57) en une nouvelle pseudo-clique :
– 4 : 'miel' 'abeille' 'ours' 'nectar' 'sucre' 'ruche' 'abeille (insecte)'

Deux usages pour le terme *miel* correspondant aux cliques 4 et 3 sont obtenus. Il est nécessaire de nommer ces cliques, c'est-à-dire de trouver une glose permettant d'identifier l'usage. Les termes les plus représentatifs pour chacune des cliques sont respectivement : *ruche* pour la clique 4 et *agréable* pour la clique 3. Nous obtenons donc deux usages :
– 'miel (ruche)'
– 'miel (agréable)'

De façon simplifiée, la représentativité d'un terme dans une clique correspond à sa côte-part des poids des relations dans lesquelles il est impliqué. Par exemple, dans la clique 3, *agréable* a un poids de $(50 + 70 + 35 + 110 + 1 =)$ 266 et *bon* a un poids de $(50 + 110 + 25 + 75 + 1 =)$ 261.

Quelques arbres d'usages



Les arbres ainsi produits sont de profondeur faible qui ne dépasse que rarement 3. Les usages se raffinent en fréquence (de gauche à droite) et en profondeur. Les usages les plus notables remontent vers la racine de l'arbre. Les gloses ne sont en aucun cas définitoires (ce ne sont pas systématiquement des hyperonymes) et n'ont comme fonction que d'évoquer un usage particulier. Enfin, les usages ne sont pas distincts, certains pouvant en englober partiellement ou totalement d'autres (par exemple, c'est le cas de *voiture*).

3.3.3 Validation par réinjection dans le jeu

Les usages ainsi identifiés et nommés sont *validés manuellement* et injectés dans le réseau JeuxDeMots. Ils sont donc disponibles pour être les termes cibles de parties proposées aux joueurs. De plus, lors d'une partie quelconque, ces usages peuvent être proposés comme réponses (en raffinant le terme général).

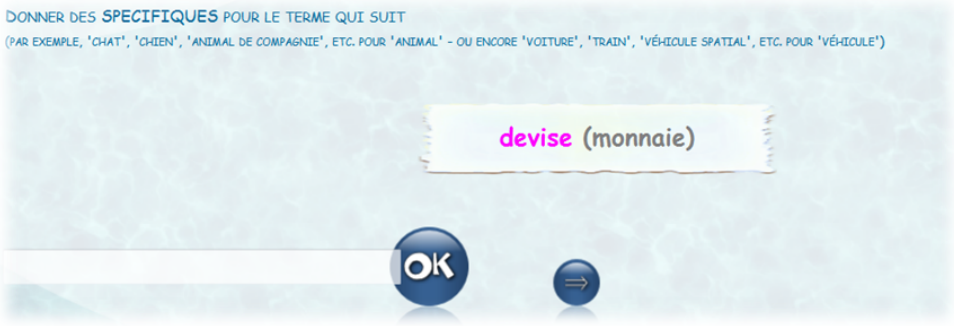


FIGURE 3.10 – JeuxDeMots : partie proposée sur un terme raffiné.

FIGURE 3.11 – JeuxDeMots : usages proposés au joueur pour le terme *livre*. Il est aussi possible de solliciter l'aide d'autres joueurs, si des raffinements sont manquants.

Un joueur peut pour un terme ne disposant pas d'usage identifié, demander aux autres joueurs de fournir des gloses possibles (sous forme de cadeaux indirects). Ainsi, pour ce terme, le réseau se voit enrichi de ces gloses, ce qui consolide les données fournies à l'algorithme de recherche de cliques, mais surtout offre un choix de termes explicites pour détermination du nommage des usages.

Enfin, un joueur peut décider de raffiner lui même un terme en lui adjoignant une glose (par exemple, saisissant au clavier *chat félin*). Ce terme raffiné est ou non validé par la suite. En cas de non validation, il est fusionné avec au choix, le terme général, un des autres raffinements, ou encore comme un usage d'un raffinement déjà existant.

Conclusion du chapitre 3

JeuxDeMots est un jeu en ligne sur le Web dont l'objectif est la construction d'un réseau lexico-sémantique. L'émergence de relations typées et pondérées entre termes s'effectue grâce au concours d'un grand nombre de personnes dont l'activité ludique induit la construction de ce réseau. Ces utilisateurs ne sont ni des linguistes ni des lexicographes, mais nous postulons que leur nombre permet d'obtenir relativement rapidement une ressource de bonne qualité, avec une couverture satisfaisante sur l'ensemble du lexique et des connaissances générales. Notre but n'est pas la constitution d'une base d'experts (lexicographes ou de spécialités), mais d'une base de connaissances *moyennes* relevant du consensus populaire, et représentant une culture générale commune. Bien que nous ne l'ayons pas mis en évidence à l'observation des données, il est vraisemblable que les associations du réseau soient quelque peu biaisées, JeuxDeMots étant un jeu en ligne. En effet, les utilisateurs ont vraisemblablement des profils particuliers, sans doute partiellement représentatifs de la population. Ceci étant dit, une majorité importante (60%) des joueurs sont des joueuses dans la trentaine, ce qui peut constituer un autre type de biais. Certaines informations relevant du profil des joueurs (âge, sexe, locuteur natif ou non, niveau d'éducation) sont occasionnellement demandées, mais ne font pas partie du système de jeu.

Le biais induit par la modalité du jeu (question ouverte) est que les termes proposés par les joueurs relèvent surtout du vocabulaire actif. Afin de compenser ce comportement, nous avons proposé un jeu *dual*, PtiClic. Cette fois ci, les termes sont proposés au sein d'un nuage de mots, et la tâche consiste à sélectionner de façon pertinente la relation que le terme pourrait entretenir avec le mot cible.

Bien qu'encore partiel, ce réseau nous permet d'identifier les différents sens d'usage pour les termes qu'il contient. Cette tâche se fait via un algorithme de recherche de sous-cliques maximales et indirectement sous le contrôle des joueurs avec un processus de validation manuelle. Une question demeure actuellement ouverte : est-il possible que, malgré l'évolution de notre réseau, des cliques actuellement séparées ne fusionnent jamais, alors qu'il faudrait manifestement les considérer comme d'un même usage ? Est-il possible qu'un biais soit induit par notre méthodologie ou par la la représentation en réseau lexical ? Pour le moment, nous n'avons pas été confronté à des résultats de calcul de cliques aboutissant à des usages manifestement en contradiction avec ceux des locuteurs.

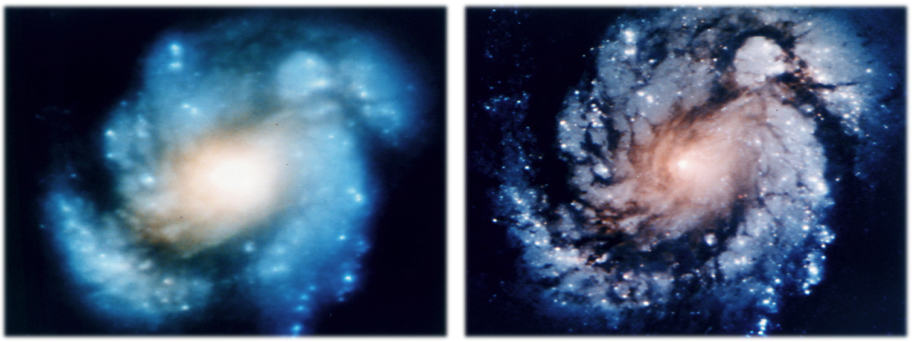


FIGURE 3.12 – Raffinement : découverte et *mise au point>optique* incrémentale des objets et de leurs relations.

La philosophie derrière la construction du réseau, une fois entendu le point clé de la mise à contribution d'utilisateurs via des jeux, est celle du *raffinement*. L'image du réseau se précise progressivement, de nouveaux objets apparaissent (nouveaux termes et nouveaux usages), des relations vagues (associations libres) se traduisent en relations plus précises. Cette évolution est covariante avec une conception possible de l'analyse sémantique, où les structures sémantiques liées au texte (relations entre les termes, les syntagmes, ou des concepts) apparaissent et se précisent à mesure du processus.

L'approche de JeuxDeMots a suscité un certain intérêt dans la communauté et a débouché sur des versions dans d'autres langues : le thaï (avec l'Université Kasetsart à Bangkok), l'anglais (LIRMM), espagnol (LIRMM, par J. Parra), l'arabe (LIG, par M. Daoud [Daoud, 2010]), le vietnamien (centre MICA à Hanoi), le japonais (par M. Mangeot-Nagata), et sans doute à venir une version malaise (avec la Maison du Monde Malais de La Rochelle et la MMU à Kuala Lumpur).

Article adjoint au chapitre 3

M. Lafourcade, A. Joubert *Word usage identification from a crowd-sourced lexical network built with online serious games.* in LRE to appear.

Annexe : sur la distribution des poids des termes

Nous nous sommes posé la question de la distribution des poids des termes dans le réseau lexical de JeuxDeMots. En particulier, si pour chaque terme nous effectuons la somme des poids de ses relations entrantes ou sortantes, que pouvons-nous observer globalement ? Une hypothèse est que le poids d'un mot (au sens ci-dessus) a un rapport covariant avec sa fréquence d'utilisation dans la langue. La fréquence d'usage d'un terme suit approximativement une loi de Zipf (voir par exemple, [**?**]) qui est une loi de puissance. Les figures 3.13 présentent, à trois échelles différentes, la distribution des termes dans le réseau JeuxDeMots en fonction des poids des relations sortantes.

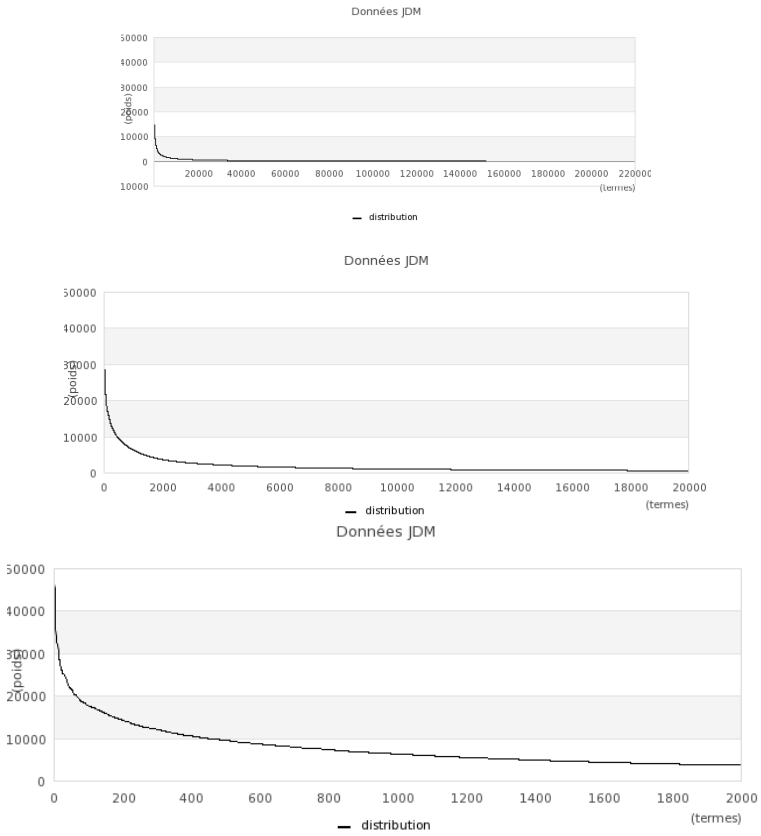


FIGURE 3.13 – Représentation en échelle linéaire — à trois niveaux de zoom différents pour l'abscisse — de la distribution des termes de JeuxDeMots en fonction des poids entrants. L'idée de *longue traine* est clairement illustrée ici.

Nous observons clairement que cette courbe a une allure très zipfienne. Cependant, si nous traçons les mêmes données non pas sur une échelle linéaire mais en échelle log-log, nous obtenons une distribution relevant plutôt d'une loi de Mandelbrot de la forme :$f(n) = K/(a + bn)^c$.

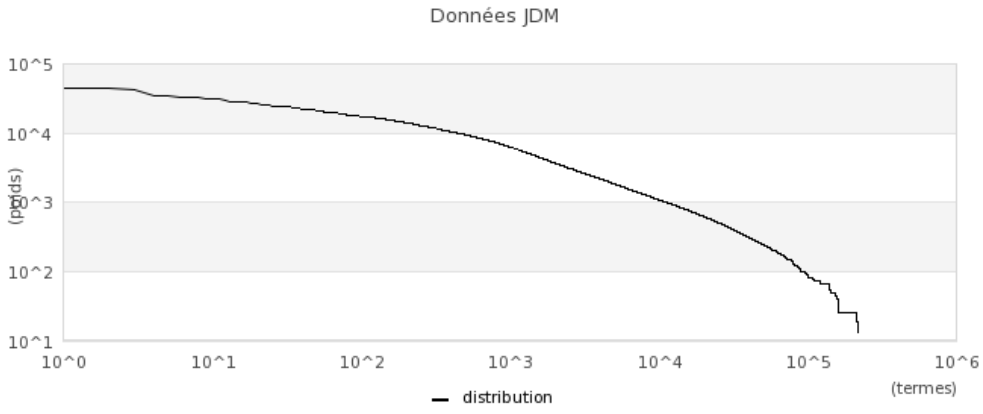


FIGURE 3.14 – Représentation en échelle log-log de la distribution des termes de JeuxDeMots en fonction des poids sortants.

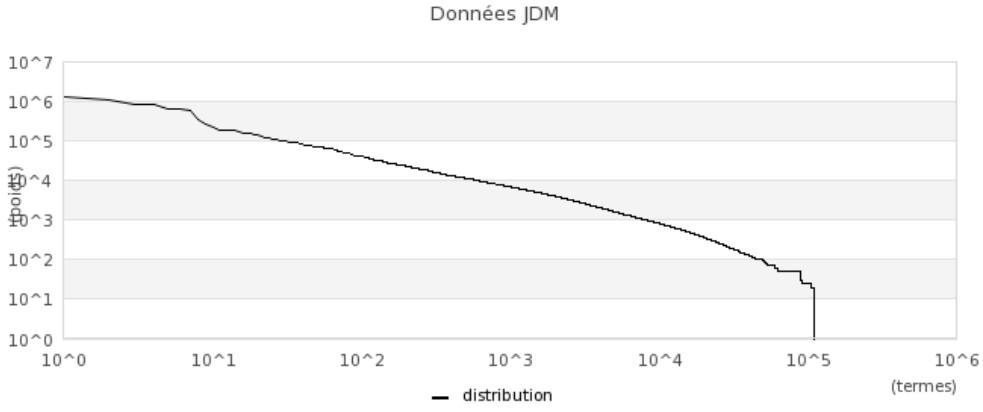


FIGURE 3.15 – Représentation en échelle log-log de la distribution des termes de JeuxDeMots en fonction des poids entrants.

Certes, et alors ? Sans pour autant clairement démontrer qu'il y a un rapport étroit entre la fréquence d'usage des termes en langue et les poids des relations de termes dans JeuxDeMots, ces distributions semblent être des indices intéressants.

Word usage identification from a crowd-sourced lexical network built with online games

Mathieu Lafourcade, Alain Joubert

LIRMM – Univ. Montpellier 2 - CNRS
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
161, rue Ada – 34392 Montpellier Cédex 5 – France
{lafourcade, joubert}@lirmm.fr

Abstract
With the help of a large number of persons playing web-based games, a large-sized evolutionary lexical network has been constructed for French. With this resource, we try to tackle the question of determining the word usages of a term. From the lexical network, a given usage correspond to a subgraph that is fully (or almost fully) connected, i.e. a clique. After introducing the notion of similarity between word usages, we are able to build the word usage tree for a given term: the root groups together all possible usages of this term and going deeper in the tree corresponds to some refinements of these word usages. As a given usage should be named in a tractable way, labeling of the various nodes of the word usage tree of a term have to be computed: the root is labeled by the term itself and each node of the tree is recursively labeled by a term stemming from the clique or quasi-clique this node represents. As some nodes of the tree might not be labeled without ambiguity, the tree has to be pruned. This paper ends with a user evaluation on the word usages identified in our lexical network.

Keywords
Game With A Purpose (GWAP), crowdsourcing, collaboratively constructed lexical network, labeled word usage tree, usage weighing, word usage identification

1. Introduction

Many applications in Natural Language Processing (NLP), in particular those requiring a process of lexical disambiguation or understanding, require the knowledge of lexical or functional relations between terms. These relations that we generally find in thesaurus or ontologies can be revealed in a manual way; it is possible to quote here, for example, the thesaurus (Roget 1852), one of the oldest, the thesaurus Larousse (Pechoin 1991) for the French language or one of the most famous lexical networks, WordNet (Miller et al. 1990). Such relations can also be determined automatically from corpuses of texts, for example (Robertson and Spark Jones 1976), (Wettler and Rapp 1992) or (Lapata and Keller 2005), in which statistical studies on the distributions of words are done. Numerous works also concerned the detection of collocations as those of (Spence and Owens 1990), (Smadja 1993) or more recently (Ferret 2002). The Latent Semantic Analysis method (LSA), presented by (Dumais 1994) or (Landauer and Dumais 1997), is also based on sets of texts; it leads to the computation of the semantic proximity between words and as such identify clusters of terms belonging to the same semantic field. We shall find a recent reflection in (Wandmacher et al. 2008) which compares human free associations and similarity measures produced by LSA. Besides, several applications in NLP require information of various natures, such as synonymy or antonymy, but also ontological relations of hyperonymy / hyponymy, holonymy / meronymy,… Building such relations, if it is manually done by a set of experts, requires resources (time and staff) which can be prohibitive, while their automatic extraction from a corpus of texts seems far too dependent on the domain of the texts chosen and cluttered by errors (a too strict error reduction process can induce a dramatic gap of knowledge extraction). In fact, we aim at achieving a word meaning inventory, more precisely a word usage inventory with their association relations (we will briefly remind the difference between meanings and usages in this paper). In corpora, what is obvious is not generally written; for example, what everyone knows is left implicit. So, one of our basic hypothesis is that *all the information needed is not expressed in corpora*, but nevertheless we attempt to find it. Moreover, as the relative weight of information is useful but is generally not given, we try to produce or harvest such data.

The method developed here relies on a contributory system, where the users develop the base, through an interface presented as an on-line game. It is typically a Game With A Purpose, also called a Human Computation Game, as first defined by (von Ahn, 2006): it is a “system that combines humans and computers to solve large-scale problems that neither can solve alone”. Our method is similar to the one used by (Zock and Quint 2004) for learning the Japanese language. Furthermore, the prototype introduced here undertakes an acquisition of evolutionary lexical information, contrary to most classic methods which generally allow for the acquisition of static lexical information, even if at present a lot of knowledge bases evolve incrementally. Furthermore, being constructed either by experts (generally lexicographers) or from corpora, they barely reflect what people would think of. We implemented a method, thanks to a collaborative process, that allows to build a large-sized lexical network, representing a common general knowledge. We admit that the obtained lexical resource can be biased by the fact that the contributors of this network are Internet users whose general profile (rather thirty-year, 60% woman, with a level of studies of at least BS) does not reflect necessarily the global distribution of the population.

The method and evaluation presented in this paper are based on a resource under construction but already freely available under a Creative Common License: the JeuxDeMots lexical network[1]. We first introduce the principles of two games which aim at building a base of relations between terms. The first of these two games (JeuxDeMots[2]) allows the construction of a lexical network, while the second game (PtiClic[2]) allows the user to strengthen associations acquired thanks to JeuxDeMots. With the network thus obtained, we tackle the problem of word usage determination, by analyzing the relations between each term and its immediate neighbors. Having a word sense inventory and relevant relations between senses and terms is mandatory prior to any text analysis application but those resources are quite rare especially for French. The similarity between the various usages of the same term can be computed, thus enabling us to build the classification tree of the usages of a term, the nodes of which being labeled. Such a word usage tree structure as of primary interest for WSD should be evaluated against users before considering using it in applications. Furthermore, one of the objectives is to be able to connect the relations, not on the very terms (with ambiguities for polysemous terms), but on their usages (thus clearing up lexical ambiguities). This stage begins to work, but the term refinement needs currently an expert’s help for validation. This leads to a double iterative process: the users on the one hand, and the expert on the other hand, incrementally increasing the lexical network.

2. Construction of the lexical network

2.1. Structure of the lexical network

The structure of the lexical network we are building is composed of nodes and links between nodes, as it was initially introduced in the late 1960s by (Collins and Quillian 1969), developed in (Sowa 1992), used in the small worlds by (Gaume 2006) and (Gaume et al. 2007), and more recently clarified by (Polguère 2006). A node in the network refers to a term (or a multiple word expression), usually in its canonical form (lemma). The links between nodes are typed and interpreted as a possible relation holding between the two terms. Some of these relations correspond to lexical functions, some of which have been made explicit by (Mel'čuk 1988), (Mel'čuk et al. 1995) and (Polguère 2003). It would have been desirable that the network should contain all the lexical functions defined in (Mel'čuk 1988), but, considering the principle of our software JeuxDeMots, detailed in section 2.2, it is not reasonably feasible. Indeed, some of these lexical functions are too much specialized; for example, Mel'čuk makes the distinction between the *Conversive*, *Antonym* and *Contrastive* functions. He also considers refinements, with lexical functions characterized as *"wider"* or *"more narrow"*. JeuxDeMots being intended for users who are "simple Internet users", and not necessarily experts in linguistics, such functions could have been badly interpreted by them. Furthermore, some of these functions are too poorly lexicalized, that is, very few terms possess occurrences of such relations; it is for example the case of the functions of *Metaphor* or *Functioning with difficulty*.

More formally, a lexical network is a graph structure composed of nodes (vertices) and links.

- A node is a 3-tuple : <name, type, weight>
- A link is a 4-tuple <start-node, type, end-node, weight>

[1] This resource is available at http://www.lirmm.fr/jeuxdemots/rezo.php for the lexical network and at http://www.lirmm.fr/jeuxdemots/diko.php for the obtained dictionary.

[2] JeuxDeMots and PtiClic are accessible at http://jeuxdemots.org and http://pticlic.org. An English version has recently been added, as well as a Thai, Japanese, Arabic and Spanish versions (http://www.lirmm.fr/jeuxdemots/world-of-jeuxdemots.php)

The name is simply the *string* holding the term. The type is an encoding referring to the information holding by the node. For instance a node can be a term or a Part of Speech (POS) like :Noun, :Verb. The link type refers to the relation considered. A node weight is interpreted as a value referring to the frequency of usage of the term. The weight of a relation, similarly, refers to the strength of the relation. Figure 1 shows a partial example of the kind of lexical network we are dealing with.

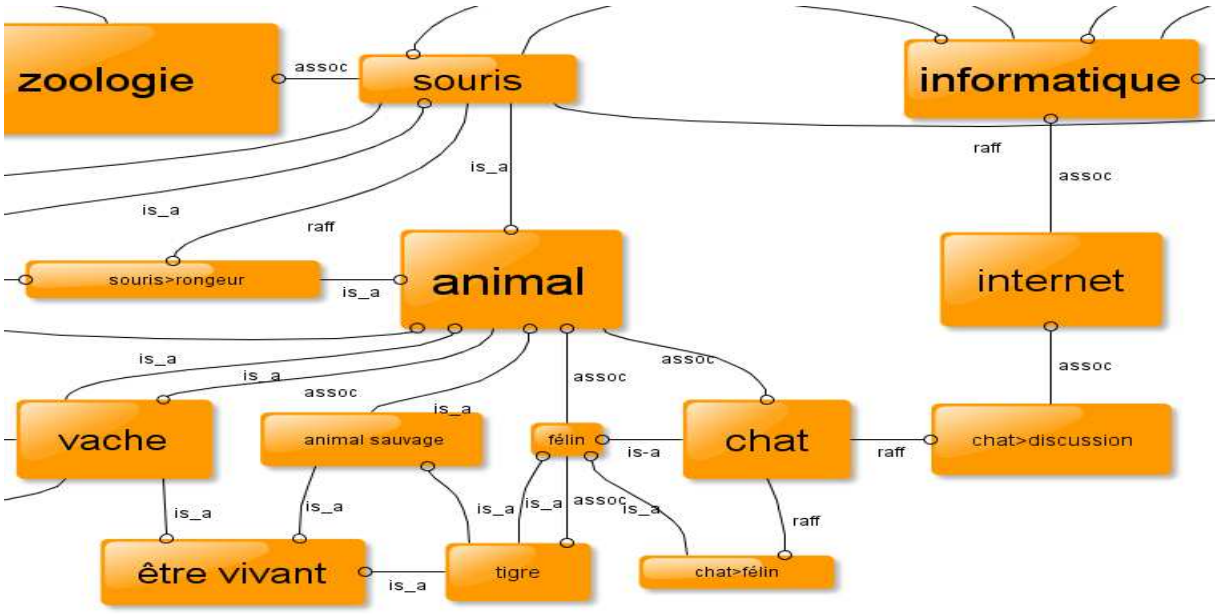


Figure 1: An example of a part of the lexical network. For the sake of clarity, the relation weights are not represented here. Only nodes corresponding to terms are shown.

For example, in Figure1, we have : <vache, is_a, animal, 200> (<cow, is_a, animal, 200>). We should insist that all information is represented in a generic way with the node and links structure. POS information (amongst others, like some semantic features, transitivity or intransitivity of verbs, etc.) is also represented by nodes, for example:

<vache, pos, :Noun, 100> (cow)
<brouter, pos, :Verb, 100> (graze)

JeuxDeMots possesses a predetermined list of relation types, and for now the players cannot add new relation types. These types of relation fall into several categories:

- Lexical relations: synonymy, antonymy, expression, lexical family

These types of relation are about vocabulary.

- Ontological relations: generic (hyperonymy), specific (hyponymy), part of (meronymy), whole of (holonymy) …

It is about relations concerning knowledge in objects of the world.

- Associative relations: free association, associated feeling, meaning

It is rather about subjective and global knowledge; some of them can be considered as phrasal associations.

- Predicative relations: typical agent, typical patient …

They are about types of relation associated with a verb and the values of its arguments (in a very wide sense).

The types of relation implemented in JeuxDeMots are thus of several natures, partially according to a distinction made by (Schwab and Lafourcade 2007): some are part of knowledge of the world (hyperonymy / hyponymy, for example), others concern linguistic knowledge (synonymy, antonymy, expression or lexical family, for example). Most players do not make this distinction which remains often vague for them.

Throughout this article, the word “relation” has to be understood as an occurrence of relation, and not as a type of relation. Such relations are oriented and non-symetrical. Let us note that between two same terms, several relations of different types can exist. Figures 5 and 10 (below) show some examples of relations acquired in JeuxDeMots.

2.2. Principle of the software

To ensure a system leading to quality and consistency of the base, it was decided that the validation of the relations anonymously given by a player should be made by other players, also anonymously. Practically, a relation is considered valid if it is given by at least one pair of players. This process of validation is similar to the one used by (von Ahn and Dabbish 2004) for the indexation of images or more recently by (Lieberman et al. 2007) to collect common sense knowledge or (Siorpaes and Hepp, 2008) for knowledge extraction. As far as we know, this has never been done in the field of lexical networks. In Natural Language Processing, other Web-based systems exist, such as *Open Mind Word Expert* (Mihalcea and Chklovski 2003) that aims to create large sense tagged corpora with the help of Web users, or *SemKey* (Marchetti et al. 2007) that exploits WordNet and Wikipedia in order to disambiguate lexical forms to refer to a concept, thus identifying a semantic keyword.

Figure 2: The home page of the game JeuxDeMots.
From here, the player has status information and can launch a game through the *jouer (play)* button.

A game takes place between two players, in an asynchronous way, based on the concordance of their propositions. When a first player (A) begins a game, an instruction concerning a type of competence (synonyms, opposite, domains …) is displayed, as well as a term T randomly picked in a base of terms. This player A has then a limited time to answer by giving propositions which, to his mind, correspond to the instruction applied to the term T. The screen displayed to this player is similar to the one presented in figure 3. The number of propositions which he can make is limited, inducing players not just type anything as fast as possible, but to have to choose amongst all answers they can think of. The same term, along with the same instruction, is later given to another player B; the process is then identical. To increase the playful aspect, the two players are given a number of points for every answer which is common to both of them. The calculation of this number of points (explained in (Lafourcade and Joubert, 2008)) is crafted to induce both precision and recall in the feeding of the database. Figure 4 presents an example of the screen displayed at the end of a game; propositions made by the two players are shown, as well as the intersection between these terms and the number of points they win.

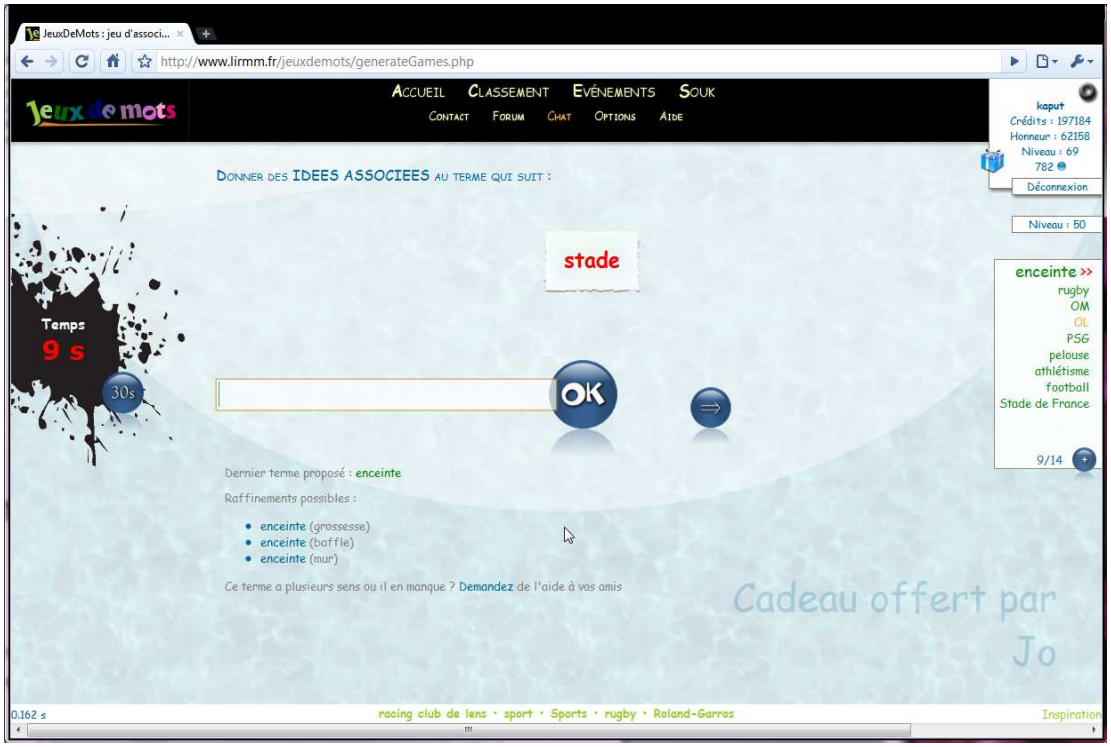


Figure 3: Example of a current game: the term played is *stade (stadium)*, The user suggested 9 terms which, in his mind, seem to be related with it through the relation *Idées associées (associated ideas)*.

Figure 4: Display of the results of the previous game: some relations like *stade (stadium)* → *athlétisme (Athletics)* will be created or strengthened. A new term *(OL)* is introduced in the database.

For the target term T, we record the common answers from both players. We do not record answers given only by one of the two players but by pairs of players. It is a compromise between validating all the answers with necessarily a high noise level and validating by intersection between several players with a reduction of knowledge recovery. The process we perform allows the construction of a lexical network connecting the terms by typed and balanced relations, validated by pairs of players. These relations are labeled by the instruction given to players and they are weighted according to the number of pairs of players who proposed them. Initially, nodes are constituted by an initial set of terms, but if both players in the same game suggest a term initially unknown, it is then added to the lexical base. Figure 5 presents some of the relations acquired for the French term *stade* (*stadium*, *stage*).

188 relations ==>

stade ---r_lieu_action:470--> jouer
stade ---r_lieu-1:450--> joueur
stade ---r_lieu_action:350--> courir
stade ---r_lieu-1:320--> sportif
stade ---r_associated:220--> sport
stade ---r_lieu:190--> ville
stade ---r_meaning:160--> terrain
stade ---r_associated:130--> football
stade ---r_carac:110--> grand
stade ---r_associated:80--> sportif
stade ---r_meaning:70--> niveau
stade ---r_hypo:70--> stade de foot
stade ---r_holo:70--> ville
stade ---r_carac:70--> olympique
stade ---r_isa:60--> lieu
stade ---r_has_part:60--> pelouse
stade ---r_familly:60--> stades
stade ---r_sentiment:60--> excitation
stade ---r_sentiment:60--> joie
stade ---r_associated:50--> Parc des Princes
stade ---r_has_part:50--> sportif
stade ---r_holo:50--> complexe sportif
stade ---r_lieu:50--> complexe sportif
stade ---r_magn:50--> enceinte
stade ---r_familly:50--> stadium
stade ---r_raff_sem:25--> stade (lieu)
stade ---r_raff_sem:25--> stade (mesure de longueur)
stade ---r_domain:25--> antiquité
stade ---r_domain:25--> sport
stade ---r_syn:25--> degré
stade ---r_syn:25--> phase
...

148 relations <==

foot ---r_lieu:450--> stade
sportif ---r_lieu:440--> stade
football ---r_lieu:160--> stade
combattre ---r_action_lieu:120--> stade
joueur ---r_associated:110--> stade
sportif ---r_holo:110--> stade
football ---r_associated:100--> stade
jouer ---r_associated:90--> stade
jouer ---r_action_lieu:90--> stade
pelouse ---r_associated:90--> stade
but ---r_holo:80--> stade
but ---r_lieu:80--> stade
Sports ---r_associated:70--> stade
joueur ---r_lieu:70--> stade
pelouse ---r_lieu:70--> stade
Parc des Princes ---r_associated:60--> stade
accélérer ---r_action_lieu:60--> stade
cirque ---r_syn:60--> stade
terrain ---r_associated:60--> stade
terrain ---r_magn:60--> stade
ballon ---r_holo:50--> stade
foot ---r_associated:50--> stade
pelouse ---r_meaning:50--> stade
sport ---r_associated:50--> stade
sports ---r_associated:50--> stade
terrain ---r_hypo:50--> stade
terrain ---r_holo:50--> stade
but (sport)---r_associated:25--> stade
degré ---r_syn:25--> stade
phase ---r_syn:25--> stade
...

Figure 5: Parts of the relations acquired for the term *stade* (*stadium*, *stage*). The first relations presented are those for which the term *stade* is the origin, then those for which the term *stade* is a goal. A relation is typed ("action place", "associated idea", "has part"…) and has weight.

We could have planned to record all the answers, from the beginning of the game, with their frequencies. Our base would have increased much more quickly, but it would have been to the detriment of its quality. The interest of our solution is to limit in a much more drastic way the "fanciful" answers or the errors due to a bad understanding of the instruction, even of the term T itself. The emergence of "original" solutions will be slower, but it will be made all the same, after eliminating the most common solutions, thanks to the process of the "taboo" terms. Indeed, when a relation *term T → proposed term* is made by a large number of couples of players, it becomes commonplace or taboo; it is displayed at the same time as the term T, so that the players do not suggest it anymore. Thus, players are brought to make other propositions, generally more original ones. This facilitates the emergence of rarer relations, but less the emergence of errors.

Even when a relation becomes taboo, its weight is not fixed, but it just evolves more slowly as this relation is less often proposed by the players. It is quite interesting that the weight of the relation continues to evolve. Indeed, after a while, for the same term several relations can be taboo. If they had 'level up' and had the same weight, we would lose the information about the relative strength of the relations.

The approach presented here complements the one developed by (Zock and Bilac 2004) and (Zock and Schwab 2008) who tried to create an index based on the notion of association to attend the navigation, to help a person find a word which they have "on the tip of their tongue". Their approach is of the bottom-up type: they know the terms (co-occurrences found in a corpus), but not the "nature" of the link; this one must be inferred, which is far from being ordinary. In our case, we know one of the two terms (the origin term) as well as the type of the relation (imposed by the instruction given to the players). We look for several second terms (the target terms). Our approach is rather of the top-down type.

In a similar way to JDM, a second game named PtiClic, presented by (Lafourcade and Zampa 2009) and more recently by (Zampa and Lafourcade 2010), takes place between two players in an asynchronous way. A target term T, origin of relations, as well as a cluster of words resulting from terms connected with T in the lexical network produced by JDM are proposed to a first player. Several instructions corresponding to types of relation are also displayed. The player associates words of the cluster with instructions he thinks correspond by a drag-and-dropping. The same term T, as well as the same cluster of words and the same instructions, are also proposed to a second player. According to a principle similar to that set up for JDM, only the propositions common to both players are taken into account, thus strengthening the relations of the lexical network. Contrary to JDM, the players of PtiClic cannot suggest new terms, but are forced to choose among those proposed. This choice of conception is meant to reduce the noise due to misspelt terms or to the confusions of meanings. Figure 6 shows an example of a displayed screen, with a cluster of words connected with the term *vache* (*cow*) in the lexical network and several types of relation.

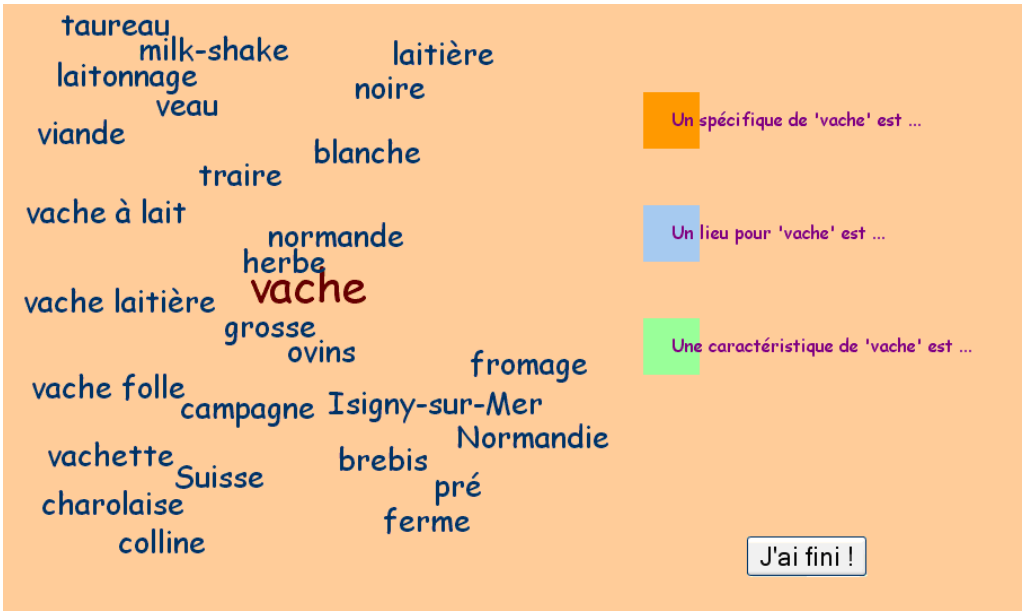


Figure 6: Example of a PtiClic screen displayed for the term *vache* (*cow*). The user is asked to drag and drop the blue terms in the proper targets when this is relevant. For example *vache laitière (milk cow)* could be dragged to *"Un spécifique de vache" (a particular type of cow).*

The collaborative building of resources by non-experts may induce some errors. In fact, as one may expect, we detected some of them, such as classical orthographic mistakes (eg: *théatre* for *théâtre*) or traditional confusions (eg: French singer *Dalida* with the biblical character *Dalila*)… These well-known mistakes are relatively rare and they can be manually detected.

According to the JeuxDeMots Web site, at the time of the writing of this paper, the lexical network contains more than 1.000.000 (one million) relations linking 221.000 terms. Around 800.000 games (with an average time of 1 minute per game) have been played corresponding to approximately 13.000 hours (about 550 days) of cumulative play.

3. Word usage determination

3.1. General principle

Perhaps naively, it is possible to consider that if a term T is polysemous, the terms which are connected with it form several groups, each of which being related to a word usage of T. We are making here the distinction between the notions of *word usage* and *meaning*, although clear cut definitions may be difficult. Word usages have a broader scope, they refer to the context and would certainly include meanings, but not the other way round as a given meaning may be associated to more than one usage. For example, in French, *sapin* (*fir*) besides being the tree, has a strong usage related to Christmas (*fir* as *Christmas tree*) and this usage tends to become autonomous. A putative definition of a word usage could be a *specific meaning in a given context, popular enough to be spontaneously given by someone.* Thus, the notion of word usage is much more accurate and relevant than the notion of meaning (or sense) which, as shown by (Véronis 2001), is relatively poor when we refer to the traditional dictionaries or to resources as WordNet. Word usage is thus a more fruitful notion than the meaning in NLP especially when we deal with semantic analysis and Word Sense Disambiguation. To give another example, *aile-plume-oiseau (wing-feather-bird)* and *aile-poulet-cuisse (wing-chicken-leg)* constitute two different word usages for the term *aile (wing)*, while they are obviously making reference to the same meaning. So, an adequate resource (at least for WSD) would then mostly refer to word usages than word senses (in the classical meaning of dictionaries) and would also try to evaluate the proper popularity level (or simply weight) of a usage. The weight is compulsory if we want to be able to fall back to a default usage when the context is not strong enough to allow a confident choice amongst possible usages.

As a very spontaneous usage of a term would have a higher weight than a rare one, we can then expect a quite large discrepancy between a proper inventory of usages and inventory of meanings (in dictionaries). Some meanings would be absent from a *popular meaning inventory* when too technical, too rare or simply unknown by the vast majority of people.

Our approach is globally based on an infinite iteration loop. Starting from the lexical network obtained with JeuxDeMots, we focus on terms and their neighborhood and compute the (sub)graph cliques for each term. The cliques obtained are then classified through a bottom-up hierarchical algorithm. The tree we obtain is a form of decision tree made of the original cliques (the leaves) and aggregated ones, the root being the term considered. A top-down process tries then to give a name to the cliques, in the same way that meanings in a bilingual dictionary are associated to some glosses to allow the reader to identify the meanings considered. The named (or labeled) tree is then pruned so as to keep the most probable ones leading to refinement (ie. usages) of the word considered. Those refinements are then injected in the lexical network.

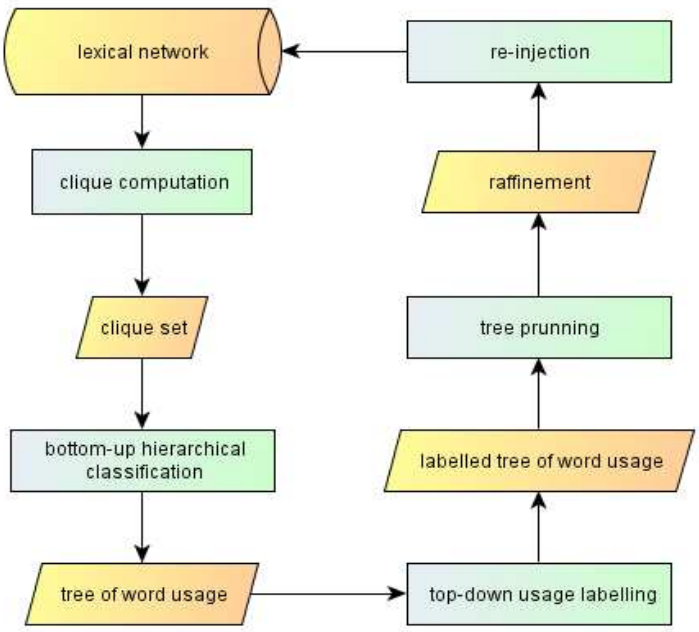


Figure 7: Overall looped processing for the usage determination of a given word in the lexical network.

In what follows, we present each of these steps, with coming first some details about the clique computation and similarity.

3.2. Clique computation

What is a clique? It is a set of terms constituting a fully connected subgraph in the lexical network (all nodes of the clique are connected to one another). These cliques are maximal as there is no other subgraph included. The terms T_{i1}, … ,T_{im} constitute the i^{th} word usage of T if the (m+1) * m / 2 relations between these (m+1) terms exist. So, a term T_j does not belong to this i^{th} word usage of T if at least one of the relations between this term T_j and one of the terms T_{i1}, … ,T_{im} does not exist. Figure 8 shows an example of clique determination.

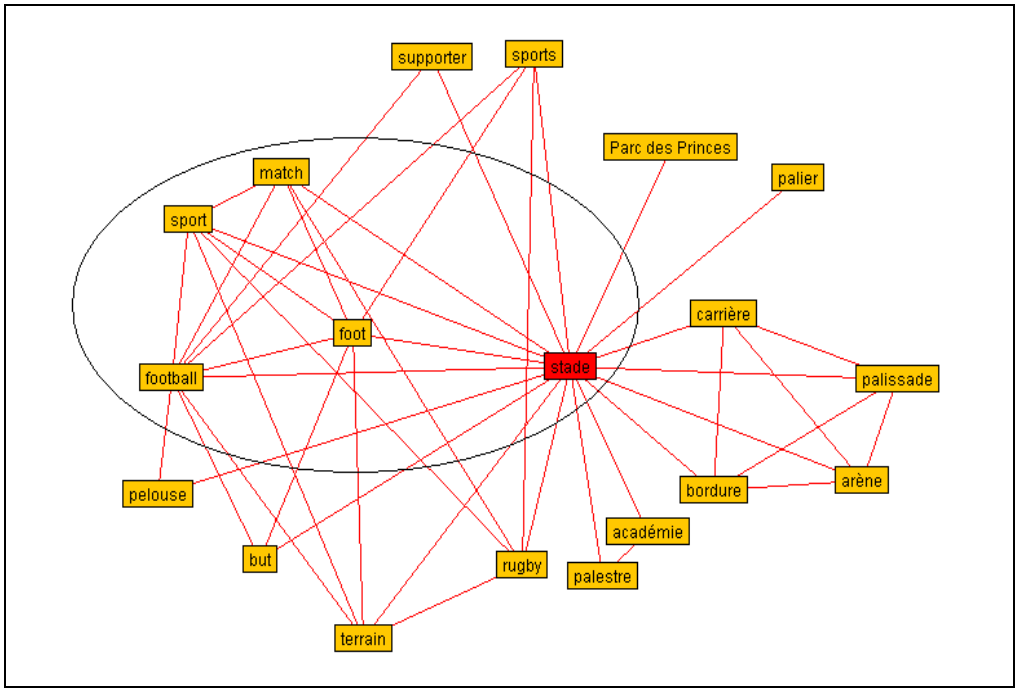


Figure 8: Example of a clique graph for the term *stade* (*stadium*, *stage*) showing the clique {*stade*, *match*, *sport*, *football*, *foot*} (*stadium*, *match*, *sport*, *football*, *foot*)

Similar approaches like (Ji et al. 2003) have been conducted on contexonyms but on resources trained on very large corpora (and again not extracted from people). The main difference with the work presented by (Ploux and Victorri, 1998) is that our lexical network is not resulting from corpuses but from users' activity. Moreover, we do not consider only synonymy type relations, but several types of relation. Doing so, we detect more usages as they do not always translate into various synonyms. For example, *poisson (fish)* has two meanings in our network: *animal (animal)* and *nourriture (food)*; this type of distinction does not appear when considering only synonyms.

For the clique identification, we take into account all relation types available in the lexical network. Of course we might consider that they do not contribute equally to the induced word usage, but the principle of JeuxDeMots induce that the most important relations to have the highest weights and that the most important relation types (for a given term) are the most populated. So, there is, *a priori*, no need to stress on specific relation type, as this information is already implicitly present in the network. Estimating the relevance of a usage consists in obtaining a measure of its importance both in terms of frequency and of lexical coverage. Considering the principle of the weighting of the relations, the weight of a usage is correlated with the weights of the relations between the terms of the clique which characterizes this usage. We have the following notations:

- C is a given clique for the term T (this is a set of terms and relations).
- C_{all} is the union of all C (that can be seen as the full pseudo clique for T).
- W(C) is the sum of the weights of the relations between the terms of C.
- Card(C) is the number of terms in C.

So, for a clique C related to the term T, we define formally the *relevance* as:

$$Rel(C) = W(C) * \log(Card(C_{all})/Card(C))$$

The *Rel* measure for a clique is nothing more than an adaptation of the tf-idf measure as defined by (Salton and McGill 1983) where, for the sake of simplicity, we have not divided by $W(C_{all})$. If there is only one clique, the relevance is equal to 0 and of course in that case we consider that there is only one usage. Figure 9 and 11 present the obtained usages for the term *stade (stadium, stage)* and *sapin* (*fir*), and the relevance of each of their usages. The REL value stand for the relevance (or more straightfully the weight) of the clique. The higher it is, the stronger the usage is.

0: 'stade' 'stade de rugby' REL = 33
stadium, rugby stadium
1: 'stade' 'Parc des Princes' REL = 38
stadium, Parc des Princes
2: 'stade' 'pelouse' 'terrain' 'football' REL = 177
stadium, field, ground, football
3: 'stade' 'lice' 'bordure' 'carrière' 'palissade' 'cirque' 'arène' REL = 48
stadium, lists, edge, quarry, boarding, cirque, arena
4: 'stade' 'piste' 'carrière' 'cirque' 'arène' REL = 78
stadium, track, quarry, cirque, arena
5: 'stade' 'échelon' 'palier' 'étape' 'phase' 'niveau' 'période' 'degré' REL = 68
stage, grade, stage, step, phase, level, period, degree
6: 'stade' 'but' 'football' 'foot' REL = 176
stadium, goal, football, foot
7: 'stade' 'but' 'point' REL = 38
stadium, goal, point
8: 'stade' 'point' 'degré' REL = 27
stage, point, degree
9: 'stade' 'supporter' 'football' REL = 98
stadium, supporter, football
10: 'stade' 'terrain' 'football' 'sport' 'foot' REL = 280
stadium, ground, football, sport, foot
11: 'stade' 'match' 'football' 'sport' 'foot' REL = 269
stadium, game, football, sport, foot
12: 'stade' 'football' 'foot' 'sports' REL = 142
stadium, football, foot
13: 'stade' 'rugby' 'sports' REL = 92
stadium, rugby, sports
14: 'stade' 'terrain' 'sport' 'rugby' REL = 250
stadium, ground, sport, rugby
15: 'stade' 'match' 'sport' 'rugby' REL = 240
stadium, game, sport, rugby
16: 'stade' 'gymnase' 'palestre' 'académie' REL = 39
stadium, gymnasium, palestrium, academy
17: 'stade' 'gymnase' 'sports' REL = 54
stadium, gymnasium, sports

Figure 9: The cliques found for the term *stade*. There are 18 on them, mostly related to *sport*, but also to *stage*.

The associations with *sapin (fir)* are mainly with the *tree* and *Christmas* semantic field. We can notice here that sapin is strongly associated to *pomme de pin (fir cone)*, but not the other way round (and thus does not take part in the clique computation). This might seem strange, but the reason is that *pomme de pin (fir cone)*, has never been selected for playing.

111 relations ==>	64 relations <==
sapin ---r_associated#0:200--> Noël	montagne ---r_lieu-1#28:620--> sapin
sapin ---r_associated#0:180--> arbre	forêt ---r_associated#0:180--> sapin
sapin ---r_isa#6:150--> arbre	Noël ---r_associated#0:170--> sapin
sapin ---r_associated#0:130--> forêt	bûche (gâteau) ---r_associated#0:120--> sapin
sapin ---r_has_part#9:130--> branche	arbre ---r_associated#0:110--> sapin
sapin ---r_has_part#9:130--> tronc	arbre ---r_hypo#8:110--> sapin
sapin ---r_associated#0:120--> cercueil	conifère ---r_syn#5:80--> sapin
sapin ---r_associated#0:115--> fiacre	sève ---r_holo#10:80--> sapin
sapin ---r_antimagn#21:110--> petit sapin	
sapin ---r_antimagn#21:110--> pin	

sapin ---r_antimagn#21:100--> pomme de pin
sapin ---r_associated#0:90--> guirlande
sapin ---r_associated#0:90--> pin
sapin ---r_isa#6:90--> conifère
sapin ---r_hypo#8:90--> pin
sapin ---r_hypo#8:90--> sapin de Noël
sapin ---r_has_part#9:90--> racine
sapin ---r_antimagn#21:90--> aiguille
sapin ---r_associated#0:80--> bois
sapin ---r_associated#0:80--> noël
sapin ---r_associated#0:80--> épines
sapin ---r_antimagn#21:80--> conifère
sapin ---r_raff_sem#1:75--> sapin (arbre)

sapin ---r_associated#0:70--> aiguille
sapin ---r_associated#0:70--> décoration
sapin ---r_associated#0:70--> résineux
sapin ---r_associated#0:70--> vert
sapin ---r_isa#6:70--> bois
sapin ---r_isa#6:70--> résineux
sapin ---r_isa#6:70--> végétal
sapin ---r_hypo#8:70--> sapin de noël
sapin ---r_has_part#9:70--> racines
sapin ---r_locution#11:70--> bois de sapin
sapin ---r_locution#11:70--> forêt de sapins
sapin ---r_locution#11:70--> sapin de Noël
sapin ---r_associated#0:60--> boule
sapin ---r_associated#0:60--> boules
sapin ---r_associated#0:60--> conifère
sapin ---r_associated#0:60--> guirlandes
sapin ---r_associated#0:60--> montagne
sapin ---r_has_part#9:60--> aiguilles
sapin ---r_holo#10:60--> Noël
sapin ---r_antimagn#21:60--> arbre
sapin ---r_associated#0:50--> branches
sapin ---r_associated#0:50--> fête
sapin ---r_associated#0:50--> hiver

...

cadeau ---r_lieu#15:70--> sapin
enluminé ---r_associated#0:70--> sapin
noel ---r_associated#0:70--> sapin
Noël ---r_has_part#9:60--> sapin
bois de chauffage ---r_isa#6:60--> sapin
boule ---r_associated#0:60--> sapin
conifère ---r_associated#0:60--> sapin
guirlande ---r_associated#0:60--> sapin
noôl ---r_associated#0:60--> sapin
noël ---r_associated#0:60--> sapin
résineux ---r_associated#0:60--> sapin
Père Noël ---r_associated#0:50--> sapin
aiguille ---r_associated#0:50--> sapin
aiguille ---r_lieu#15:50--> sapin
bois feuillus ---r_associated#0:50--> sapin
boules ---r_associated#0:50--> sapin
branche ---r_lieu#15:50--> sapin
cadeau de Noël ---r_associated#0:50--> sapin
cyprès ---r_associated#0:50--> sapin
décorations ---r_associated#0:50--> sapin
décorer ---r_patient#14:50--> sapin
forêt ---r_has_part#9:50--> sapin
fête ---r_associated#0:50--> sapin
montagne ---r_associated#0:50--> sapin
orme (arbre) ---r_associated#0:50--> sapin
pin ---r_holo#10:50--> sapin
résineux ---r_syn#5:50--> sapin
sapin ---r_lemma#19:50--> sapin
sapins ---r_lemma#19:50--> sapin
vert ---r_associated#0:50--> sapin
épine ---r_holo#10:50--> sapin
épine ---r_family#22:50--> sapin
sapin de Noël ---r_associated#0:35--> sapin aiguille

Figure 10: Some of the relations produced for *sapin (fir).*The strongest (first) association is with *Christmas*.

0: 'sapin' 'fiacre' REL = 52
fir, hansom
1: 'sapin' 'cercueil' REL = 55
fir, coffin
2: 'sapin' 'montagne' REL = 38
fir, mountain
3: 'sapin' 'épicéa' 'ginkgo' 'conifère' 'cèdre' 'mélèze' 'résineux' REL = 66
fir, spruce, ginko, conifer, cedar, larch,conifer
4: 'sapin' 'vert' 'arbre' REL = 126
fir, green, tree
5: 'sapin' 'épicéa' 'épinette' 'conifère' REL = 59
fir, spruce, spruce, conifer
6: 'sapin' 'aiguille' REL = 43
fir, needle
7: 'sapin' 'conifère' 'arbre' REL = 139
fir, conifer, tree
8: 'sapin' 'guirlande' 'Noël' REL = 111
fir, garland, Christmas
9: 'sapin' 'boule' 'boules' REL = 51
fir, ball, balls

10: 'sapin' 'boule' 'Noël' REL = 108
fir, ball, Christmas
11: 'sapin' 'Noël' 'sapin de Noël' 'sapin de noël' REL = 84
fir, Christmas, Christmas Tree, Christmas tree
12: 'sapin' 'Noël' 'fête' REL = 152
fir, Christmas, celebration
13: 'sapin' 'arbre' 'bois' 'forêt' REL = 219
fir, tree, wood, forest
14: 'sapin' 'conifères' REL = 71
fir, conifers

Figure 11: 16 cliques for the term *sapin* (*fir*) as found in the lexical network at the time of writing. The most relevant clique is {**sapin, arbre, bois, forêt}** *(fir, tree, wood, forest)* with a score of 219, followed by {**sapin, Noël, fête}** *(fir, Christmas, celebration)* with a score of 152.

3.3. Clique Similarity

The similarity between two objects can be defined according to (Tversky 1977) as being a function of their common characteristics with regard to all their characteristics. In NLP, we find several definitions of the similarity, for example (Manning and Schütze 1999), or more recently (Fairon and Ho 2004). In our case, it corresponds to the ratio between the weight of the relations connecting two cliques and the total weight of the relations on all the terms of these two cliques. We note W(E) the weight sum of the relations between the terms of the set E. The similarity between two cliques C_1 and C_2 will be equal to the *Jaccard* indice:

$$Sim(C_1,C_2) = W(C_1 \cap C_2) / W(C_1 \cup C_2)$$

We should note here, that the Jaccard indice is in our case applied on the set of relations and that the 'cardinality' of this set is the sum of the weights. Usually the Jaccard indice is applied on the true cardinality of the sets, considering equally all elements of the set. Figure 12a shows the similarities between the cliques of the term *sapin* (*fir*) in the form of a matrix.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0.04	0	0.08	0.04	0	0	0	0	0	0.01	0
3	0	0	0	1	0.31	0.66	0.06	0.54	0.02	0	0.02	0.02	0.02	0.2	0.36
4	0	0	0.04	0.31	1	0.29	0	0.75	0.03	0	0.03	0.03	0.03	0.68	0.15
5	0	0	0	0.66	0.29	1	0.09	0.56	0.03	0	0.03	0.03	0.03	0.16	0.48
6	0	0	0.08	0.06	0	0.09	1	0.05	0	0	0	0	0	0	0.33
7	0	0	0.04	0.54	0.75	0.56	0.0	1	0.03	0	0.03	0.03	0.03	0.64	0.46
8	0	0	0	0.02	0.03	0.03	0	0.03	1	0.14	0.69	0.6	0.77	0.01	0
9	0	0	0	0	0	0	0	0	0.14	1	1	1	1	0	0
10	0	0	0	0.02	0.03	0.03	0	0.03	0.69	1	1	1	1	0.01	0
11	0	0	0	0.02	0.03	0.03	0	0.03	0.6	1	1	1	1	0.01	0
12	0	0	0	0.02	0.03	0.03	0	0.03	0.77	1	1	1	1	0.01	0
13	0	0	0.01	0.2	0.68	0.16	0	0.64	0.01	0	0.01	0.01	0.01	1	0.06
14	0	0	0	0.36	0.15	0.48	0.33	0.46	0	0	0	0	0	0.06	1

Figure 12a: Similarity matrix between the cliques of the term *sapin* (*fir*).
The clique numbers are those from figure 11. For example, the similarity between cliques 4 (*fir*, *green*, *tree*) and 7 (*fir*, *conifer*, *tree*) equals 0.75. The matrix is symmetric. Values have been rounded after the second decimal.

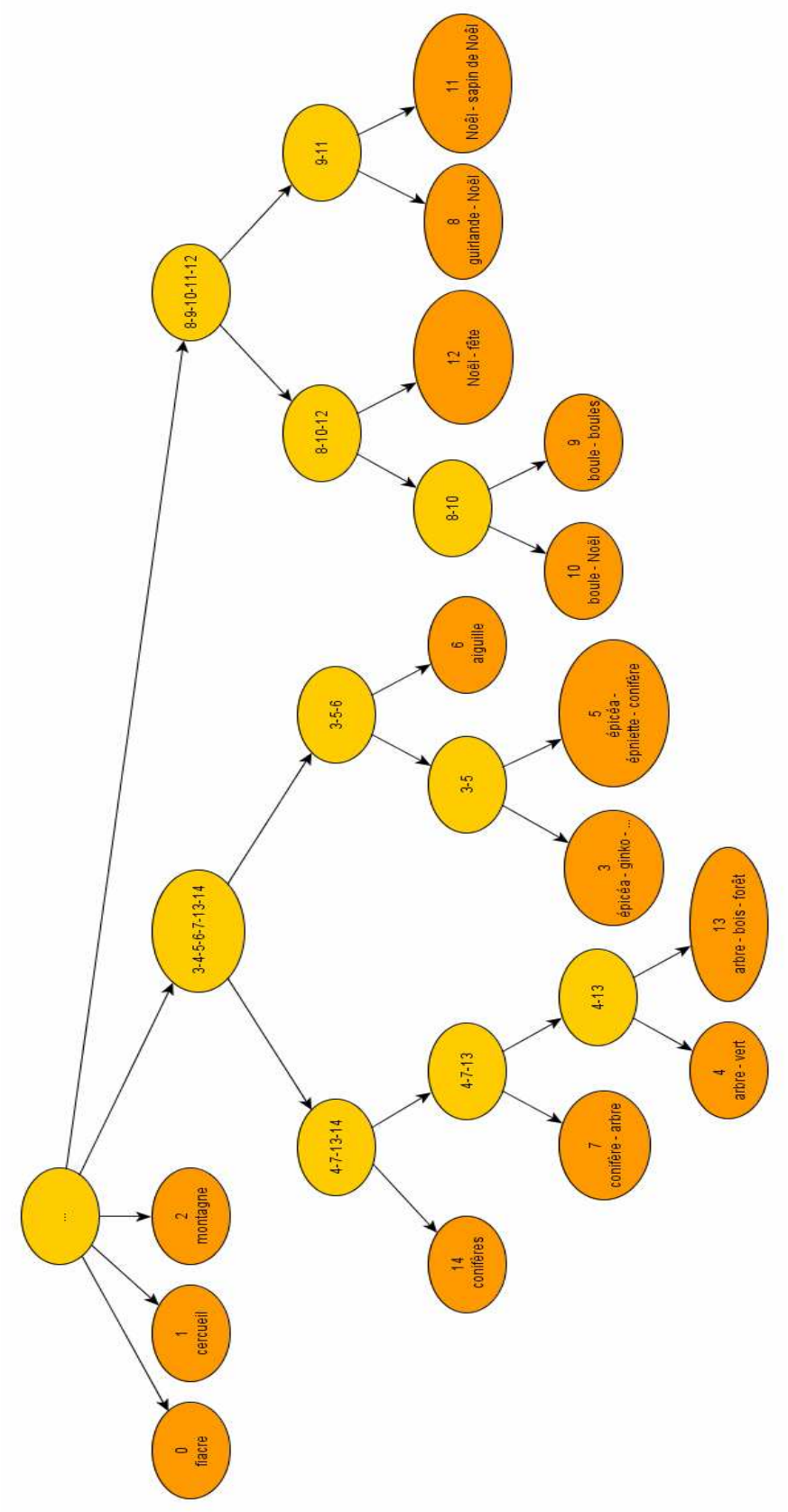


Figure 12b: The aggregate tree result of the bottom-up classification algorithm applied on *sapin (fir)*.

3.4. Construction of the word usage tree by bottom-up classification

Our aim is to obtain a representation of the various usages of a term T in the form of a tree, with the root grouping together all the meanings of T and the branches corresponding to its various usages. Generally, most terms possess several not separate cliques. In that case, the further away we go from the root of the tree, the more fine distinctions of usages we meet. In fact, we build the tree of the usages of a term T according to a "bottom - up" method: from all of its cliques, that is, from its leaves and going back up to its root which groups together all the meanings of T. For that purpose, we merge its cliques, two by two, beginning with those whose coefficient of similarity is the highest: thus, we build quasi-cliques representing groups of usages, close during the first fusions, less and less close during the successive fusions. The merging algorithm ends when all coefficients of similarity are below a given threshold close to zero (empirically set to 0.05 in our experiment) or when there is only one element left in the table.

The usage tree of a term is a structure expressing the refinements of its various meanings as deduced from the state of the lexical network. It thus constitutes a decision tree, a data structure which can be exploited for disambiguation. Furthermore, nodes of this tree are weighted (cf 3.7) allowing to identify usages that are the most common, which is both useful for guessing default cases and ordering usages from the most activated for people to the least activated.

The algorithm we use is a very classical agglomerative bottom-up approach. It is possible, because of two properties : we have a metric on clique (the similarity) and we are also able to fusion two cliques to produce a new (pseudo)clique (this is a classical alternative to the linkage criteria).

3.5. Labeling

Labeling the various nodes of the usage tree of a term is made during a width-first search, that is, according to a "top-down" method. The tree root is labeled by the term itself. Every node of the tree is labeled with a term stemming from the clique or the quasi-clique this node represents; the selected term is the one whose sum of the weights of its relations with the root term is the highest, after eliminating all the terms labeling the nodes of the tree situated in a depth lower than that of the concerned node. Thus, it is possible that a node cannot be labeled if all the terms which define it have already been used in the labeling of nodes previously done. In this case, this node, but also its brother and all its successors, are not labeled.

Figure 13 shows the tree obtained for the term *sapin* (*fir*). We can note here that there is a definite usage associated to *Noël (Christmas)*. The usage associated to *mountain* has been pruned, leaving us with four refinements at level one (see 3.6).

3.6. Tree pruning and refinement terms

The relevance of an inner node of the tree is computed as defined above considering the quasi clique obtained during the merging. We ignored the usages of the tree whose relevance is below a given threshold (empirically set to 50 in our experiment). This threshold corresponds to the configuration where both terms have been associated to each other with only two pairs of players (one pair for each direction) and maybe it could have been accidental. In figure 13, in the tree for the term *sapin*, we discard the *sapin (montagne)* node at level 1 whose relevance is only equal to 38 (clique number 2 in figure 11). The nodes for *sapin* such as *fiacre*, *cercueil*, *arbre* and *Noël* have a relevance of respectively 52, 55 and more than 300 for the last two. These last four nodes constitute four refinements for the term *sapin*. After their validation by an expert, such refinements are placed as new nodes in the lexical network. They may be thus suggested as origin terms in the JDM and PtiClic games, thus leading the users to create new relations outgoing from these refinement terms. On the other hand, when a JDM player proposes a destination term for which several refinements already exist, JDM displays these refinements for the player who is invited to choose one: this process leads to create incoming relations for refinement terms.

Figure 14 presents the labeled and pruned tree for *stade*. Two usages at level 1 have been dropped because they were not relevant enough.

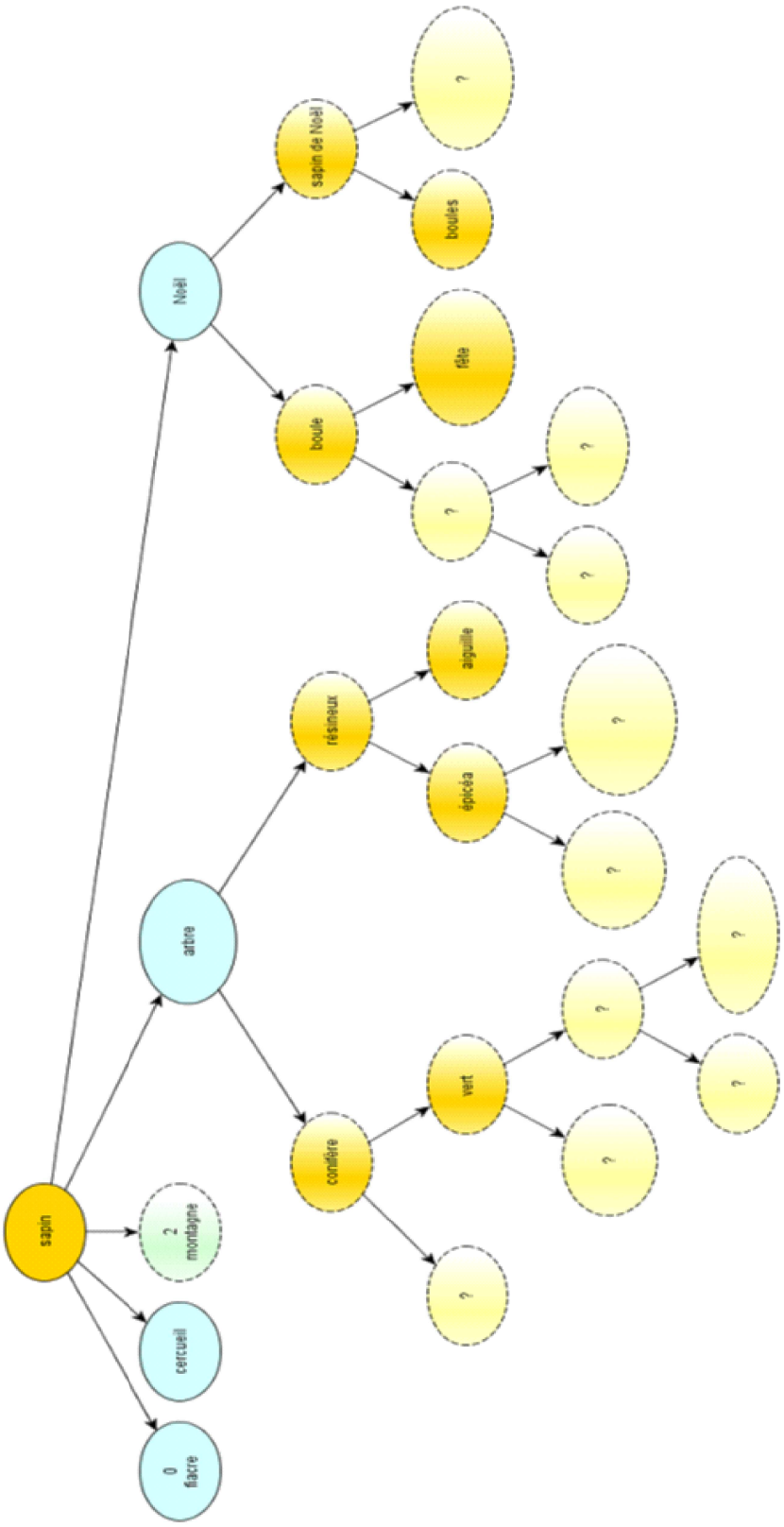


Figure 13: Labeled word usage tree for the term *sapin* (*fir*). The *montagne (mountain)* usage has been pruned because its relevance has been too low so far (it may change with the evolution of the lexical network). Nodes with '?' as their name, were not labeled, because of possible label shortage.

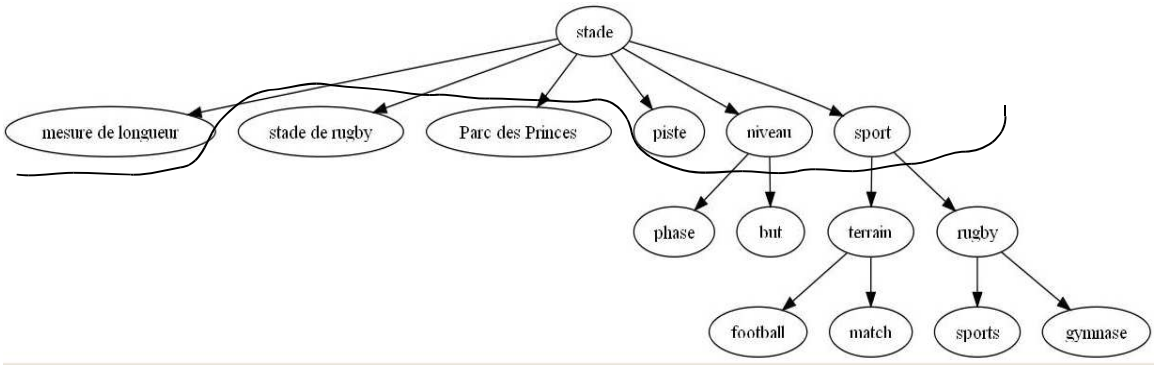


Figure 14: Labeled word usage tree for the term *stade* (*stadium, stage*). We pruned this tree by deleting nodes which could not be labeled or whose relevance was too low (below 50 in our experiment)

So, our model contains a double iterative process (as illustrated on figure 15). First, the players supply the database (the lexical network) furnishing relations and eventually adding new terms to the already existing base of terms; the weighting of the relations is computed according to the propositions of the players. Secondly, after the automatic construction of a word usage tree, an expert validates (or not) the meanings thus detected and so adds refinement nodes in the lexical network that can be proposed to the players. On a timely basis, the system recomputes the refinements for the terms in the database, and proposes new ones to be validated. Also, there is checked for the consistency between the new refinements and the previous ones (if they had been already computed beforehand). So far, the approach has been experimentally proven consistent and monotonous, as recomputation of refinement has never led to a set of refinement incompatible with a previous one.

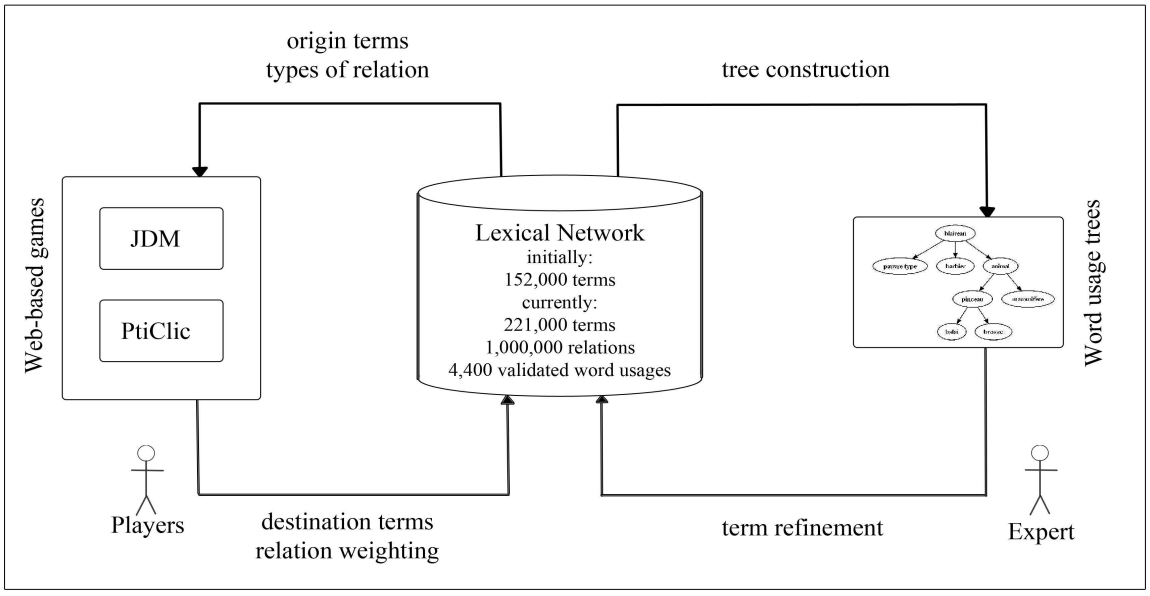


Figure 15: The double iterative process: the players add relations and possibly new terms, the expert validates refinements. Both actions increase the lexical network with relations and refinements.

3.7. Weighting usages

Considering the weighting of the relations in our lexical network, it is possible to define a relative weight for each meaning of a term. For that purpose, we consider only the cliques corresponding to nodes with a depth equal to 1 in the usage tree. The i^{th} meaning of the term T is represented by the clique C_i. This clique C_i can be a clique (effective clique and not a quasi-clique) initially detected in our lexical network: its relevance Rel (C_i) is then the one defined in section 3.2. More frequently, C_i is a quasi-clique resulting from the fusion of cliques

initially detected in the network: in that case, we define its relevance Rel (C_i) as being the sum of the relevances of the initial cliques whose fusion it is. The relative weight of the i^{th} meaning of the term T, represented by the clique C_i, will have for value the ratio between the relevance of the clique C_i and the sum of the relevances of all the cliques representing the different meanings of the term T (equal to the sum of the relevances of the initial cliques of T, implemented as leaves in the usage tree). This relative weight can be written:

$$RW(Ci) = Rel(C_i) / \sum_{j=1,n} Rel(Cj)$$

where *Rel (C)* corresponds to the relevance of the meaning represented by the clique C
and *n* to the number of meanings T possesses; thus, *n* is the number of nodes with depth 1in the usage tree of T.

More generally, the relative weight of a clique can be defined as being the ratio between the sum of the relevance of the cliques which it is the fusion and the sum of the relevance of the cliques which are leaves in the usage tree.

Let us bear in mind that the relevance of a clique cannot be an absolute value: it depends inevitably on the number of times the term T was played during the construction of our lexical network. The weighting we define here can thus be only a relative weighting. Figure 16 shows the labeled usage tree for the term *palais (palace)*, with each node's relative weight.

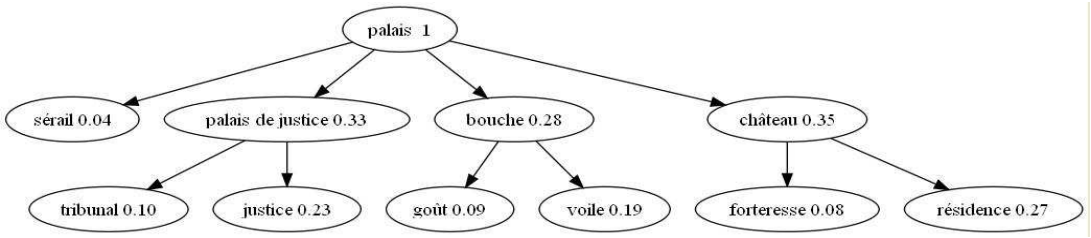


Figure 16: Word usage tree named for the term *palais (palace)* also showing the relative weight of every clique, and in particular each of the meanings (or lexical fields) named: *sérail (seraglio)*, *palais de justice (law courts)*, *bouche (mouth*, for *palate)* and *château (castle)*.

4. Evaluation

4.1. Obtained Word Usages

Regularly, we examine the highly polysemous terms, and we ask an expert lexicographer to validate the meanings detected by JDM, after we pruned the usage tree. The role of the expert does not extend beyond this validation: he doesn't add any meaning, even if he thinks some are missing, and he removes only those that obviously correspond to mistakes. As the time of writing this article, we thus obtain 4412 validated word usages for 1263 terms, which correspond to a mean of around 3.5 usages per term. Out of these 1263 terms more than 80% are labeled as very common terms (at least one meaning should be known at the age of 12). The terms are mostly common nouns also some usages are tagged with other part-of-speech like verb or adjective. For example *dîner (dinner, to have dinner)* in French can be at the same time a noun or a verb.

4.2. User Evaluation

Evaluating the quality of such word usages is difficult, specifically in the absence of adequate gold standard. As far as we aware of, there is no such resources for French. So, we decided to make a user based evaluation, trying to access qualitatively and quantitatively the word usages we obtained so far. We undertook the evaluation only at the first level of the usage tree computed for a given term. We should note however, that a large scale evaluation is currently in progress (Joubert et al. 2011). It uses a game of riddles, called AKI: the system, using the lexical network, tries to guess a term thanks to indications supplied by the user. AKI can be also considered as a TOT software (to find the word on the Tip Of the Tongue). So far, AKI can find out in about 75% of the case for any kind of terms (event Proper Noun, recent terms, etc.). For general vocabulary, AKI reaches around 99% of success (6 failures for 500 tries) where people under the very same conditions achieve about 80% of success.

We based our evaluation on naive users (i.e. not lexicographers) for two reasons. First, finding lexicographers for this task is not easy to say the least and certainly not more than few ones. Secondly, we wanted to have an evaluation confronting common people to our data. The idea is to identify word usages the same way (as a result at least) an average person would do. We remind here that the JDM lexical network does not aim at being more than an average representation of associations between terms.

We asked 30 non-expert persons to undertake four slightly different tasks. Given a word and the set of associated named usages, they had to evaluate the number of missing usages and the number of supernumerary usages (either too specific or plain wrong). The task has to be done on two sets of different 50 words. From the first set, persons are not allowed to consult any dictionary (task Dict-); from the second set, they can check on dictionaries if they want to do so (task Dict+). The dictionary we proposed for reference is Wiktionary for French, but users were allowed to use any resources they want. Furthermore, the set of words is either taken from common words or non-common words.

An example of what is asked to the people is the following (translated for the purpose of this article):

> For the word *sapin*, we propose the following usages:
>
> - sapin(arbre) *(fir - tree)*
> - sapin(Noël) *(fir - Christmas)*
> - sapin(cercueil) *(fir - coffin)*
> - sapin (fiacre) *(fir - hansom)*
>
> Do you think some usages are missing, if yes how many ? Do you think some proposed usages are inappropriate, if yes how many ?

Word usages are ordered by decreasing relevance (as defined in section 3.2).

The four sets of words proposed to each evaluator are completely random (although verifying the constraints described before, that is to say either common or uncommon ones) and they are distinct (a given word may be present only in one set). Two different evaluators may have no distinct sets. The evaluators have all an age above 20 and had a similar proportion of 17 females and 13 males. The level of education was basically 2 years of university or more. The following tables present the collated results of this evaluation.

Common words	**Dict -**	**Dict +**
Missing usages	0.45	1.52
Added usages	0.66	0.37

Uncommon words	**Dict -**	**Dict +**
Missing usages	0.25	1.67
Added usages	0.76	0.28

For users, added usages are those considered wrong (or at least far fetched). Missing usages are those which should have been present.

4.3. Result Analysis

How can we interpret those results? Without dictionary there is systematically less than one usage felt as missing or added. For the Dict- task, by debriefing the evaluators it appears that the added usage is quite often a proper usage that was unknown to the user (technical, old or rare). Conversely in the Dict+ task, the missing usages value rises as more usages, unknown to the user and found in the dictionary. It seems that globally we are missing much more usages than adding wrong ones. This is quite inline which the way the lexical network is constructed (by players indirect contribution). Missing usages are those quite specific, rare and basically unknown to users.

The task on uncommon words tends to strengthen this analysis. Indeed, without any dictionary people feel that they are more added usages than with common words and less missing usages. The result of the Dict+ is contravariant with the Dict-. Indeed, missing usage value rises and added usage value diminishes.

We can take several precise examples for illustrations. For the word *sapin* (a common word) we got :

sapin	**Dict -**	**Dict +**
Missing usages	0.5	0.9
Added usages	0.2	0.4

The wiktionnary definitions are: **sapin** *masculin*

1. (Botanique) Arbre conifère résineux de la famille des abiétinées à aiguilles persistantes, au tronc droit, dont le fruit est un cône.
2. Bois de cet arbre utilisé en menuiserie.
3. (Par métonymie) Cercueil.
4. (Familier) (Vieilli) Fiacre.

Some people were doubting about the *sapin (fiacre)* usage although this is a correct one. The usage of *sapin* as *wood (mater)* is missing but roughly only one person out of two have been thinking of it without checking in a dictionary. The added usage compared to the dictionary is the *sapin (Noël)* although it is present as a locution.

For the word *frégate*, we found out the following usages:

- frégate (navire) *(frigate boat)*
- frégate (oiseau) *(frigate bird)*

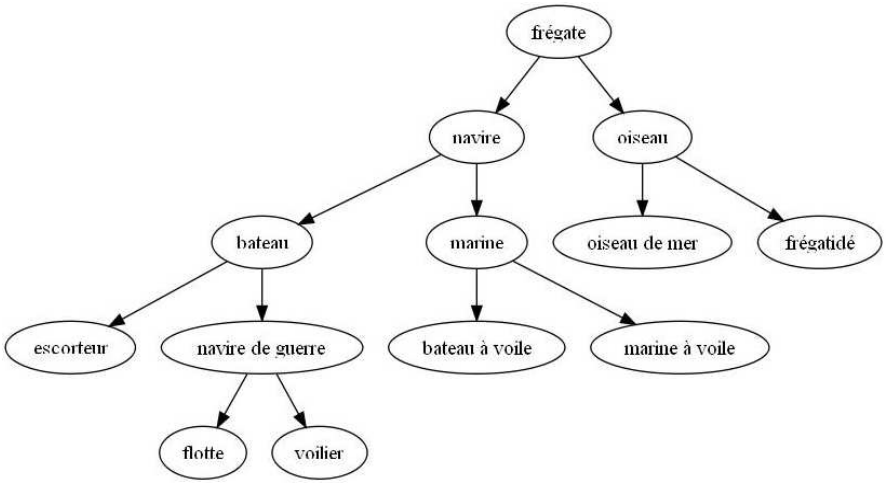


Figure 17: Usage tree for the term *frégate*.

frégate	**Dict -**	**Dict +**
Missing usages	0.15	0
Added usages	0.1	0

The wiktionnary definitions are: **frégate** *féminin*

1. (Histoire) (Marine) (Militaire) Bâtiment de guerre qui n'avait qu'une seule batterie couverte et qui portait de vingt à soixante bouches à feu.
2. (Zoologie) Oiseau de mer palmipède, d'une très grande envergure, et qui saisit à la surface de l'eau les poissons dont il se nourrit.

For *frégate*, some people made the distinction between the *ancient boat* and the *modern boat*. On a rare occasion the evaluator was doubtful on the bird meaning. Comparing with the dictionary we got an exact match.

For the word *blaireau*, we found out and proposed the following usages:

- blaireau (animal) *(badger)*
- blaireau (pauvre type) *(dork)*
- blaireau (barbier) *(shaving brush)*

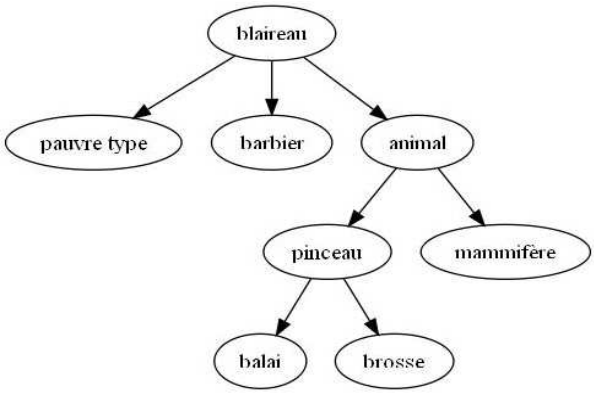


Figure 18: Usage tree for the term *blaireau*.

blaireau	**Dict -**	**Dict +**
Missing usages	0.3	1.1
Added usages	0	0

The wiktionnary definitions are: **blaireau** *masculin*

1. Mammifère omnivore, bas sur pattes, au pelage noir, gris et blanchâtre, qui se creuse de profonds terriers.
2. (Arts) Brosse en poils de cet animal dont se servent les peintres et les doreurs.
3. Pinceau garni de ces poils dont on se sert, en se rasant, pour étaler et faire mousser le savon.
4. (Argot) (Vieilli) Nez.
5. (Argot) Individu grossier et antipathique ; imbécile, idiot.

For *blaireau* some people found that the *(painting) brush* (meaning 2 from Wikipedia) is indeed missing. Most people missed the *nez* (nose – meaning 4) meaning, which is quite old.

All in all, although being preliminary, those results are very encouraging, both on the soundness of the method for determining and naming word usages and the quality of the resource collected so far (although evaluating this resource was not the primary goal of this paper). They seem to correspond to what people know and not specifically to some resources made by lexicographers or experts.

What is the effect of the label on the evaluation? If for a given word usage a different label would have been chosen to which extend the results might be modified? In fact there is no much choice for a reasonable label, and generally choosing a substitute like an hyperonym (for example animal instead of bird, in case of frigate) does not alter the results. Of course, this is less and less true as we go deeper in the tree, but also there is less and less choices (if we stick to our labeling approach described in 3.5). Perhaps a deeper evaluation on this particular point should be conducted.

What can we do with the unlabeled usages? So far the answer is simple: nothing. But we should keep in mind that the network is in constant evolution and that some cliques existing now may be fusionned in the near future due to the players' activity, or on the contrary being reinforced with new terms allowing them to be labeled.

5. Conclusion

JeuxDeMots is an on-line game on the Web whose objective is the construction of a lexical network, as such as a game with a purpose. Making a game was justified by the assumption that it will attract a lot of people of various horizons permitting to construct data in cheap, fast and reliable way. The emergence of labeled and weighted relations between terms is made through the gaming activity of a large number of users which cooperated indirectly. These users are not certainly linguists, but we strongly believe that both their number and variety will allow to obtain a lexical network with a satisfactory coverage and precision for general knowledge. This process relies on the principle enunciated by (Fisher 2011) that affirms an estimation given by a group of persons is generally better than everyone's estimation. Our purpose is not the constitution of an experts' database or a strictfull ontology on some domain, but rather representing a common general knowledge.

About the results first obtained on word usage trees, they seem to correspond in their main structures to those a human non-expert would build. In particular, the main branches, directly stemming from the root, correspond in the majority of cases to the meanings of the root term as we could find them in a dictionary. These main branches are subdivided into sub-branches which are so many refinements in the usages. In their detailed structures however, we notice elements that are different from what a human would have written as we showed on examples above. Are these differences (can we really speak about abnormalities?) due to our method of construction of the trees of the labeled usages with help of players who are not experts, or are they due to the fact that the lexical network is not "complete" enough yet?

So far, the usage identified by our system are first validated by hand before being actually inserted in the lexical network. One may ask if a fully automated processing including validation can be devised ? It might be doubtful while the question is still open. But, some way of making the users validate (or invalidate) proposed refinements might be the path to follow. This is essentially the same as an expert validation bit viewed in a distributed and collaborative way. A perspective of our work is to strongly validate that the insertion of the identified word usages in the lexical network and proposing them to the players has some virtuous properties. Hence, the word usages are going to be associated to other terms of lexical database, amongst then other word usages, leading incrementally to a lexical network between word usages. That is to say obtaining this way, a lexical network between less ambiguous terms. It would be interesting to assess whether or not reapplying our algorithm leads to some convergence as expected. The results observed so far seem to indicate this is the case.

References

vonAhn L. and Dabbish L. (2004) « Labelling Images with a Computer Game », *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 319-326.

vonAhn L. (2006) « Game with a purpose », *IEEE Computer Magazine*, 39(6), Vienna, pp. 96-98.

Collins A. and Quillian M.R. (1969) « Retrieval time from semantic memory », *Journal of verbal learning and verbal behaviour* 8 (2), pp. 240-248.

Dumais S.T. (1994) « Latent Semantic Indexing (LSI) and TREC-2 », *The Second Text REtrieval Conference*, National Institute of Standards and Technology Special Publication, vol 500, n°215, pp. 105-116.

Fairon C. and Ho N.D. (2004) « Quantité d'information échangée : une nouvelle mesure de la similarité des mots », *Journées internationales d'Analyse statistiques des Données Textuelles (JADT)*, Louvain-la-Neuve (Belgique)

Ferret O. (2002) « Using Collocations for Topic Segmentation and Link Detection », *Proc. of the Coling Conference on Computational Linguistics*, Taipei, pp. 261-266.

Fisher L. (2011) « L'intelligence collective », *Cerveau et Psycho*, 43, pp. 53-57

Gaume B. (2006) « Cartographier la forme du sens dans les petits mondes lexicaux », *Journées internationales d'Analyse statistiques des Données Textuelles (JADT)*, Besançon, France, pp. 451-465.

Gaume B., Duvignau K. and Vanhove M. (2007) « Semantic associations and confluences in paradigmatic networks », in : *Typologie des rapprochements sémantiques*, M. Vanhove éd.

Ji H., Ploux S. and Wehrli E. (2003) Lexical knowledge representation with contexonyms. In *Proceedings of the 9th MT summit*, pp. 194-201

Joubert A., Lafourcade M., Schwab D. and Zock M. (2011) « Evaluation et consolidation d'un réseau lexical grâce à un assistant ludique pour le "mot sur le bout de la langue" », *Conférence sur le Traitement Automatique des Langues Naturelles (TALN'11)*, Montpellier (to be published)

Lafourcade M. and Joubert A. (2008) « Détermination des sens d'usage dans un réseau lexical construit à l'aide d'un jeu en ligne », *Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08)*, Avignon, pp.189-199.

Lafourcade M. and Zampa V. (2009) « JeuxDeMots and PtiClic: games for vocabulary assessment and lexical acquisition », *Proc. of Computer Games, Multimedia & Allied Technology 09 (CGAT'09)*, Singapore.

Landauer T.K. and Dumais S.T. (1997) « A solution to Plato's problem : the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge », *Psychological Review*, 104, pp. 211-240

Lapata M. and Keller F. (2005) « Web-based Models for Natural Language Processing », *ACM Transactions on Speech and Language Processing*, vol.2, n°1, pp. 1-30.

Lieberman H., Smith D.A. and Teeters A. (2007) « Common Consensus: a web-based game for collecting commonsense goals », *International Conference on Intelligent User Interfaces (IUI'07)*, Hawaii, USA

Manning C.D. and Schütze H. (1999) *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge.

Marchetti A., Tesconi M., Ronzano F., Rosella M. and Minutoli S. (2007) SemKey: A Semantic Collaborative Tagging System, *Proceedings of WWW2007*, Banff, Canada.

Mel'čuk I.A. (1988) *Dictionnaire explicatif et combinatoire du français contemporain*, Les Presses de l'Université de Montréal.

Mel'čuk I.A., Clas A. and Polguère A. (1995) *Introduction à la lexicologie explicative et combinatoire*, Editions Duculot AUPELF-UREF.

Mihalcea R. and Chklovski T. (2003) Open Mind Word Expert: Creating Large Annotated Data Collections with Web Users' Help, *Proceedings of the EACL 2003 Workshop on Linguistically Annotated Corpora (LINC 2003)*, Budapest.

Miller G.A., Beckwith R., Fellbaum C., Gross D. and Miller K.J. (1990) « Introduction to WordNet: an on-line lexical database », *International Journal of Lexicography* 3 (4), pp. 235-244.

Pechoin D. (1991) *Thésaurus: Des idées aux mots, des mots aux idées*, Larousse, Paris.

Ploux S. and Victorri B. (1998) « Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes », *Traitement Automatique des Langues*, 39, n°1, 18 p.

Polguère A. (2003) *Lexicologie et Sémantique lexicale*, Les Presses de l'Université de Montréal.

Polguère A. (2006) « Structural properties of Lexical Systems: Monolingual and Multilingual Perspectives », *Proc. of the Workshop on Multilingual Language Resources and Interoperability* (COLING/ACL 2006), Sydney, pp. 50-59.

Robertson S. and Spark Jones K. (1976) « Relevance weighting of search terms », *Journal of the American Society for Information Science*, n° 27, pp. 129-146.

Roget P.M. (1852) *Thesaurus of English words and phrases*, Longman, London.

Salton G. and McGill M.J. (1983) *Introduction to Modern Information Retrieval*, McGraw-Hill.

Schwab D. and Lafourcade M. (2007) « Modelling, Detection and Exploitation of Lexical Functions for Analysis», *ECTI Journal*, 2007, vol.2, ISSN 1905-050X, pp. 97-108.

Siorpaes K. and Hepp M. (2008) « Games with a purpose for the semantic Web », IEEE Intelligent Systems, 23(3), pp.50-60.

Smadja F. (1993) « Retrieving collocations from text: Xtract », *Computational Linguistics*, 19, pp. 143-177.

Sowa J. (1992) *Semantic networks*, Encyclopedia of Artificial Intelligence, edited by S.C. Shapiro, Wiley, New York.

Spence D.P. and Owens K.C. (1990) « Lexical co-occurrence and association strength », *Journal of Psycholinguistic Research*, 19 (5).

Tversky A. (1977) *Features of similarity*, Psychological Review, 84, pp.327-352.

Véronis J. (2001) « Sense tagging: does it make sense? », *Corpus linguistics' 2001 Conference*, Lancaster, U.K.

Wandmacher T., Ovchinnikova E. and Alexandrov T. (2008) « Does Latent Semantic Analysis reflect Human Associations? », *Proc. of the Lexical Semantics workshop at ESSLLI'08*, Hamburg, Germany.

Wettler M. and Rapp R. (1992) « Computation of Word associations based on the co-occurrences of words in large corpora », *Proc. of the 1st Workshop on Very Large Corpora*, Academic and Industrial Perspectives, pp. 84-93.

Zampa V. and Lafourcade M. (2010) « PtiClic et PtiClic-kids: jeux avec les mots permettant une double acquisition », *Proc. of TICE2010, 7e colloque TICE*, Nancy.

Zock M. and Bilac S. (2004) « Word lookup on the basis of associations: from an idea to a roadmap », *Proc. of Coling workshop: Enhancing and using dictionaries*, Geneva, pp.29-34.

Zock M. and Quint J. (2004) « Why have them work for peanuts, when it is so easy to provide reward Motivations for converting a dictionary into a drill tutor », Papillon-2004, 5th workshop on Multilingual Lexical Databases, Grenoble.

Zock M. and Schwab D. (2008) « Lexical access based on underspecified input », COGALEX workshop, Coling, Manchester, pp. 9-17.

CHAPITRE 4

Analyse de textes et propagation

Ce chapitre présente trois approches pour une analyse thématique et sémantique de textes.La première approche s'appuie sur des vecteurs d'idées et un algorithme de propagation en remontée-descente sur un arbre morphosyntaxique calculé pour le texte. Si le texte analysé est une définition de dictionnaire, la structure produite (vecteur ou signature) peut être utilisée pour la construction d'un lexique sémantique. Si le texte est un document, la structure peut être utilisée comme élément d'indexation pour un moteur de recherche. Créer un index sur une collection de textes avec une structure délexicalisée, comme les vecteurs d'idées, permet d'effectuer des recherches thématiques sur cette collection. La seconde approche vise à extraire du texte des termes clés et/ou à en calculer des pertinents mais absents du texte. Il s'agit de construire une signature lexicale depuis le texte qui sera sa représentation thématique, mais contiendra également des vocables précis (comme des entités nommées). Enfin, nous présentons un modèle d'analyse sémantique de textes bioinspirée, fondé sur la mise en concurrence de colonies de fourmis artificielles, et sur la création de liens explicites entre les syntagmes au sein de la structure de calcul.

Articles joints

M. Lafourcade et Ch. Boitet (2002) *UNL lexical Selection with Conceptual Vectors.* In proc. of LREC'2002, Las Palmas, Canary Island, Spain, May 27, 2002, 7 p.

M. Bouklit et M. Lafourcade (2006) *Propagation de signatures lexicales dans le graphe du Web.* In proc of RFIA'2006, Tours, France, 25 au 27 janvier 2006, 9 p.

M. Lafourcade, F. Guinand (2010) *Artificial Ants for Natural Language Processing* in Artificial Ants. N. Monmarché, F. Guinand, P. Siarry Eds. Wiley ISBN : 9781848211940. pages 454-492.

Encadrement - Didier Schwab [Schwab, 2005], Thibault Zamora [Zamora, 2005], Sébastien Maranzana [Maranzana, juin 2007], Cédric Lopez [Lopez, 2009].

L'analyse thématique de textes vise à produire à partir d'un segment textuel une structure traduisant les idées majoritaires contenues dans ce segment. Par *segment*, nous entendons un texte pouvant aller du syntagme, à la phrase, jusqu'à un texte de longueur arbitraire. Dans ce qui suit, nous présentons trois types d'approches. La première, la *construction de vecteurs thématiques*, cherche à produire un ou plusieurs vecteurs d'idées pour le texte dans sa globalité, mais effectue également une désambiguïsation lexicale d'ordre thématique. La seconde approche, le *calcul de mots-clés*, vise à produire une signature lexicale thématique pour un texte. Nous insistons dans ce dernier cas sur

la notion de *calcul* et non pas uniquement d'*extraction*, car les termes de la signature ne sont pas nécessairement des termes présents dans le texte. Enfin, la troisième approche s'inspire de l'activité des colonies de fourmis pour mettre en œuvre un modèle visant à effectuer une analyse couplant désambiguïsation lexicale et construction de chemins interprétatifs.

4.1 Construction de vecteurs thématiques

Comment analyser un texte si nous avons besoin de vecteurs qui sont issus de définitions textuelles que nous devons analyser ? Avant de procéder au traitement des textes, un lexique associant termes et vecteurs doit être construit. En dehors d'une indexation manuelle complète, qui s'avérerait à la fois difficile et fastidieuse, un apprentissage supervisé peut être conçu. Le processus d'apprentissage est une tâche *sans fin* qui consiste à choisir de manière pseudo-aléatoire un mot qui doit être appris (ou révisé). Le vecteur de chaque définition de ce mot est alors le résultat d'un traitement sur la définition (dont le détail est décrit dans la suite). Nous obtenons alors un vecteur global, que nous qualifierons de *thématique*, dans la mesure où les activations relèvent des *idées* associées et non de relations ou fonctions lexicales particulières. Pour chaque définition d'acception d'un terme issu de dictionnaires à usage humain, nous pouvons calculer un tel vecteur d'idées et le stocker. Dans le chapitre 2, nous avons présenté l'architecture globale d'un tel système d'apprentissage/calcul de vecteurs d'idées à partir de définitions. Nous précisons ici les détails du calcul d'un vecteur d'idées à partir d'un texte.

L'idée de *bouclage* associé à un *affinage incrémental* des structures produites est une notion que nous plaçons à la base de la constitution d'une base lexicale, que ce soit sous la forme de vecteurs ou d'un réseau. Nous avons métaphoriquement illustré une telle idée à travers la figure 3.12 (en conclusion) en ce qui concerne les raffinements de sens, mais elle s'applique avec force également à l'analyse de texte.

4.1.1 Algorithme de remontée-descente

Comment pouvons-nous construire un vecteur conceptuel pour un texte donné ? Nous proposons ici une approche par *remontée et descente* de vecteurs sur un arbre d'analyse (analyse RD). À partir du texte, la première étape consiste à construire un arbre d'analyse morphosyntaxique. Il s'agit d'un arbre de dérivation (en constituants) dont les feuilles *reconstituent quasiment* la phrase originale. Une feuille réfère à un mot auquel sont associées une ou plusieurs définitions (trouvées dans les différents dictionnaires) et un vecteur conceptuel. Pour simplifier, nous ne considérons que les noms, verbes, adjectifs et adverbes, à l'exclusion des mots-outils. En grammaire, un mot-outil ou mot grammatical, appartient à une catégorie de mots tels que les articles et les prépositions, dont le rôle syntaxique induit en partie le rôle sémantique, mais n'a *a priori* qu'un impact thématique direct réduit. Après filtrage en fonction de l'accord avec les attributs morphosyntaxiques, un vecteur conceptuel *global non contextualisé* obtenu à partir des vecteurs de ses k définitions est attaché à la feuille. La façon la plus simple et directe pour le faire (mais pas la meilleure) est de calculer le vecteur moyen : $V(w) = V(w.1) \oplus \cdots \oplus V(w.k)$. Si le mot est inconnu (i.e. il n'est pas dans le dictionnaire), le vecteur nul est considéré.

Les vecteurs sont ensuite propagés vers le haut (propagation *ascendante*, voir figure 4.1). Considérons un sommet N dans l'arbre avec p fils N_i $(1 \leq i \leq p)$. Le nouveau vecteur calculé pour N est la somme pondérée de tous les vecteurs associés aux N_i :

$$V(N) = \alpha_1 N_1 \oplus \cdots \oplus \alpha_p N_p$$

Les poids α dépendent de la fonction syntaxique des nœuds. Par exemple, un mot gouverneur [1] se verra attribuer un poids plus important ($\alpha = 2$) qu'un mot standard ($\alpha = 1$). L'objectif de cette pondération est de permettre la différentiation de phrases formées avec des mots identiques mais ne

1. ou tête : il fait référence à la partie principale de la phrase, où de façon plus générale, du syntagme.

jouant pas le même rôle. Par exemple, les vecteurs calculés pour *le navire à vapeur* et *la vapeur du navire* ne seront pas identiques.

Lorsque le vecteur de la racine de l'arbre est déterminé, commence alors une propagation vers le bas (dite *descendante*). Le vecteur d'un sommet est faiblement contextualisé par ses parents (figure 4.2) :

$$V'(N_i) = V(N_i) \oplus \gamma(V(N_i), V(N))$$

Nous rappelons que : $\gamma(X, Y) = X \oplus (X \odot Y)$ (voir Annexe du chapitre 1).

Cette propagation est effectuée de manière récursive vers le bas, jusqu'à atteindre les feuilles de l'arbre. Au niveau des feuilles, un processus implicite de sélection lexicale est entrepris. Le nouveau vecteur *global contextualisé* est la somme pondérée des vecteurs des définitions dans lesquelles les poids sont reliés de manière non linéaire à la quantité d'information mutuelle entre le contexte (sommet N) et un sens donné :

$$\begin{aligned} V'(w) &= \beta_i V(w.1) \oplus \cdots \oplus \beta_i V(w.k) \\ \text{avec} \qquad \beta_i &= \cot(V(N), V(w.i)) \end{aligned} \tag{4.1}$$

Si le vecteur de contexte $V(N)$ est très proche de $w.i$, alors le vecteur global $V(w)$ pour le mot w est quasiment égal à $V(w.i)$ (nous rappelons que *cot* fait référence à la fonction *cotangente* avec $\cot(0) = +\infty \quad$ et $\quad \cot(\pi/2) = 0$, voir 1.4.3).

Nous pouvons définir un peu plus précisément cette fonction de *contextualisation forte* Γ. Soit un terme w ayant n acceptions $w_1 \cdots w_n$. La fonction $\Gamma(w, V)$ retourne le vecteur de w fortement contextualisé par le vecteur V :

$$\Gamma(w, V) = V(w.i) \qquad si \quad V = V(w.i) \tag{4.2}$$

$$\Gamma(w, \vec{\mathbf{0}}) = V(w) \tag{4.3}$$

$$\begin{aligned} \Gamma(w, V) &= \beta_i V(w.1) \oplus \cdots \oplus \beta_i V(w.n) \\ \text{avec} \qquad \beta_i &= \cot(V(N), V(w.i)) \end{aligned} \tag{4.4}$$

Nous étendons cette définition au cas où w n'a aucune acception (autre que lui-même) :

$$\Gamma(w_\varnothing, V) = V(w) \tag{4.5}$$

Le processus de propagation ascendant et descendant est itéré jusqu'à ce qu'un nombre maximum de cycles soit atteint, ou jusqu'à ce que le vecteur de la racine se stabilise. La stabilisation est détectée de façon empirique quand entre deux cycles, la variation de la distance angulaire entre les deux versions du vecteur racine est faible. En toute généralité la convergence du processus n'est pas garantie, et elle l'est d'autant moins pour des phrases fortement ambiguës pour lesquelles certains phénomènes d'oscillations peuvent avoir lieu. Ceci étant dit, nous observons également que ces phénomènes sont dynamiquement stables dans le temps (ils se répètent avec régularité et seraient donc automatiquement détectables, au moins en théorie mais en pratique avec un coût de calcul déraisonnable).

La désambiguïsation lexicale est effectuée implicitement par la sélection lexicale durant le processus. Cette sélection est elle-même issue de la descente par contextualisation des vecteurs des termes environnants (le contexte pour chaque terme). Une acception dont le vecteur partage de l'information avec le contexte sera favorisée au détriment des autres. Il s'agit bien d'une analyse thématique, car les fonctions syntaxiques du texte ne sont pas exploitées, mis à part l'identification

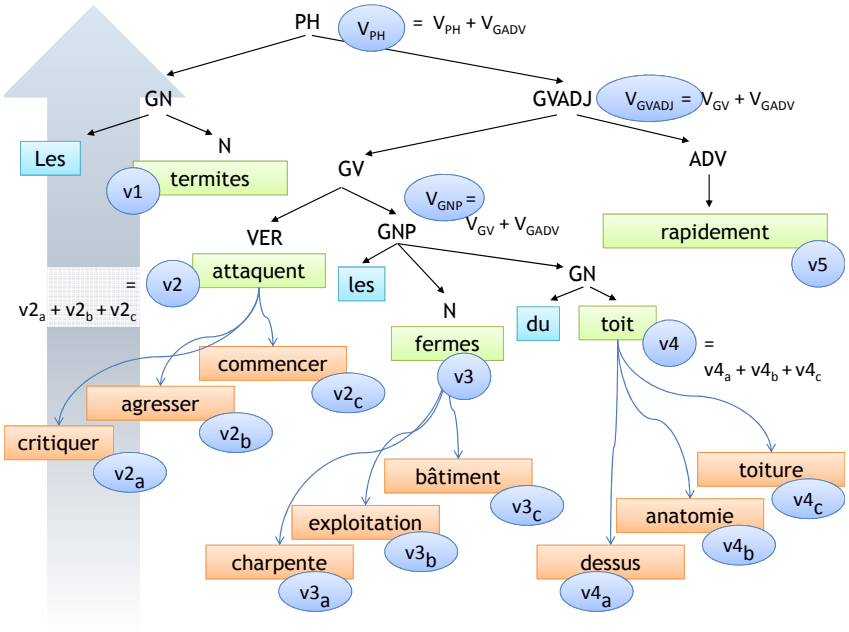


FIGURE 4.1 – Représentation graphique simplifiée de la *propagation montante* des vecteurs d'idées. Les vecteurs ascendants s'agglomèrent par somme vectorielle pondérée.

de la tête (gouverneur) pour en augmenter la pondération. La propagation (le long d'une structure, ici un arbre morpho-syntaxique) est l'idée centrale de l'approche présentée. La propagation s'accompagne d'un processus de contextualisation forte qui est à la base de la sélection (la fonction de contextualisation pour les vecteurs d'idées est présentée au chapitre 1).

Ce modèle d'analyse présente au moins deux défauts majeurs. D'une part, les diverses interprétations sont fusionnées, et d'autre part les contraintes entre les différents sens sélectionnés pour les mots ne sont pas structurellement représentées. La méthode que nous présentons dans la suite, avec les *approches bioinspirées*, tente de pallier ces inconvénients.

4.1.2 Algorithme de remontée simple

Pouvons-nous nous contenter d'une simple remontée ? L'approche par montée de vecteurs uniquement (analyse remontante) a été testée et consiste en définitive à effectuer une simple somme pondérée des vecteurs des termes présents dans le texte. La structure syntaxique induit la pondération entre les vecteurs intervenants dans la somme. L'absence de redescente ne permet pas la sélection des acceptions des termes en fonction du contexte, ce contexte étant justement représenté par les vecteurs descendants.

4.2 Extraction et calcul de termes-clés thématiques

▷ *Ces travaux relèvent partiellement du travail de Cédric Lopez [Lopez, 2009] dans le cadre de son Master 2R.*

Comment pouvons-nous construire une signature lexicale pour un texte donné ? L'indexation d'un texte peut prendre la forme d'un ensemble de mots-clés qui seraient représentatifs du texte. Nous faisons remarquer ici qu'il ne peut s'agir d'une adaptation du modèle saltonien par sélection

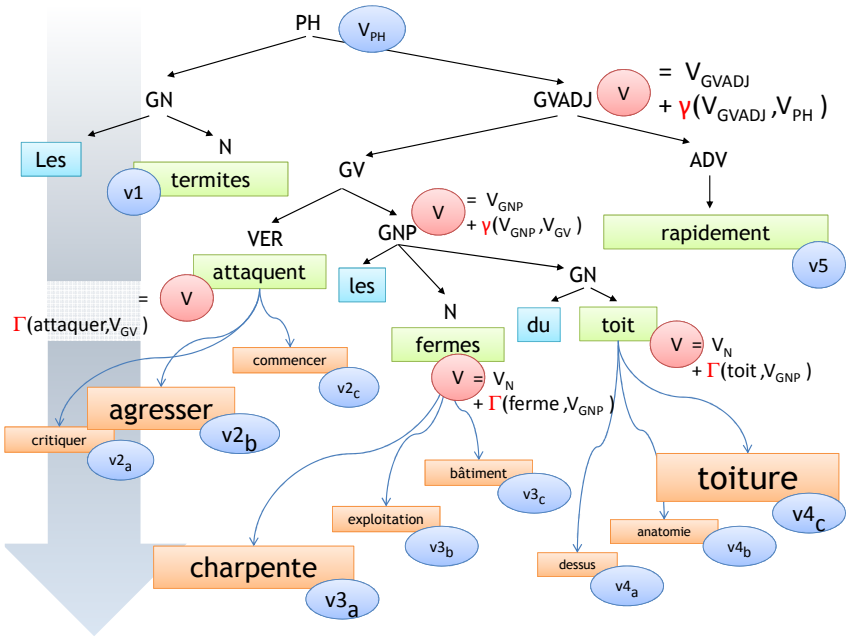


FIGURE 4.2 – Représentation graphique simplifiée de la *propagation descendante* des vecteurs d'idées. Les vecteurs descendants s'agglomèrent par contextualisation (faible γ et forte Γ). Les vecteurs des acceptions sont invariants.

des termes les plus activés. En effet, une source de connaissance externe apte à représenter les relations qu'entretiennent *normalement* les termes entre eux, semble nécessaire afin d'effectuer un *filtrage* (ne garder qu'un unique synonyme représentatif d'un ensemble de termes équivalents, par exemple) mais également une *augmentation* (c'est-à-dire, calculer des concepts pertinents implicites dans le texte).

De ce que nous avons présenté précédemment, nous pourrions envisager d'extraire ces mots à partir du vecteur calculé globalement sur le texte (ou alternativement sur chacun des vecteurs des paragraphes ou des phrases). Le passage d'un vecteur vers une liste de termes se fait par énumération du *voisinage* (voir chapitre 1). Toutefois, une telle approche manque singulièrement de précision, car en dehors de tout contexte, tous les termes fortement activés pour les vecteurs sont potentiellement candidats.

Considérons, par exemple, le syntagme suivant :

carambolage sur l'A7

Nous aimerions obtenir une liste de mots-clés pertinents traduisant les idées évoquées par le syntagme chez un lecteur. Nous espérons, par exemple, obtenir des termes comme :

carambolage, *A7*, *autoroute*, *accident de la route*, *automobile*, etc.

Dans le cas général, la liste est pondérée, et peut être arbitrairement longue (dans la limite de la taille du lexique). L'intérêt d'une approche lexicalisée est multiple. D'une part, elle offre une structure d'indexation de textes plus précise que l'approche vectorielle conceptuelle, mais, il est vrai, au prix d'une perte de rappel. Nous pourrions naïvement penser qu'une approche par énumération du voisinage du vecteur moyen du segment textuel pourrait faire l'affaire. Cependant, une telle méthode

est non seulement relativement coûteuse au niveau du calcul et sans doute de façon inutile, mais surtout semble manquer de finesse quant à la sélection des termes.

L'algorithme que nous présentons est fondé sur plusieurs principes :
- l'amorçage par extraction de mots-clés centraux ;
- la sélection itérée de mots-clés périphériques à partir des termes centraux ;
- la capture itérée de mot-clés connexes à partir des termes précédents.

4.2.1 Amorçage par mots-clés centraux

Du texte, sont extraits les mots-clés les plus saillants à partir d'une analyse saltonienne classique en TF-IDF. La fréquence inverse en documents (IDF) peut être extraite d'un corpus de référence ou d'une ressource lexicale externe (comme par exemple, le réseau JeuxDeMots). Dans un contexte global, la valeur de poids d'un terme (ou de popularité si nous prenons comme référence l'approche de Google avec l'algorithme PageRank[Page *et al.*, 1998]) peut être déterminée par la somme des poids des relations entrantes pour ce terme. Il ne s'agit ici que de termes *sémantiquement pleins* à savoir les noms, verbes, adjectifs et adverbes. Nous formulons l'hypothèse suivante :

> En général, la fréquence d'un substantif, verbe, adjectif ou adverbe T dans la langue peut être approchée dans un réseau lexical par la somme des termes incidents à T.

Cette hypothèse est un affinage de celle proposée en annexe du chapitre 3. Nous ferons remarquer que contrairement à une approche de comptage simple dans un corpus, nous obtenons une valeur de fréquence pour des termes composés (pied à coulisse, pomme de terre, etc.) mais également pour des termes désambiguïsés (*tour*>*bâtiment*, *lapin*>*viande*, etc.).

Nous avons mené une expérience informelle visant à savoir si notre hypothèse était rapidement réfutable. Nous avons effectué le comptage de termes sur une année du Monde (1994), en ne gardant que les *termes pleins*, sans majuscule (de façon à supprimer la plus grande partie des entités nommées). Nous avons, par ailleurs, exploité une liste établie par le lexicologue Étienne Brunet, rassemblant les 1 500 mots les plus fréquents de la langue française [2]. De façon similaire à la liste du Monde, nous n'avons gardé que les termes pleins.

Pour chaque terme de l'union de ces deux ensembles, nous avons effectué un calcul des poids des arcs entrants dans le réseau lexical de JeuxDeMots. Nous nous sommes posé la question de savoir s'il y avait une corrélation raisonnable entre cette mesure pour un terme dans le réseau (nous parlerons abusivement de son poids) et les données des deux ensembles ci-dessus. Nous avons obtenu :

$$\rho(\text{JDM}, \text{Le Monde}) = 0.62$$
$$\rho(\text{JDM}, \text{Brunet}) = 0.67$$
$$\rho(\text{Brunet}, \text{Le Monde}) = 0.55$$

(JDM est l'ensemble des termes et leur poids, issu de JeuxDeMots.) Que dire de ces résultats ? La corrélation n'est pas extrêmement forte, mais suffisamment élevée pour considérer qu'elle n'invalide pas d'office notre hypothèse (qui est que les données du réseau lexical peuvent être adéquates pour calculer une approximation de la fréquence des termes dans la langue). Ceci est au moins vrai pour les termes les plus fréquents, le taux de corrélation ayant tendance à baisser à mesure que les ensembles sont étendus vers des termes moins fréquents. Par ailleurs, il semblerait que le réseau lexical constitue une donnée intermédiaire (concernant les fréquences) entre un comptage sur un corpus (Le Monde) et une étude linguistique plus fine (Brunet). Il serait intéressant de refaire ce type d'étude à plus large échelle, par exemple en variant le corpus de comptage et en élargissant l'ensemble de termes considérés.

2. http://eduscol.education.fr/cid47916/liste-des-mots-classee-par-frequence-decroissante.html

À l'étape 1 de notre processus, nous avons donc entre un et trois mots-clés centraux (figures 4.3), qui correspondent (si tout s'est bien passé) aux principaux thèmes du texte. Pour s'assurer que ces termes centraux ne sont pas trop proches les uns des autres (par exemple, ils pourraient être synonymes entre eux), nous retenons d'office le premier puis nous ne sélectionnons que les suivants s'ils ne sont *pas trop similaires*.

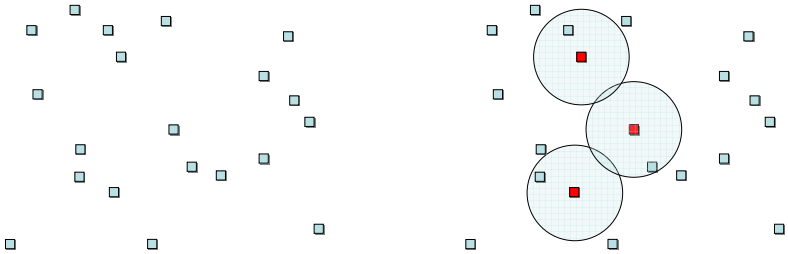


FIGURE 4.3 – Extraction de mot-clés - (a) étape 0 : l'ensemble des termes d'un texte donné et (b) étape 1 : création d'un noyau de termes clés centraux.

4.2.2 Sélection de mots-clés périphériques par diffusion dans le texte

À partir de ces mots-clés, nous itérons pour chacun d'eux une recherche, parmi les termes du texte en retenant ceux qui sont à une distance faible (au sens de la distance de vecteurs d'idées). L'itération est poursuivie tant que de nouveaux mots-clés sont sélectionnés, avec une distance plafond décroissante (figures 4.4). Le nombre de pas d'itération est fini et, de plus, connus à l'avance, dans la mesure où il dépend de la réduction du seuil.

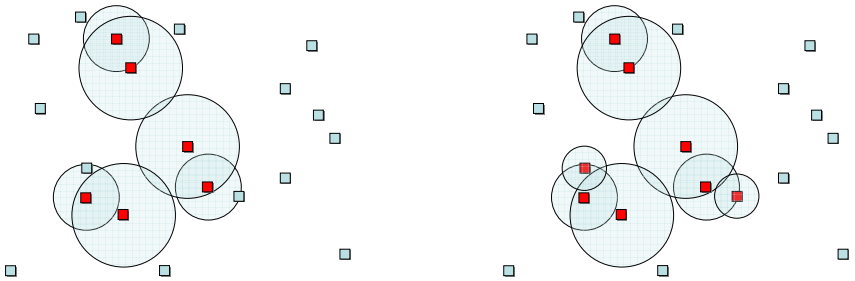


FIGURE 4.4 – Extraction de mot-clés - (a) extraction à l'itération 1 de mots-clés périphériques et (b) extraction à l'itération 2 de mots-clés périphériques. Le processus s'arrête faute de mots clés suffisamment proches.

L'idée fondamentale sous-jacente à l'approche présentée ici, est celle de la *diffusion* et de la *sélection*. La recherche de termes proches à partir d'une source correspond à l'émission d'un signal dans l'espace du texte. Ce signal implicite est émis tour à tour par les termes-clés dans le milieu (l'espace vectoriel des vecteurs d'idées, espace peuplé uniquement des termes du texte). Le signal s'épuise à chaque itération et parcours une distance de plus en plus faible, jusqu'à finalement s'arrêter. Seuls les termes atteints par ce signal sont sélectionnés et le relayent.

Nous préférons parler ici de *diffusion* (plutôt que de propagation) car nous pouvons considérer que le signal se déplace dans un milieu continu (au sens du modèle proposé). La propagation, elle, est

considérée comme le déplacement d'une information au sein d'une structure (un graphe). Les notions de voisinage de points dans l'espace et de diffusion sont ici étroitement liées, au même titre dans un autre contexte que la notion de voisinage dans un graphe et de propagation.

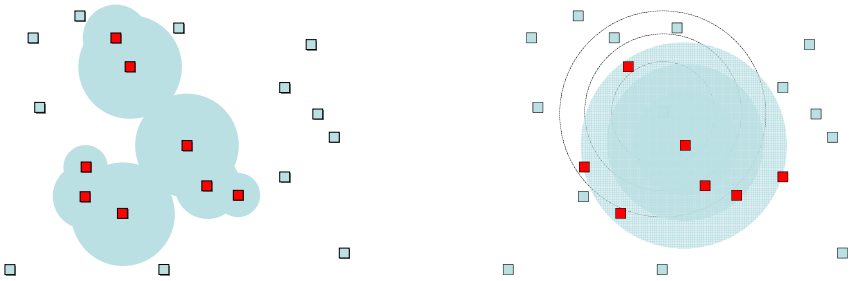


FIGURE 4.5 – (a) Extraction de mot-clés - ensemble des mots du texte constituant la signature. (b) comparaison avec sélection des mots-clés par voisinage itéré depuis le premier mot-clé ou le vecteur centroïde.

Les figures 4.5 illustrent les types de résultats obtenus avec la méthode de diffusion proposée (à gauche) ici et celle qui aurait consisté à partir d'un point central et à en sélectionner le voisinage (à droite). Le point de départ peut être le premier mot-clé (le noyau réduit à un terme) ou le vecteur centroïde du noyau (les trois premiers mots-clés). D'une façon générale, partir d'un point central ne permet que de capturer les termes relevant de la thématique dominante.

4.2.3 Capture de mot-clés connexes par propagation dans le réseau

Nous cherchons enfin à extraire du réseau lexical des mots clés connexes. Il s'agit de termes ayant les propriétés suivantes :

- ils sont à une distance angulaire faible d'au moins un des mots-clés extraits ;
- ils sont fréquents dans la langue et/ou relativement conceptuels ;
- ils ne font *a priori* pas partie du texte (cette condition n'étant pas restrictive en soi dans le processus de calcul, mais il est évident que nous ne cherchons pas à rajouter des termes déjà sélectionnés).

Pour ce faire, nous restons dans le modèle proposé de diffusion, à ceci près que l'espace constituant le milieu est celui du lexique tout entier (et non plus la restriction aux termes du texte). À chaque terme est associée une signature lexicale (un ensemble pondéré de termes) qui est une forme compilée des associations de ce terme dans le réseau lexical. Cette signature lexicale représente une approximation raisonnable du voisinage du terme dans l'espace pouvant être construite à partir du réseau. Nous effectuons la somme itérée des voisinages booléens (valeurs ramenées à 1) des k mot-clés trouvés à l'issue de l'étape précédente. Nous retenons au plus k termes connexes en éliminant ceux déjà trouvés, et ceux dont la valeur est inférieure ou égale à 1.

Par exemple, pour le segment textuel donné en exemple (*carambolage sur l'A7*), nous obtenons :

automobile :3
transport par route :2
accident de la route :2
voiture>automobile :2
autoroute :2
voiture :2

Le modèle d'extraction de mots-clés par diffusion et son extension avec la capture de mot-clés connexes issus du réseau semble donner des résultats intéressants et souvent proches de ce que produit une indexation manuelle. Dans son travail de Master 2, C. Lopez (2009) a évalué précisément

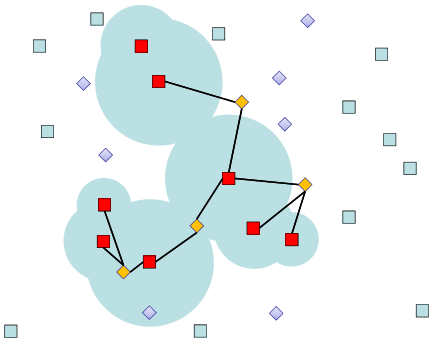


FIGURE 4.6 – Capture de mot-clés issus du réseau lexical.

les rappels et précisions pour la diffusion seule. Une F-mesure supérieure à 70 % a été trouvé pour des textes d'actualité (d'environ une page). Nos propres expériences ont montré que l'extension permettait de retrouver, dans 90 % des cas (en rappel), les mots-clés thématiques accompagnant les articles du Monde de l'année 1994.

L'approche directe par centroïde (vecteur qui est la somme pondérée de tous les termes de l'article) produit une dispersion et une redondance des mots-clés (multiplicité des synonymes). Sans extension, nous retrouvons les mots-clés thématiques dans 20 % des cas, et avec dans 55 % des cas. L'approche directe par un noyau unitaire (un seul mot-clé) concentre trop fortement la thématique supposée du document qui est devenue unique. Dans ce cas, sans extension nous retrouvons les mots-clés thématiques dans environ 30% des textes et dans 45 % avec extension.

Un échantillon très réduit de 20 articles a été fournis à une dizaine de personnes. Après lecture, il leur a été proposé plusieurs listes de mots-clés. La liste A était issue de la méthode par diffusion seule, la liste B correspondait à la diffusion plus l'extension, et la liste C était constituée de mots du texte sélectionnés aléatoirement (constituant ainsi une référence de base). Les individus devaient ordonner les listes par préférence décroissante. La liste placée en tête recevait 2 points, la seconde 1 point et la troisième (et dernière) 0 point. En moyenne pour les 30 articles, les listes A ont obtenu $1,1$ points, les listes B $1,85$ points et les listes C, $0,05$ points. L'expérience est certes extrêmement modeste mais néanmoins encourageante.

4.3 Analyse sémantique bioinspirée

Comment calculer en même temps des vecteurs d'idées, faire de la désambiguïsation lexicale et calculer des chemins interprétatifs ? La représentation des sens des mots constitue une étape majeure dans le contexte de la désambiguïsation lexicale, mais également dans le domaine de la traduction automatique (TA) pour le problème du transfert lexical en particulier, qui consiste à remplacer du signifiant dans une langue source par du signifiant dans une langue cible en fonction du contexte [Attali *et al.*, 1992]. Le modèle des vecteurs d'idées en général, et conceptuels en particulier, a pour objectif de représenter l'activation thématique des entrées lexicales, des locutions et segments de textes, jusqu'aux documents dans leur globalité. De manière imagée, nous pouvons dire que les vecteurs encodent les *idées* associées aux mots ou aux expressions. Les principales applications de ce modèle sont l'analyse thématique de texte et la désambiguïsation lexicale [Lafourcade, 2001a]. En pratique, nous avons présenté un système, possédant des capacités d'apprentissage automatique, utilisant les vecteurs conceptuels et exploitant des dictionnaires unilingues français (disponibles sur Internet) et des dictionnaires de synonymes parmi différentes sources. Jusqu'à présent, sur la seule base du français, le système comprend 200 000 entrées lexicales qui correspondent à envi-

ron un demi-million de vecteurs (le nombre moyen de sens pour les mots polysémiques est d'environ 5). La même étude est actuellement conduite pour l'anglais, mais n'est à ce jour pas encore opérationnelle. Le processus d'analyse et de désambiguïsation lui-même constitue le cœur de l'étude dont les grandes lignes sont présentées ici.

La compréhension d'un texte requiert la comparaison des différents sens des mots polysémiques dans le contexte de leur utilisation. La difficulté provient de la caractérisation du contexte qui est lui-même défini par les mots avec l'ensemble de leurs sens et leur catégorie syntaxique dans le texte (verbe, nom, adjectif, etc.). À partir des catégories, des relations grammaticales entre les mots sont extraites et ces relations peuvent être représentées sous la forme d'une structure arborescente appelée *arbre d'analyse morphosyntaxique*. Cependant les relations *sémantiques* entre les mots ne sont en aucune façon présentes dans cet arbre. Pour représenter ces relations et la structure afférente, nous considérons les mots comme les entités de base d'un réseau d'interactions dans lequel la dynamique implicite permet de révéler le sens le plus probable parmi l'ensemble des sens attachés aux mots polysémiques.

La thèse que nous défendons ici est qu'un texte peut être considéré comme un *système complexe*. Nous en proposons la définition générale suivante et comme à notre connaissance il ne semble pas en exister aujourd'hui d'unique ou consensuelle dans le domaine du TAL (contrairement en physique, en chimie ou encore en automatique) nous tentons une définition pour l'analyse sémantique de textes.

> *Un système complexe est un système composé de nombreuses entités au comportement dynamique en interaction entre elles et avec leur environnement. Le comportement global du système, non déductible des caractéristiques des entités elles-mêmes, émerge de l'auto-organisation du système.*
>
> *L'analyse sémantique de textes peut-être vue comme un système complexe dont les acteurs sont les objets du texte (mots, acceptions, syntagmes, etc.) et les relations qu'ils entretiennent (rôles syntaxiques, rôles sémantiques, etc.), et dont la dynamique vise à élaborer des chemins d'interprétation entre acteurs et la mise en évidence des plus adaptés à l'environnement.*

La nature offre un éventail varié de systèmes répondant à la définition générale. Ils sont caractérisés par des propriétés globales qui ne sont obtenues ni par un processus de supervision, ni par une coordination centralisée. Bancs de poissons, nuées d'oiseaux [Chaté & Grégoire, 2004], colonies de bactéries [BJ2004], tas de sable [Bak, 1996], réseaux d'interactions protéiques [Amar *et al.*, 2004] et sans aucun doute les langages sont autant d'exemples de systèmes de ce type. Leurs propriétés résultent des interactions locales entre les entités elles-mêmes et entre les entités et leur environnement. Leur capacité d'auto-organisation, qui peut être définie comme un processus dynamique, holistique et décentralisé de structuration, permet leur auto-adaptation aux changements dynamiques et imprévisibles qui se produisent au sein de leur environnement.

L'action d'une entité peut affecter les actions ultérieures d'autres entités dans le système ; cette interdépendance s'exprime au niveau global par la formule bien connue : *le tout est plus que la somme de ses parties*, ou, autrement formulée, *l'action du tout produit davantage que la simple somme des actions de ses parties*[3]. Les actions, dans le contexte de notre étude, correspondent aux sens des mots qui constituent le texte, et la somme des actions produit le sens global du texte, qui est, à n'en pas douter, bien plus que la simple somme des sens des mots considérés. En pratique, l'un des problèmes qui se présentent est que les sens ne sont pas, à proprement parler, des éléments actifs. Ainsi, pour exprimer la dynamique du système dans son ensemble, nous avons décidé d'ajouter au système un support d'activité composé de *transporteurs de sens*. Ces *transporteurs* ont pour but de permettre les interactions entre les éléments qui composent le texte. Ils doivent être à la fois légers (du fait de leur nombre potentiellement important) et indépendants (les sens des mots sont

3. [Langton, 1996] *Why do we need artificial life ?* page 305.

des valeurs intrinsèques aux mots eux-mêmes). De plus, lorsque certains sens de mots différents sont compatibles (*employé* avec *travail* par exemple), le système doit en conserver une trace. L'ajout dans le graphe d'une arête liant les deux sens compatibles, indépendamment de leur position dans le texte, est la solution que nous avons retenue pour marquer cette compatibilité.

La prise en compte de l'ensemble de ces éléments et de ces contraintes nous a conduit à choisir les fourmis artificielles pour jouer le rôle des transporteurs de sens. La motivation première du choix des fourmis tient à leur capacité à exprimer leurs interactions, qui sont rappelons-le à la fois locales et indirectes, par des marques numériques, les phéromones, et par des marques structurelles matérialisées par la modification de leur environnement. Dans le cadre de notre problème de désambiguïsation lexicale, l'environnement est un arbre d'analyse morphosyntaxique.

Nos travaux sur la question ont pour objectif d'exploiter la capacité d'auto-organisation des colonies de fourmis pour la détermination de solutions au problème de la désambiguïsation lexicale, pour lequel la définition d'une fonction d'évaluation globale appropriée est extrêmement délicate. Plus précisement, la désambiguïsation lexicale est la détermination du sens des mots, en contexte (d'un énoncé, d'un discours ou d'un dialogue). L'idée retenue consiste à concevoir un système à base de fourmis artificielles qui se déplacent à l'intérieur d'un graphe, résultant de l'analyse syntaxique du texte étudié, et qui le modifient. Une solution s'exprime comme un ensemble de chemins qui mettent en évidence les compatibilités entre les sens des mots. Pour construire de tels chemins, les fourmis disposent de deux types d'objets. Les marques numériques, les phéromones, sont déposées sur les arêtes et indiquent l'importance relative de certains chemins. D'autres arêtes, que nous appellerons les *ponts*, sont créées par les fourmis entre des sommets voisins sur le plan sémantique. Ces arêtes constituent des marques structurelles. La combinaison des phéromones et des ponts permet aux fourmis de coopérer de manière indirecte et asynchrone. Ce principe de base a été identifié par Pierre-Paul Grassé qui l'a nommé *stigmergie* [Grassé, 1959].

Bien que les algorithmes à fourmis aient été largement utilisés pour le traitement de problèmes d'optimisation classiques, nous n'avons pas connaissance de leur utilisation dans le domaine du traitement algorithmique de la langue naturelle (TAL). Cependant, une idée proche a déjà été mise en œuvre par le projet COPYCAT [Hofstadter, 1995]. Dans ce travail, l'environnement lui-même contribue au calcul de la solution et cet environnement est modifié par une population d'agents dont le rôle et la motivation varient (voir également les travaux de Mitchell [Mitchell, 1993]). En 1992, Gale, Church et Yarowsky ont utilisé une technique relevant de l'approche bayésienne naïve pour la désambiguïsation lexicale [Gale *et al.*, 1992]. Certaines propriétés de ces modèles semblent bien adaptées à des tâches d'analyse sémantique et lexicale, dans la mesure où les sens des mots peuvent être considérés en concurrence à l'accès aux ressources.

Au delà des concepts relevant de la *bioinspiration* (stigmergie, mise en concurrence pour l'obtention de ressources, adaptation à un environnement changeant), quelques principes nous paraissent essentiels pour approcher la résolution du problème : (1) l'information mutuelle où la proximité sémantique est un facteur déterminant pour l'activation lexicale, (2) la structure syntaxique du texte et en particulier les fonctions syntaxiques peut servir de guide pour la propagation de l'information, (3) des *ponts conceptuels* entre éléments de l'environnement (les termes et les syntagmes) peuvent être construits (et supprimés) dynamiquement et fournissent une lecture partielle du résultat. Ces ponts sont des éléments qui permettent l'échange d'information mutuelle au-delà des voisinages locaux et constituent des raccourcis pouvant s'agglomérer de proche en proche. Finalement, comme noté par Hofstadter [Hofstadter, 1995], (4) la randomisation biaisée (à ne pas confondre avec le chaos) en particulier à travers l'utilisation de la fonction sigmoïde, joue un rôle majeur dans le modèle, sachant que l'effet global n'est lui pas du tout aléatoire.

Les détails concernant le modèle d'analyse de texte bioinspiré, ainsi que certains résultats sont présents dans l'article inclus (M. Lafourcade, F. Guinand (2010) *Artificial Ants for Natural Language Processing*).

Conclusion du chapitre 4

Globalement, l'exploitation de l'information mutuelle fondée sur des structures thématiques pouvant prendre la forme de vecteurs d'idées, permet d'atteindre environ 75 % de réussite en désambiguïsation lexicale. Avec les mêmes moyens, le rattachement de groupes prépositionnels peut être effectué avec succès dans des ordres de grandeur similaires. Les approches bioinspirées, avec une agentification forte, semble constituer une approche prometteuse quant à la mise en œuvre des processus de calcul pour la réalisation de l'analyse de textes. Toutefois, nous désirons explorer plus avant la possibilité d'approches holistiques où la stratification en phases (morphologique, syntaxique puis sémantique) laisserait la place à des approches où les tâches seraient plus finement entrelacées. De plus, bien que nous ne l'ayons pas explicité tel quel, l'apprentissage par le calcul de vecteurs d'idées lors de l'analyse de textes semble aussi être un élément important devant être intimement lié au processus d'analyse lui-même. L'analyse sémantique doit de plus tirer davantage profit de la notion de chemins interprétatifs qui peuvent fournir des structures profondes d'explications. Un couplage de l'analyse de textes avec les réseaux lexicaux semble alors une direction à prendre. Un réseau lexical ne serait plus seulement un fournisseur de structures sémantiques (vecteurs d'idées, signatures lexicales) mais serait en lui-même l'environnement où s'activeraient termes et concepts lors de l'analyse.

Articles adjoints au chapitre 4

M. Lafourcade et Ch. Boitet (2002) *UNL lexical Selection with Conceptual Vectors.* In proc. of LREC'2002, Las Palmas, Canary Island, Spain, May 27, 2002, 7 p.

M. Bouklit et M. Lafourcade (2006) *Propagation de signatures lexicales dans le graphe du Web.* In proc of RFIA'2006, Tours, France, 25 au 27 janvier 2006, 9 p.

M. Lafourcade, F. Guinand (2010) *Artificial Ants for Natural Language Processing* in Artificial Ants. N. Monmarché, F. Guinand, P. Siarry Eds. Wiley ISBN : 9781848211940. pages 454-492.

Annexe : à propos de la fonction sigmoïde

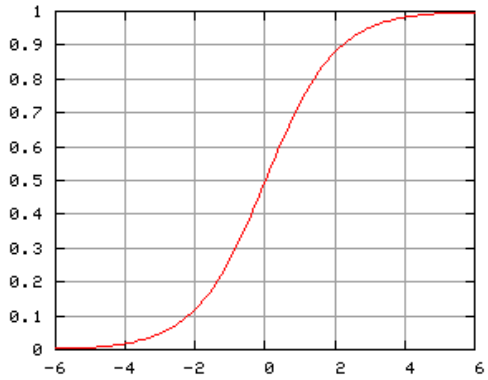


FIGURE 4.7 – Fonction sigmoïde, cas particulier de fonction logistique (source Wikipédia).

La fonction sigmoïde est de la forme :

$$f(x) = \frac{1}{1+e^{-x}} = \frac{1}{2} + \frac{1}{2}\tanh(\frac{x}{2})$$

et son expression la plus générale, comme famille de fonctions logistiques, est :

$$f(x) = K\frac{1}{1+ae^{-rx}} \qquad \text{K, r} \in \mathbb{R}^+ \quad \text{et} \quad a \in \mathbb{R}$$

Cette fonction est souvent utilisée dans les réseaux de neurones comme fonction d'activation. La production d'un agent (une fourmi) dans le modèle bioinspiré, correspond en effet à l'émission d'un signal destiné à se propager dans l'environnement. La probabilité d'émission du signal est ainsi simulée par la fonction sigmoïde. Le coût de l'émission du signal fait baisser l'activation (en abscisse dans la figure 4.7) d'une petite quantité, rendant l'émission d'un nouveau signal moins probable. À activation nulle, la probabilité d'émission est de $1/2$, ce qui correspond au milieu de la phase de transition d'un état non-actif vers un état actif.

La fonction sigmoïde est un cas particulier de fonction logistique (de P. F. Verhulst) modélisant une évolution de population comportant un frein et une certaine capacité d'accueil K (on peut passer de l'une à l'autre par transformation affine). Ces fonctions peuvent également modéliser des réactions autocatalyliques. Les fonctions en S (logistiques, tangente hyperbolique ou fonction de Heaviside, entre autres) apparaissent aussi fréquemment dans les méthodes bayésiennes.

UNL Lexical Selection with Conceptual Vectors

Mathieu LAFOURCADE*, Christian BOITET**

*LIRMM
161, rue Ada
F-34392 Montpellier cedex 5, France
Mathieu.Lafourcade@lirmm.fr

**GETA, CLIPS, IMAG
385, av. de la bibliothèque, BP 53
F-38041 Grenoble cedex 9, France
Christian.Boitet@imag.fr

Abstract

When deconverting a UNL graph into some natural language LG, we often encounter lexical items (called UWs) made of an English headword and formalized semantic restrictions, such as "look for (icl>do, agt>person)", which are not yet connected to lemmas, so that is it necessary to find a "nearest" UW in the UNL-LG dictionary, such as "look for (icl>action, agt>human, obj>thing)". Then, this UW may be connected to several lemmas of LG. In order to solve these problems of incompleteness and polysemy, we are applying a method based on the computation of "conceptual vectors", previously used successfully in the context of thematic indexing of French and English documents.

Keywords: disambiguation, deconversion, UNL-French localization, transfer, conceptual vectors, lexical selection

Introduction

The UNL project of network-oriented multilingual communication has proposed a standard for encoding the meaning of natural language utterances as semantic hypergraphs intended to be used as pivots in multilingual information and communication systems. In the first phase (1997-1999), more than 16 partners representing 14 languages have worked to build deconverters transforming an (interlingual) UNL hypergraph into a natural language utterance.

The UNL-French deconverter first performs a "localization" operation within the UNL format, and then classical transfer and generation steps (Boitet & al., 1982; Boitet, 1997; Slocum, 1984). This raises interesting issues about the status of the UNL language, designed as an interlingua, but diversely used as a linguistic pivot (disambiguated abstract English), or as a purely semantic pivot.

When deconverting a UNL graph into some natural language LG, we often encounter lexical items (called UWs) made of an English headword and formalized semantic restrictions, such as `"look for (icl>do, agt>person)"`, which are not yet connected to lemmas, so that is it necessary to find a "nearest" UW in the UNL-LG dictionary, such as `"look for (icl>action, agt>human, obj>thing)"`. Then, this UW may be connected to several lemmas of LG. In order to solve these problems of incompleteness and polysemy, we apply a method based on the computation of "conceptual vectors", previously used successfully in the context of thematic indexing of French and English documents.

We first present our general technique of disambiguation using conceptual vectors (DCV), then the context of disambiguation in a deconversion from UNL into a natural language, and the application of DCV to this problem.

1. Conceptual Vectors

1.1 Outline of the method

In short, our method is as follows. First, we prepare a very large dictionary of wordsenses with associated conceptual vectors. We begin by associating very "crude" conceptual vectors manually to a small set of terms, our "kernel". The dimensions are the 873 leaves of Roget's thesaurus for English, adapted to French. We can also "unfold" some of these dimensions into more detailed specific thesaurii.

We then use a large coverage French analyzer to transform all definitions of all the terms known by the analyzer into annotated tree structures. Then, we attach the crude conceptual vectors to the kernel terms, and empty conceptual vectors to all other words and all non lexical nodes, and perform simulated annealing on the whole tree. The conceptual vector of the root becomes the conceptual vector for the word sense in question, while the conceptual vectors of non kernel terms become new initial vectors for them. This way, the kernel grows.

In December 2001, we had 64,000 terms, an average of 3.3 word senses (definitions) per term, and 210,000 conceptual vectors.

We use several distances between conceptual vectors, among them the classical Arg_cosine, which has a natural interpretation in terms of "angular distance" and models well the notion of "distance from a point of view". This particular distance is used to classify the conceptual vectors of each term into a binary decision tree. The leaves contain the conceptual vectors of the individual definitions and the internal nodes a weighted average of the conceptual vectors of their daughters. This is useful because we use all kinds of dictionaries, with the result that two definitions may be different but very close.

This "learning process" is iterated constantly over the growing set of terms.

To disambiguate a particular occurrence of a term in a document, we first analyze the whole document into a possibly large decorated tree (several dozen pages are routinely processed as one tree). We then attach to the lexical nodes their average conceptual vectors, and perform simulated annealing on the document tree. The conceptual vectors near the top of the tree give a thematic characterization of the corresponding parts of the document (section, paragraph…).

The conceptual vector of each lexical node has also changed into a "contextually recooked" vector. It is now possible to find the closest conceptual vector in its binary decision tree. This "contextual CV-based disambiguation process" produces either a set of possible senses (the leaves of that subtree), or one sense (the closest among them).

1.2 Mathematical basis

1.2.1 Conceptual vector space

The conceptual vector model is based on the projection on a mathematical model of the linguistic notion of semantic fields. The question of how to choose (or build) a concept set is far beyond the scope of this model and is left to people studying ontologies. In our prototype applied to French and English, we have chosen (Larousse 1992) where 873 concepts are identified.

The main hypothesis is that this set constitutes a generator space for the words (terms in general) and their meanings and as such, any word would project its meanings on this space.

Let C be a finite set of n concepts. A conceptual vector V is a linear combination of elements of C. For a meaning A, vector V_A is the description (in extension) of activations of all concepts of C. For example, the different meanings of *to tidy up* and of *to cut* could respectively be projected on concepts of C as follows (for clarity sake, CONCEPT [intensity] are ordered by decreasing intensity values).

V(to tidy up) = CHANGE [0.84], VARIATION [0.83], EVOLUTION [0.82], ORDER [0.77], SITUATION [0.76], STRUCTURE [0.76], RANK [0.76] …

V(*to cut*) = GAME [0.8], LIQUID [0.8], CROSS [0.79], PART [0.78] MIXTURE [0.78], FRACTION [0.75], TORTURE [0.75] WOUND [0.75], DRINK [0.74] …

Lexical items associated with their vectors are stored in conceptual lexicons. Each meaning of a polysemous word is associated to a different vector. The global vector of a term is (with some simplification) a normalized vector sum of all its meanings. For instance:

V(head) = HEAD [0.83], . BEGINNING [0.75], ANTERIORITY [0.74], PERSON [0.74] INTELLIGENCE [0.68], HIERARCHY [0.65], …

The following metaphor may help apprehending why the angular distance can be used as an artifact for thematic proximity. Let us see the space of all word senses as a sky full of stars. The empty space between two stars may be pointed to although there is no star (word sense) between them. Stars form constellations, some parts of the space being crowded, others being underpopulated. Then, a meaning is a direction in the space, but not an actual point, as, from the observer point of view, the Euclidean distance between the observer and the point cannot be assessed. The angle between two directions defines their distance.

We don't consider the vector norm for the following reason. Take a vector representing the idea of the red color. Take another vector collinear but with twice its norm. Does the second vector represent the idea of something redder? If yes, then the first one is less red, which means that it might be more blue (or yellow or green or darker or lighter, etc.). But, in this case, it should not point to the same direction, which is not what we supposed at first. The vector norm may be used as a measure of the intensity of expression of the idea (like from screaming to whispering) but not directly as an estimator of thematic activation and closeness.

1.2.2 Distance and test functions

We define Sim(X, Y) as one of the similarity measures often used in information retrieval (Morin 99). Using this measure, we can express the angular distance D_A between two vectors X and Y by:

$$D_A(X,Y) = \mathrm{Arc\,cos}(\mathrm{Sim}(X,Y)) = \mathrm{Arc\,cos}\left(\frac{X \cdot Y}{\|X\| \times \|Y\|}\right)$$

Intuitively, this function constitutes an evaluation of the thematic proximity. Mathematically, it is the measure of the (hyper)angle between the two vectors. We consider, that, if $D_A(X, Y) \leq \pi/4$, X and Y are thematically close and share many concepts. For $D_A(X, Y) \geq \pi/4$, the semantic proximity between X and Y is considered as loose. Around $\pi/2$, meanings are almost without any relation. At $\pi/2$, they have strictly no relationship (which never happens in practice).

This is a real distance function (contrary to the similarity measure) as it verifies the properties of reflexivity, symmetry and triangular inequality.

$$\begin{cases} D_A(X, X) = 0 \\ D_A(X, Y) = D_A(Y, X) \\ D_A(X, Y) + D_A(Y, Z) \geq D_A(X, Z) \end{cases}$$

We have by definition $D_A(0, 0) = 0$ and $D_A(X, 0) = \pi/2$ with 0 as the null vector. The null vector has no associated word in any language,

as it represents the "empty idea", which does not activate any concept.

Let X be a lexical property. We define the test function $P_x(V)$ of V against X as:

$$P_p(X) = \frac{\pi}{2} - D_A(V(p), X)$$

We use test functions to give a score to lexical items in inverse proportion of their distance to (the set of words meeting some) lexical constraints. In the context of UNL, these properties will be the UNL restrictions as expressed in the UWs.

1.2.3 Useful vector operations

The normalized vector sum of X and Y is the vector V defined by:

$$V = X \oplus Y \mid v_i = \frac{x_i + y_i}{\|V\|}$$

The sum can be generalized to any number of vectors:

$$V = \sum_i X_i \mid v_i = \frac{\sum x_i}{\|V\|}$$

The term to term vector product of X and Y is the vector V defined by:

$$V = X \otimes Y \mid v_i = \sqrt{x_i y_i}$$

We can interpret the sum as the mean (or barycenter) of the vectors. The normalized term to term product can be seen as a kind of intersection between vector components. Note that the norm of the resulting vector of the product is lower or equal to 1.

1.3 Lexical contextualization

Outside of any context, when a word *w* has *n* meanings, it is associated to *n* vectors Vi and the global vector of *w* is the barycenter of all Vi (with weights all set to 1). The construction of a contextualized vector V is done by modifying these weights according to the context. It is then a vector sum where weights are Pp(X) values :

$$Vp(w) = \sum_{i=0}^{n} P_p(V_i(w)).V_i(w)$$

For instance, the vector associated to the (highly) polysemic word *head* in the context of P_{body} refers properly to the body part.

$V_{body}(head)$ = HEAD [0.97], . PERSON [0.85] INTELLIGENCE [0.78], BODY [0.75], …

2. UNL-French deconversion

2.1 The UNL project and language

2.1.1 The project

The pivot paradigm is used: the representation of an utterance in the UNL interlingua (UNL stands for "Universal Networking Language") is a hypergraph where normal nodes bear UWs ("Universal Words", or interlingual acceptions) with semantic attributes, and arcs bear semantic relations (deep cases, such as agt, obj, goal, etc.). Hypernodes group a subgraph defined by a set of connected arcs. A UW denotes a set of interlingual acceptions (word senses), although we often loosely speak of "the" word sense denoted by a UW.

Because English is known by all UNL developers, the syntax of a normal UW is: "<English word or compound> (<list of restrictions>)", e.g. "look for (icl>action, agt>human, obj>thing)".

Going from a text to the corresponding "UNL text" or interactively constructing a UNL text is called "enconversion", while producing a text from a sequence of UNL graphs is called "deconversion".

This departure from the standard terms of analysis and generation is used to stress that this is not a classical MT project, but that UNL is planned to be the source format preferred for representing textual information in the envisaged multilingual network environment. The schedule of the project, beginning with deconversion rather than enconversion, also reflects that difference.

Each group is free to use its own software tools and/or lingware resources, or to develop directly with tools provided by the UNL Center (UNU/IAS & UNDL).

Emphasis is on a very large lexical coverage, so that all groups spend most of their time on the UNL-NL lexicons, and develop tools and methods for efficient lexical development. By contrast, grammars have been initially limited to those necessary for deconversion, and are gradually expanded to allow for more naturalness in formulating text to be enconverted.

2.1.2 The UNL components

2.1.2.1 Universal Words

The nodes of a UNL utterance are called Universal Words (or UWs). The syntax of a normal UW consists of a headword and a list of restrictions.

Because English is known by all UNL developers, the headword is an English word or compound. The restrictions are given as an

attribute value pair where attributes are semantic relation labels (those used in the graphs and some more thesaurus-oriented) and values are other UWs (restricted or not).

A UW denotes a collection of interlingual acceptions (word senses), although we often loosely speak of "the" word sense denoted by an UW. For example, the unrestricted UW "look for" denotes all the word-senses associated to the English compound word "look for". The restricted UW "look for(icl>action, agt>human, obj>thing)" represents all the word senses of the English word "look for" that are an action, performed by a human that affects a thing. In this case this leads to the word sense: "look for – to try to find".

2.1.2.2 UNL hypergraphs

A UNL expression is a hypergraph with a unique entry node (a connected graph may be labelled and given an entry node, thereby becoming a subgraph or "scope"). The arcs bear semantic relation labels (deep cases, such as agt, obj, goal, etc.).

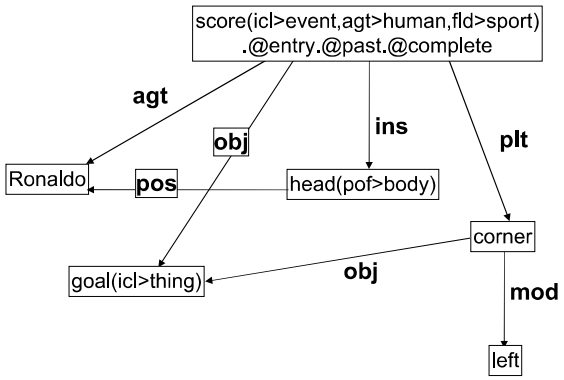


Fig. 1: a possible UNL graph for "Ronaldo has headed the ball into the left corner of the goal"

In a UNL graph, UWs appear with attributes describing what is said from the speaker's point of view. This includes phenomena like speech acts, truth values, time, etc.

These hypergraphs are written as text using the UNL "language": a graph is written as a list of arcs, connecting the different nodes. For example, the graph presented in Fig. 1 can be written as:

```
agt(score(…).@entry.@past.@complete,
    Ronaldo)
obj(score(…).@entry.@past.@complete,
    goal(icl>thing))
pof(head(pof>body),
    Ronaldo)
ins(score(…).@entry.@past.@complete,
    head(pof>body))
plt(score(…).@entry.@past.@complete,
    corner)
obj(corner, goal(icl>thing))
mod(corner, left)
```

Fig. 2: Linear writing of the same UNL graph

2.2 The place of disambiguation in the French deconverter

2.2.1 Overview

The global deconversion process for French (without conceptual vectors) has been presented in (Sérasset & Boitet, 1999). Deconverting consists in transforming a UNL graph into one or more utterances in a natural language. To this purpose, we segment the process into 7 phases, as illustrated below. The third phase (graph-to-tree) produces a decorated tree which is fed into an Ariane-G5 TS (structural transfer).

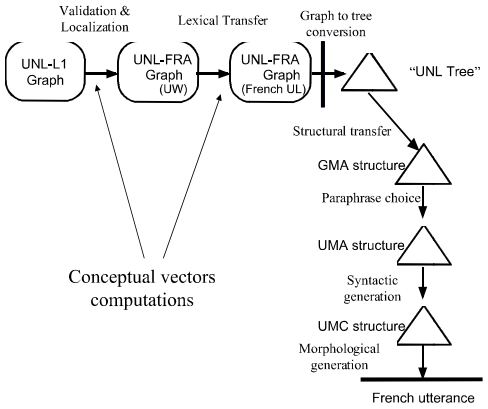


Fig. 3: architecture of the French deconverter with lexical selection enhanced using conceptual vectors

2.2.2 Transfer

2.2.2.1 Validation

When a UNL graph is to be deconverted, its correctness is first checked: it has to be connected, and the features on the nodes must be in a predefined list. Validation is necessary to improve the robustness of the deconverter, as there is no hypothesis on the way a graph is created. Invalid graphs are rejected.

2.2.2.2 Lexical & cultural localization

Some lexical units used in the graph may not be present in the French deconversion dictionary. This problem may appear under different circumstances. First, the French dictionary (constantly under development) may be incomplete. Second, the UW may use an unknown notation to represent a known French word sense. Third, the UW may represent a word sense absent from the French language.

We solve these problems with the following method : Let w be a UW in the graph G. Let D be the French dictionary (a set of UWs). We substitute w in G by w' such that : $w' \in D$ and $\forall x \in D\ d(w, w', G) = d(w, x, G)$. where d is a pseudo-distance function.

Without vectors, if different French UWs are at the same pseudo-distance of w, w' is chosen at

random among these UWs (default in non-interactive mode).

On the "cultural" aspect, some crucial information may be missing, depending on the language of the source utterance (sex, modality, number, determination, politeness, kinship, etc).

It is in general impossible to solve this problem fully automatically in a perfect manner, as we do not know anything about the document, its context, and its intended usage: FAHQDC[1] is no more possible than FAHQMT on arbitrary texts. We have to rely on necessarily imperfect heuristics. The heuristics we have chosen use conceptual vector contextualization.

2.2.2.3 Lexical transfer

After the localization phase, we perform a lexical transfer. It would seem natural to convert the graph into a tree and then to do it in Ariane-G5. But lexical transfer is context-sensitive, and we want to avoid the possibility of transferring differently two tree nodes corresponding to one and the same graph node.

Each graph node is replaced by a French lexical unit (LU), along with some variables. A lexical unit used in the French dictionary denotes a derivational family (e.g. in English: destroy denotes destroy, destruction, destructible, destructive…, in French: détruire covers détruire, destruction, destructible, indestructible, destructif, destructeur). This is where conceptual vectors can help to select the most probable meaning according to the (vector) context.

There may be several possible lexical units for one UW. This happens when there is a real synonymy or when different terms are used in different domains to denote the same word sense[2]. In that case, without conceptual vectors the choice of the lexical unit is done at random or interactively as there is no information about the task the deconverter is used for.

The same problem also appears because of the strategy used to build the French dictionary. In order to obtain a good coverage from the beginning, we have underspecified the UWs and linked them to different lexical units. This way, we considered a UW as the denotation of a set of word senses in French.

Hence, we can reuse previous dictionaries, and use the dictionary even if it is still under development and incomplete. In our first version, we also solve this problem by a random selection of a lexical unit.

2.2.2.4 Graph to tree conversion

The subsequent deconversion phases are performed in Ariane-G5. Hence, it is necessary to convert the UNL hypergraph into an Ariane-G5 decorated tree.

The UNL graph is directed. Each arc is labelled by a semantic relation (agt, obj, ben, con…) and each node is decorated by a UW and a set of features. One node is distinguished as the "entry" of the graph.

An Ariane-G5 tree is a general (non binary) tree with decorations on its nodes. Each decoration is a set of variable-value pairs.

The graph-to-tree conversion algorithm has to maintain the direction and labelling of the graph along with the decorations of the nodes.

Our algorithm splits the nodes that are the target of more than one arc, and reverses the direction of as few arcs as possible. An example of such a conversion is shown in figure 2.3.

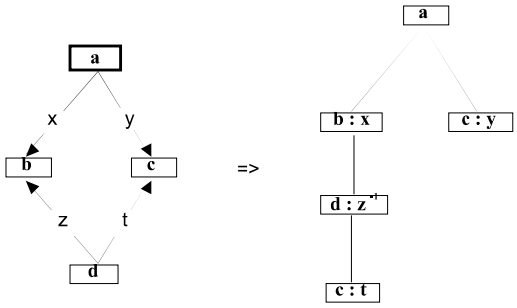


Fig. 4: example graph to tree conversion

The graph to tree conversion algorithm has been extensively described in (Sérasset & Boitet, 2000).

2.2.2.5 Structural transfer

The purpose of the structural transfer is to transform the tree obtained so far into a Generating Multilevel Abstract (GMA) structure (Boitet, 1994). In this structure, non-interlingual linguistic levels (syntactic functions, syntagmatic categories…) are underspecified, and (if present), are used only as a set of hints for the generation stage.

3. Integration of DCV in the deconversion process

3.1 DCV in localization

The vector contextualization generalizes both kinds of localization (lexical and cultural). The selected UW is the one which vector is the closest to the contextualized vector.

All restrictions of the UWs in the UNL graph are taken as information for the lexical contextualization. For example, the vector attached to the word *head* will be contextualized with *body* because of the presence of the restriction `pof>body`. Although different weights could be associated to the kind of restriction (as `icl`, `agt`, `fld`, …) we simply consider these restrictions equally, because we

[1] fully automatic high quality deconversion.

[2] strictly speaking, the same collection of interlingual word senses (acceptions).

cannot know in advance which are the most relevant.

Some restrictions are not valued (like `plt`) and are converted (from a correspondence table) into appropriate contexts (like *place* for `plt`).

3.2 Computing conceptual vectors on the tree

When the graph has been converted into a tree, a full conceptual vector analysis can be undertaken. Vectors are attached to the tree and then propagated up to the root. Usually, we back propagate toward the leaves, to induce a vector mutual activation. We usually perform cycles until the root vector converges. As convergence is not guaranteed in general, we set a maximum number of cycles to force the process to end.

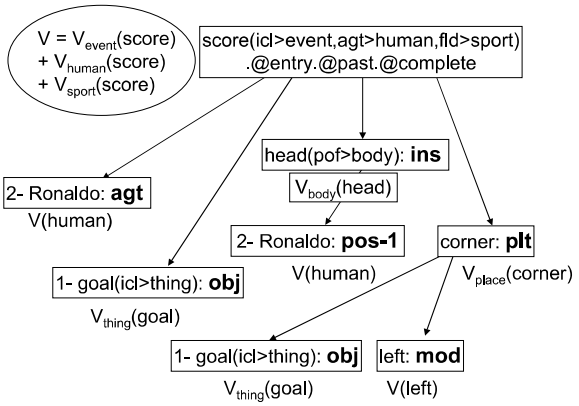


Fig. 5: Vector attachment to a UNL tree

In the upward process, the vector of a node N with k daughters $n_1 \ldots n_k$ is the sum of its original vector plus the mean (normalized sum) of all daughter vectors:

$$\uparrow V'(N) = V(N) \oplus \sum_{i=0}^{n} V(n_i)$$

For example, the vector of the node *corner* will be itself plus the sum of V(goal) (contextualized as an object) and V(left).

In the downward process, the vector is weakly contextualized by its father, by the application of the term to term product.

$$\downarrow V'(n_i) = V(n_i) \oplus V(N) \otimes V(n_i)$$

The global effect of this process is to make each vector "resonate" with both other vectors and the restriction, to produce a vector that is as close as possible to the different possible meanings and to the context.

3.3 DCV in lexical transfer

The transfer is a generalization of the lexical selection to another conceptual lexicon. From a (contextualized) vector V of a conceptual lexicon A, we select the item of the closest vector in the conceptual lexicon B. This strategy is used each time several lexical unit compete for one UW. We should note here that the same concept set has been used for the construction of the conceptual lexicon of English and French. This approach makes the comparison of vector feasible, although it may be questioned from an ontological point of view between two different languages.

Conclusion

When deconverting a UNL graph into a natural language LG, we often encounter lexical items (called UWs) made of an English headword and formalized semantic restrictions, which are not yet connected to lemmas, so that is it necessary to find a "nearest" UW in the UNL-LG dictionary. Then, this UW may be connected to several lemmas of LG.

In order to solve these problems of incompleteness and polysemy, we apply a method based on the computation of "conceptual vectors", previously used successfully in the context of thematic indexing of French and English documents.

References

Blanc É. & Guillaume P. (1997) *Developing MT lingware through Internet : ARIANE and the CASH interface*. Proc. of Pacific Association for Computational Linguistics 1997 Conference (PACLING'97), Ohme, Japon, 2-5 September 1997, 1/1, pp. 15-22.

Blanchon H. (1994) *Perspectives of DBMT for monolingual authors on the basis of LIDIA-1, an implemented mockup*. Proc. of 15th International Conference on Computational Linguistics, COLING-94, 5-9 Aug. 1994, 1/2, pp. 115—119.

Boitet C., Réd. (1982) *"DSE-1"— Le point sur ARIANE-78 début 1982*. Contrat ADI/CAP-Sogeti/Champollion (3 vol.), GETA, Grenoble, janvier 1982, 616 p. (200 p. + annexes)

Boitet C., Guillaume P. & Quézel-Ambrunaz M. (1982) *ARIANE-78, an integrated environment for automated translation and human revision*. Proc. of COLING-82, Prague, July 1982, North-Holland, Ling. series 47, pp. 19—27.

Boitet C. (1994) *Dialogue-Based MT and self-explaining documents as an alternative to MAHT and MT of controlled languages*. Proc. of Machine Translation 10 Years On, 11-14 Nov. 1994, Cranfield University Press, pp. 22.21—29.

Boitet C. & Blanchon H. (1994) *Multilingual Dialogue-Based MT for Monolingual Authors: the LIDIA Project and a First Mockup*. Machine Translation, Vol. 9, N° 2, pp. 99—132.

Boitet C. (1997) *GETA's MT methodology and its current development towards personal networking communication and speech translation in the context of the UNL and C-STAR projects*. Proc. of PACLING-97, Ohme, 2-5 September 1997, Meisei University, pp. 23-57. (invited communication)

Brown R. D. (1989) *Augmentation*. (Machine Translation), Vol., N° 4, pp. 1299-1347.

Ducrot J.-M. (1982) *TITUS IV*. In *Information research in Europe. Proc. of the EURIM 5 conf. (Versailles)*, edited by Taylor P. J., London, ASLIB.

Kay M. (1973) *The MIND system*. In *Courant Computer Science Symposium 8: Natural Language Processing*, edited by Rustin R., New York, Algorithmics Press, Inc., pp. 155-188.

Lafourcade M. (2001) *Lexical sorting and lexical transfer by conceptual vectors*. Proc. of MMA'01, 29-31/1/01, SigMatics & NII, Tokyo, 10 p.

Lafourcade M. & Prince V. (2001) *Synonymy and conceptual vectors*. Proc. of NLPRS-2001, Tokyo, 27-3011/01, pp. 127-134.

Larousse (1999) *Thésaurus Larousse - des idées aux mots - des mots aux idées*. In *In extenso*. Larousse, Paris. Edited by Péchouin D., 1146 p. (2ième édition)

Maruyama H., Watanabe H. & Ogino S. (1990) *An Interactive Japanese Parser for Machine Translation*. Proc. of COLING-90, 20-25 août 1990, ACL, 2/3, pp. 257-262.

Melby A. K., Smith M. R. & Peterson J. (1980) *ITS : An Interactive Translation System*. Proc. of COLING-80, Tokyo, 30/9-4/10/80, pp. 424—429.

Moneimne W. (1989) *TAO vers l'arabe. Spécification d'une génération standard de l'arabe. Réalisation d'un prototype anglais-arabe à partir d'un analyseur existant*. Nouvelle thèse, UJF. (159 p. + annexes)

Nirenburg S. & al. (1989) *KBMT-89 Project Report*., Center for Machine Translation, Carnegie Mellon University, Pittsburg, April 1989.

Nyberg E. H. & Mitamura T. (1992) *The KANT system: Fast, Accurate, High-Quality Translation in Practical Domains*. Proc. of COLING-92, 23-28 July 92, ACL, 3/4, pp. 1069—1073.

Sérasset G. & Boitet C. (2000) *On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter*. Proc. of COLING-2000, Saarbrücken, 31/7—3/8/2000, ACL, 7 p. (submitted)

Slocum J. (1984) *METAL: the LRC Machine Translation system*. In *Machine Translation today: the state of the art (Proc. third Lugano Tutorial, 2–7 April 1984)*, edited by King M., Edinburgh University Press (1987).

Wehrli E. (1992) *The IPS System*. Proc. of COLING-92, 23-28 July 1992, 3/4, pp. 870-874.

Propagation de signatures lexicales dans le graphe du Web

Propagation of lexical signatures in the Web graph.

M. Bouklit[1] M. Lafourcade[2]

[1] Algorithmique et Combinatoire.
Laboratoire d'Informatique Algorithmique, Fondements et Applications.
Université Denis Diderot (case courrier 7014).
2, place Jussieu. 75251 Paris cedex 5 - France
[2] Traitements Algorithmiques du Langage.
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier.
161, rue Ada. 34392 Montpellier Cedex 5 - France

bouklit@liafa.jussieu.fr
lafourcade@lirmm.fr

Résumé

L'analyse du graphe formé par les pages web et les liens hypertextes qui les relient, communément appelé graphe du Web, a permis d'améliorer la performance des moteurs de recherche actuels. Ainsi, lancé en 1998, le moteur de recherche Google classe les pages grâce à la combinaison de plusieurs facteurs dont le principal porte le nom de PageRank. Nous présentons dans cet article l'algorithme LexicalRank propageant deux signatures lexicales : l'une interne, l'autre externe. Une signature lexicale est un ensemble de termes pondérés décrivant une page.

Mots Clef

Recherche d'informations, graphe du Web, PageRank, moteur de recherche, signature lexicale.

Abstract

Theorical analysis of the Web graph is often used to improve the efficience of search engines. The PageRank algorithm, proposed by Brin and Page in 1998 is used by Google search engine to improve the results of requests. In this paper, we present the LexicalRank algorithm which propagates two lexical signatures : one interns, the other external one. A lexical signature is a set of weighted terms describing a page.

Keywords

Information Retrieval, Web graph, PageRank, search engine, lexical signature.

1 Introduction

Le Web (ensemble des pages hypertextes disponibles sur Internet) est devenu une partie intégrante de la vie quotidienne de millions de gens. La nature même des médias électroniques, ainsi que la volonté de ses inventeurs [BLCL+94] lui ont donné une nature *hypertexte* : les documents sont structurés en *pages*, qui se *pointent* les unes vers les autres, par un système de références.

La croissance exponentielle du Web rend problématique l'appréhension de sa structure globale. Pourtant, une connaissance du contenu et de la structure du Web est indispensable pour réaliser de nombreuses tâches essentielles à la vie de l'internaute, telles que la *recherche d'informations* (où trouver une page sur tel sujet ?) ou la *mesure d'audience* (ma page est-elle populaire ?).

C'est pourquoi les moteurs de recherche ont développé des méthodes de tri automatique des résultats. Leur but est d'afficher dans les dix à vingt premières réponses les documents répondant le mieux à la question. Dans la pratique, aucune méthode de tri n'est parfaite, d'autant plus que la question de la justesse d'un classement est en grande partie subjective. Un classement est justifié au mieux par un sondage, le plus souvent au jugement du lecteur. Cependant, la variété des méthodes offre à l'usager la possibilité de traquer l'information de différentes manières : cette variété augmente donc ses chances d'améliorer ses recherches.
La suite de l'article est organisée comme suit. La section 2 décrit tout d'abord quelques méthodes de tri automatique des résultats comme *PageRank*, une mesure de popularité des pages Web. La section 3 introduit notre modèle *LexicalRank* et l'algorithme qui en est déduit. *LexicalRank*

propage dans le Web deux signatures lexicales : l'une interne, l'autre externe. Rappelons qu'une signature lexicale est un ensemble de termes pondérés décrivant une page. Nous pensons que l'utilisation de ces signatures permettront d'améliorer la performance des moteurs de recherche. La section 4 présente enfin les résultats issus de nos expérimentations.

2 Méthodes de tri automatique des résultats

2.1 Tri par contenu

La méthode de tri la plus ancienne et la plus utilisée est la méthode de tri par contenu : on la trouvait dans les moteurs Voila, Lycos, AltaVista, Excite, InfoSeek, ... Elle est basée sur le nombre d'occurrences des termes de la recherche dans les pages, de leur proximité, de leur place dans le texte [Sal89, YLYL95].

Malheureusement, cette méthode présente l'inconvénient d'être facile à détourner par des auteurs désireux de placer leurs pages en tête de liste : pour cela, il suffit de répéter les mots importants soit dans l'en-tête, soit dans le texte en utilisant des techniques de *spamming* (écrire le texte en blanc sur fond blanc par exemple) pour modifier à son avantage le classement.

2.2 PageRank

Les limites du tri par contenu ont alors conduit à rechercher, à partir de principes tout à fait différents, d'autres méthodes complémentaires indépendantes du contenu des documents. C'est dans ce contexte que sont apparues des méthodes de tri basées sur une notion de popularité.
En 1998, Sergei Brin et Larry Page alors étudiants en thèse à l'Université Stanford mettent au point une méthode qui va révolutionner le Web [PBMW98]. Cette méthode consiste à estimer la popularité des pages web en se servant de la structure induite par les pages web et les liens hypertextes qui les relient communément appelé *graphe du Web*. Plus précisément, elle classe les pages en utilisant un indice numérique (le «rang») calculé globalement pour chaque page d'où le nom *PageRank*. Ce rang donne en fait une *bonne* estimation de la popularité de la page. C'est ce même rang qui permettra en particulier d'ordonner les résultats d'une requête d'un usager. Quelques mois plus tard, le moteur de recherche Google [Goo98] voit le jour ...

Dans la suite, nous appellerons $G = (V, E)$ le graphe formé par les page web V et les liens hypertextes qui les relient E. N représentera le nombre de pages de V. En pratique, G est principalement obtenu par une succession de parcours du Web (*crawls*). En effet, il y a en amont du processus les *robots* qui chalutent continuellement le Web dans l'intention de découvrir de nouvelles pages et à défaut de mettre à jour les anciennes. Ces pages sont stockées dans un entrepôt de données. Viennent ensuite les hyperliens[1] qui sont stockés séparément pour former un sous graphe du Web[CGMP98]. Ce graphe est alors utilisé pour le calcul des rangs de page.

Le surfeur aléatoire. L'axiome caché derrière l'algorithme de PageRank est assez étrange, voire peu flatteur pour les internautes. Il dit que les pages les plus intéressantes sont celles sur lesquelles on a le plus de chance de tomber en cliquant au hasard. Exprimé autrement, le cerveau est un outil secondaire quand il s'agit de trouver les «bonnes» pages web. Toutes les variantes de PageRank peuvent s'interpréter comme un *surfeur aléatoire*, censé modéliser un internaute lambda, dont le comportement, bien qu'aléatoire, est soumis à certaines règles qui définissent la variante.

Le plus souvent, ces règles se traduisent par un processus stochastique de type markovien. A partir d'une distribution initiale de probabilité sur l'ensemble des pages web, le processus est itéré et, sous réserves de garanties de convergence et d'unicité de la limite, tend vers une distribution de probabilité qui est par définition le PageRank de la variante en question. On comprend donc qu'il existe en réalité une multitude de PageRanks même si on parle souvent du PageRank au singulier[BP98].

Modèle initial. Le niveau zéro du *surfeur aléatoire*, proposé par [PBMW98], suppose que notre internaute, quand il est sur une page donnée, va ensuite cliquer de manière équiprobable sur un des liens sortants.
Si $R_{n+1}(p)$ représente la probabilité de présence de notre surfeur à l'instant $n+1$ sur la page p, l'équation de propagation du rang s'écrit donc :

$$R_{n+1}(p) = \sum_{q \to p} \frac{R_n(q)}{deg(q)} \tag{1}$$

où $q \to p$ désigne "q pointe sur p" et où $deg(q)$ est le degré externe de q.

Vectoriellement, si on appelle M la matrice d'adjacence de G, et $A_{i,j} = \frac{M_{i,j}}{deg(i)}$ (par convention $0 = \frac{0}{0}$), l'équation de propagation se formule ainsi :

$$R_{n+1} = A^t R_n \tag{2}$$

Rechercher une distribution de probabilité sur V vérifiant (1) revient à trouver la distribution asymptotique de la chaîne de Markov homogène dont la matrice de transition est A. Si A est apériodique et irréductible[2], il est bien connu [SC96] que le processus itératif (2) converge géométriquement vers une distribution de probabilité R vérifiant (1) quelque soit la distribution de probabilité initiale Z. Dans ce cas, la matrice A est *stochastique* car c'est une matrice positive dont la somme de chacune des lignes vaut

[1] Notons que [PBMW98] ignore les ancres pour faciliter le calcul du PageRank.

[2] Une matrice est dite *irréductible* si son graphe est fortement connexe, et *apériodique* si le p.g.c.d. des longueurs des circuits est 1.

1. C'est précisément cette distribution de probabilité R obtenue par l'algorithme 1 que l'on appelle *PageRank*.
La norme ℓ_1 d'une matrice M (notée $\| M \|_1$) désigne l'application de $\mathcal{M}_{n,p}(\mathbb{K})$, par :

$$\| M \|_1 = \sum_{\substack{1 \le i \le n \\ 1 \le j \le p}} |M(i,j)|$$

où $\mathbb{K}$ désigne $\mathbb{R}$ ou $\mathbb{C}$.

Algorithme 1: PageRank : modèle originel [PBMW98]

Données
– une matrice irréductible et apériodique A ;
– une distribution de probabilité Z ;
– un réel ϵ.
Résultat
Le vecteur propre principal de probabilité de A^T avec une précision ϵ.
début
  $R_0 = Z$
  répéter
    $R_{n+1} = A^T R_n$
    $\delta = \|R_{n+1} - R_n\|_1$
  jusqu'à $\delta < \epsilon$
fin

Le facteur *zap*. Un graphe du Web n'est en général pas fortement connexe. En appliquant l'algorithme 1, [PBMW98] ont constaté que le PageRank remonte dans les composantes fortement connexes terminales d'où le nom de *puits de rang*. En plus des feuilles, il existe donc de nombreux puits de rang dont on ne peut pas sortir en cliquant [BKM+00]. Pour échapper aux puits de rang et aux feuilles, il est nécessaire «de temp en temps» de sauter aléatoirement vers une page quelconque du Web.

Méthode du rang par défaut. Pour modéliser les sauts aléatoires, [PBMW98] proposent de doter chaque page d'un rang par défaut appelé *source de rang*. Ainsi chaque page p se voit attribuer un rang de $Z(p) \geq 0$. (1) devient alors :

$$R_{n+1}(p) = \sum_{q \to p} \frac{R_n(q)}{deg(q)} + Z(p) \qquad (3)$$

L'écriture vectorielle de cette équation est $R_{n+1} = A^T R_n + Z$. Quand $||Z||_1 = 1$, Z représente une loi de distribution sur l'ensemble des pages de V appelée distribution de *zap*. Le plus souvent, on choisit pour Z une loi de distribution uniforme : $\forall p \in V$, $Z(p) = \frac{1}{|V|}$. Mais il a été proposé que cette distribution puisse être «personnalisée» [BMPW98].

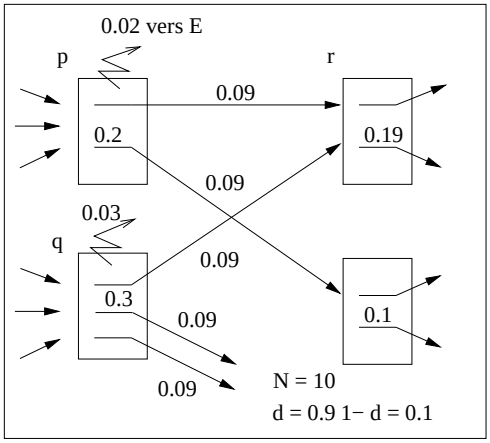


Figure 1: propagation de rang

Facteur d'amortissement. Nous supposerons ici que la matrice A est stochastique. L'équation (3) n'admet pas d'interprétation probabiliste directe. [BP98] propose une variante empirique du modèle en introduisant un facteur d'amortissement $d \in]0,1[$, ce qui donne :

$$R_{n+1}(p) = d \times \left(\sum_{q \to p} \frac{R_n(q)}{deg(q)} \right) + (1-d) Z(p) \qquad (4)$$

avec pour condition $\sum_{p \in V} Z(p) = 1$ et $\forall n$, $\sum_{p \in V} R_n(p) = 1$.

La figure 1 illustre une propagation de rang d'une paire de pages à l'autre. On y suppose $d = 0.9$ et $N = 10$. En observant la page p sur cette figure, nous remarquons que :
– d =90 % de son rang (soit 0.18) est redistribué équitablement sur ses liens sortants (soit $\frac{0.18}{2} = 0.09$) affectant ainsi le rang des pages pointés par p.
– $1 - d$ =10 % de son rang (soit 0.02) est dissipée au profit d'une répartition globale sur l'ensemble du graphe contribuant ainsi à alimenter chaque page d'un rang égal à $\frac{1-d}{N} = \frac{0.1}{10} = 0.01$.
Nous pouvons vérifier par exemple que le rang de la page r est bien 0.19 :

$$\begin{aligned} R_{n+1}(r) &= d\frac{R_n(p)}{d^+(p)} + d\frac{R_n(q)}{d^+(q)} + \frac{1-d}{N} \\ &= 0.9\frac{0.2}{2} + 0.9\frac{0.3}{3} + \frac{0.1}{10} \\ &= 0.19\,. \end{aligned}$$

L'écriture vectorielle de cette équation est :

$$R_{n+1} = dA^t R_n + (1-d)Z \qquad (5)$$

On espère approcher asymptotiquement la valeur de R vérifiant l'équation de conservation :

$$R = dA^T R + (1-d)Z \qquad (6)$$

Algorithme 2: PageRank : modèle par ajout d'un facteur *zap* [BP98]

Données
- une matrice stochastique A ;
- une distribution de *zap* recouvrante Z ;
- un coefficient de *zap* $d \in]0,1[$;
- un réel ϵ.

Résultat
Le vecteur propre de probabilité R de $\hat{A^T}$ associé à la valeur propre maximale à une précision ϵ.

début
  $R_0 = Z$
  répéter
    $R_{n+1} = d.A^T R_n + (1-d).Z$
    $\delta = \|R_{n+1} - R_n\|_1$
  jusqu'à $\delta < \epsilon$
fin

Soit la matrice $\hat{A} = dA + (1-d)\mathbf{1}^T \times Z^T$ où $\mathbf{1}$ désigne le vecteur ligne ne contenant que des 1. Remarquons que l'équation (6) peut être reformulée comme suit : $R = \hat{A}^T R$. En effet, comme $\sum_{p \in V} R(p) = 1$, on obtient alors : $\mathbf{1} \times R = I_1$.
Il en découle que : $Z = Z \times \mathbf{1} \times R$
Puisque nous avons supposé que A est stochastique alors dans ce cas R est un vecteur propre de la matrice $\hat{A^T}$ pour la valeur propre 1.
On obtient ainsi l'algorithme 2. Observons que l'initialisation R_0 du processus est égal à la distribution Z "par défaut" mais peut être choisie autrement. Par exemple, prendre le résultat d'un calcul précédent peut souvent accélerer la convergence. Cette remarque s'applique à la méthode précédente dite de rang par défaut, même si Z y a l'interprétation de «rang par défaut», voir (3). Le plus souvent, on imposera à toute distribution de *zap* d'être *recouvrante* : on garantit ainsi que toutes les pages connues sont potentiellement accessibles après un *zap*. Par exemple, la distribution uniforme est bien évidemment uniforme. Il est en de même de la distribution sur les pages d'acceuil à la seule condition que les pages connues d'un site soient accessibles à partir de la page d'acceuil.

Modèle du surfeur aléatoire. La définition de l'équation (5) peut s'interpréter dans un modèle de surfeur aléatoire plus évolué comme suit. A chaque étape, notre internaute lambda a la possibilité sur une page :

1. soit de *cliquer* sur l'un des liens sortants ($A^T R$) avec la probabilité d.
2. soit de *zapper*, avec la probabilité $1-d$ cette fois, sur une page choisie aléatoirement selon la distribution de Z.

Choix de d. Depuis l'article originel [PBMW98], 0.85, a toujours été une valeur de référence. Selon [Hav99], l'introduction du paramètre d est destinée à améliorer la «qualité» du PageRank en garantissant la convergence vers un unique vecteur rang. La matrice A est explicitement supposée stochastique en éliminant les pages sans liens. Le modèle probabiliste de l'internaute que nous décrivons comme support (implicite) au modèle PageRank prédit que le nombre de «clics» consécutifs suit une distribution géométrique de raison d. En particulier, la longueur moyenne d'un chemin entre deux *zap* vaut

$$\sum_{k=0}^{\infty} k d^k (1-d) = \sum_{k=1}^{\infty} d^k = \frac{d}{1-d}$$

Cela pourrait donner une façon empirique de trouver d. Pour $d = 0.85$, on en déduit une longueur moyenne entre deux zaps successifs d'environ 5.67. A titre de comparaison, différentes études donnent suivant l'époque et la méthode employée, des nombres variant entre 3 et 10 [CP95, WM04]. Plus supprenant, [MFJR+04] estime cette moyenne à 5,6 ...

3 Description du modèle

Nous présentons dans cette section l'algorithme *LexicalRank* qui propage dans le graphe G deux signatures lexicales : une interne et une externe. Nous espérons ainsi ouvrir la voie à une nouvelle famille d'algorithmes PageRank fondés sur la propagation de signatures lexicales.

Définition 1 (signature lexicale). *Une* signature lexicale *d'une page est un ensemble de termes décrivant une page.*

Formellement, la signature lexicale $S(p)$ d'une page p est un ensemble de termes pondérés permettant de caractériser thématiquement cette page : $S(p) = \{(t_1; p_1), (t_2; p_2), \ldots, (t_k; p_k)\}$
Par exemple, on peut avoir comme signature lexicale (d'une page hypothétique p parlant des différents sens de bottes) :

$S(p)$ = {(‹*chaussure*› ;0,9), (‹*assemblage*› ;0,68), (‹*végétaux*› ;0,4), (‹*coup*› ;0,35), (‹*réunion*› ;0,32), (‹*escrime*› ;0,2), (‹*fleuret*› ;0,14), (‹*épée*› ;0,13), (‹*balle*› ;0,09), (‹*bottine*› ;0,8), (‹*pied*› ;0,78), (‹*touffe*› ;0,05), (‹*attaque*› ;0,04), (‹*jambe*› ;0,04)}

On remarquera que cette signature ne comporte que des termes proches thématiquement des trois principales acceptions de ‹*botte*› : la chaussure, le coup et la réunion de végétaux.
Nous associons à chaque page deux signatures lexicales : l'une interne, l'autre externe.

Définition 2 (signature interne). *La* signature interne *d'une page p (notée $I(p)$) est la signature lexicale que souhaite donner l'auteur de la page p.*

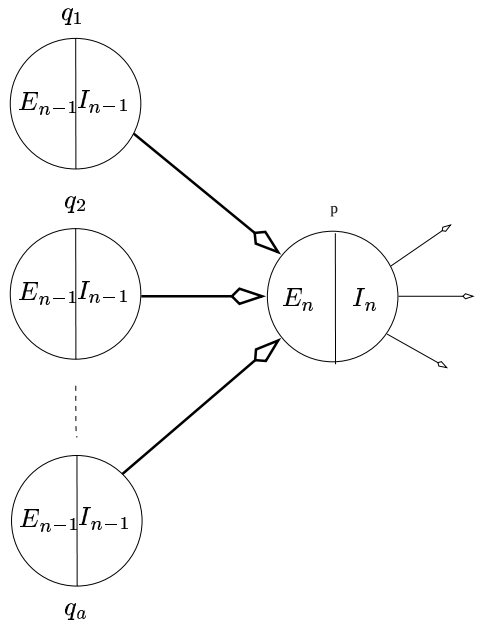


Figure 2: propagation avant

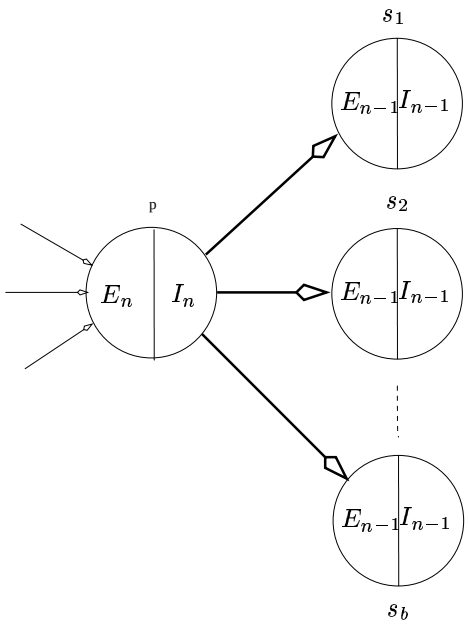


Figure 3: propagation arrière

Définition 3 (signature externe). *La* signature externe *d'une page p (notée $E(p)$) est la signature lexicale perçue par les auteurs des pages qui la pointent.*

Définition 4 (contenu). *Le contenu d'une page p (notée $C(p)$) est la signature lexicale caractérisant le contenu de la page p en dehors des liens hypertextes (contenu brut).*

Le contenu d'une page est une information que l'on peut compléter à l'aide des signatures interne et externe. Nous pensons que l'utilisation de telles signatures permettront d'améliorer la performance des moteurs de recherche. En effet, ce modèle peut nous permettre de réduire le spamming en otant les pages dont les signatures internes et externes sont éloignées. Ces signatures sont obtenues en appliquant itérativement les équations de *propagation avant et arrière*.

Définition 5 (Equation de *propagation avant*). *La signature externe d'une page à l'instant n est obtenue à partir des précedentes signatures internes des pages qui la pointent :*

$$E_0(p) \leftarrow \emptyset \tag{7}$$

et

$$E_n(p) \leftarrow f(I_{n-1}(q_1), \cdots, I_{n-1}(q_a)) \tag{8}$$

où $q_1, \ldots$ et q_a désignent les predécesseurs de p dans G (figure 2).

Définition 6 (Equation de *propagation arriere*). *La signature interne d'une page p à l'instant n est obtenue à partir de son contenu et des précedentes signatures externes des pages qu'elle pointe :*

$$I_0(p) \leftarrow C(p) \tag{9}$$

et

$$I_n(p) \leftarrow g(E_{n-1}(s_1), \cdots, E_{n-1}(s_b), C(p)) \tag{10}$$

où $s_1, \ldots$ et s_b représentent les successeurs de p dans G (figure 3).

De notre point de vue, la signature interne d'une page ne doit pas reposer uniquement sur le contenu de cette page. Elle doit aussi s'exprimer à partir des signatures externes des pages pointées. En effet, rappelons qu'un auteur est au moment de la création d'une page d'abord un internaute. Ce dernier construit sa page en fonction de la perception des pages qu'il estime nécessaires (signatures externes).

3.1 L'algorithme

L'algorithme 3 consiste à appliquer itérativement les deux équations de propagation des signatures lexicales (8) et (10). Initialement, pour chaque page p, sa signature interne $I_0(p)$ est égale à son contenu $C(p)$ et sa signature externe $E_0(p)$ est vide.

3.2 Calcul de $C(p)$

Pour calculer la signature $C(p)$ d'une page p nous utilisons les techniques classiques en usage en recherche d'informations. En particulier, nous nous basons sur le modèle TF×IDF. Le facteur *TF* correspond à la fréquence relative d'un terme donné dans une page. Le facteur *IDF* correspond à la fréquence inverse de ce terme sur l'ensemble du corpus. Formellement, nous avons pour chaque terme t :

$$\mathit{TFIDF}(t) = \mathit{TF}(t) \times \log(\frac{N}{DF(t)}) \tag{11}$$

Algorithme 3: LexicalRank

Données
– un graphe du Web $G = (V, E)$;
– un entier k.
Résultat
Signatures internes et externes des pages de V.
début
  pour $p \in V$ **faire**
    $I_0(p) = C(p)$
    $E_0(p) = \emptyset$
  fin
  pour $n = 1$ *à* k **faire**
    pour $p \in V$ **faire**
      Appliquer l'équation de propagation avant (8) aux predécesseurs de p dans G pour obtenir $E'_n(p)$
      Appliquer l'équation de propagation arrière (10) aux successeurs de p dans G pour obtenir $I'_n(p)$
    fin
    Normaliser le vecteur signature externe E'_n pour obtenir E_n
    Normaliser le vecteur signature interne I'_n pour obtenir I_n
  fin
  Retourner I_k et E_k les signatures interne et externe de V
fin

N est le nombre total de documents du corpus et $DF(t)$ est le nombre de pages contenant le terme t.

Généralement, la valeur de fréquence d'un terme correspond à son nombre d'occurences. En pratique, une occurence d'un terme voit son poids (par défaut égal à 1) augmenté ou diminué en fonction de sa postion dans la page. Par exemple, un terme présent dans le titre de la page verra son poids fortement augmenté. D'une façon générale, une heuristique satisfaisante consiste à privilégier plutôt les termes en début de page.

Notre approche est incrémentale, toutefois nous ne désirons pas recalculer les facteurs N et $DF(t)$ à chaque ajout de document. un telle approche sera impraticable. Nous adoptons donc une solution alternative approchée, en nous basant sur les informations fournies par le moteur de recherche Google. Le facteur N correspond au nombre de pages contenant le terme le plus fréquemment rencontré sur le Web jusque là. Le facteur $DF(t)$ lui correspond au nombre de pages contenant le terme t tel que renvoyé par Google. Par exemple, Google indique qu'il y a 3000000 de pages contenant le terme *chien*.

3.3 Fonction f

La fonction f est celle qui combine les signatures lexicales internes pour le calcul d'une signature externe. Il s'agit simplement, dans l'expérience que nous avons menée, d'une somme normalisée des termes pondérés.
Soit $I_{n-1}(q_i) = \{(t_1; p_1), (t_2; p_2), \ldots (t_k; p_k)\}$ la signature interne d'une page q_i, nous définissons la somme normalisée de signatures lexicales comme suit :

$$f(I_{n-1}(q_1), I_{n-1}(q_2) \ldots I_{n-1}(q_a)) = \frac{I_{n-1}(q_1) \oplus I_{n-1}(q_2) \oplus \ldots \oplus I_{n-1}(q_a)}{||I_{n-1}(q_1) \oplus I_{n-1}(q_2) \oplus \ldots \oplus I_{n-1}(q_a)||} \quad (12)$$

où l'opérateur $\oplus$ désigne l'union *ensembliste* de deux ensembles de termes pondérés.
Par exemple, soient :

$I(p_1)$ = {(‹*chaussure*› ;0,9), (‹*assemblage*› ;0,68), (‹*végétaux*› ;0,4), (‹*coup*› ;0,35), (‹*réunion*› ;0,31), (‹*escrime*› ;0,2), (‹*fleuret*› ;0,14), (‹*épée*› ;0,13)}

et

$I(p_2)$ = {(‹*chaussure*› ; 0,5), (‹*végétaux*› ; 0,68), (‹*bottine*› ; 0,6), (‹*coup*› ;0,35), (‹*pied*› ;0,32), (‹*viande*› ;0,2), (‹*fleur*› ;0,14), (‹*épée*› ;0,13)}

alors

$I(p_1) \oplus I(p_2)$ = {(‹*chaussure*› ;1.4), (‹*assemblage*› ;0,68), (‹*végétaux*› ;1.08), (‹*bottine*› ;0,6), (‹*coup*› ;0,7), (‹*pied*› ;0,32), (‹*réunion*› ;0,31), (‹*escrime*› ;0,2), (‹*fleuret*› ;0,14), (‹*épée*› ;0,26), (‹*viande*› ;0,2), (‹*fleur*› ;0,14)}

3.4 Fonction g

La fonction g est celle qui combine les signatures lexicales externes et le contenu de la page pour le calcul d'une signature interne. Il s'agit simplement, dans l'expérience que nous avons menée, d'une somme normalisée des termes pondérés des signatures externes (correspondant aux URL) d'une part, et de la signature du contenu d'autre part. En l'absence d'une étude plus poussée, le poids global des signatures externes est équivalent au poids du contenu.

$$g(E_{n-1}(s_1), E_{n-1}(s_2) \ldots E_{n-1}(s_b)) = \frac{E_{n-1}(s_1) \oplus E_{n-1}(s_2) \oplus \ldots \oplus E_{n-1}(s_b)}{||E_{n-1}(s_1) \oplus E_{n-1}(s_2) \oplus \ldots \oplus E_s(s_b)||} \oplus C(p) \quad (13)$$

Cette formule est simplifiée dans la mesure où à chaque URL (contenue dans la page) on associe le même poids. En pratique, le poids de chaque URL dépend de sa position dans le document (de la même manière que pour le calcul de $C(p)$).

4 Résultats

Un logiciel implémentant *LexicalRank* nous permet de nous promener dans le graphe de page en page. Notre graphe de travail est un site web consacré à l'étude des réseaux pairs à pairs. Les arcs du graphe ne reflètent donc que la structure du site (qui présente une cohérence assez forte).

Le logiciel permet de suivre l'évolution au cours des itérations de l'algorithme des signatures interne et externe de la page courante . Les tables 2 à 4 illustrent l'évolution des signatures lexicales d'une page Web introduisant les réseaux pairs à pairs. Certaines pages web dans le site y font référence. En quelques itérations, nous avons constaté sur certaines pages l'émergence dans leurs signatures lexicales de termes liés implicitement à la page. En effet, dès les premières itérations les termes *exploration* et *identification* apparaissent en tête de la signature externe de la page dans la mesure un certain nombre de pages dans le site traite de la difficile problématique de l'exploration de ces réseaux ou de l'identication des pairs et font référence à la première page.

Par ailleurs, on constate dans certains cas que les termes associés à des liens navigationnels (*monter*, *suivant* ou *précédent*) se retrouvent en bas des signatures lexicales.

5 Conclusion et perspectives

Nous avons présenté dans cet article l'algorithme *LexicalRank* propageant deux signatures lexicales : l'une interne, l'autre externe. Nous espérons ainsi ouvrir la voie à une nouvelle famille d'algorithmes PageRank fondés sur la propagation de signatures lexicales. Nous avons obtenu des résultats très prometteurs. Nous avons constaté au cours de l'exécution de l'algorithme l'emergence de termes précis caractérisant implicitement la page.

Dans nos expérimentations, nous avions choisi des fonctions f et g qui soient dans un premier temps combinaison linéaire des signatures lexicales. D'autres fonctions plus élaborées sont actuellement à l'étude. Nous projetons enfin une validation complète de *LexicalRank* en l'incorporant dans un moteur de recherche et en effectuant une série de tests de satisfaction sur une population témoin.

rang	terme	poids
1	‹*réseau*›	58,20
2	‹*clients*›	39,62
3	‹*précédent*›	37,64
4	‹*matières*›	37,64
5	‹*participant*›	31,46
6	‹*rapport*›	29,77
7	‹*fichier*›	29,51
8	‹*interconnectés*›	24,67
9	‹*généralement*›	24,67
10	‹*utilisé*›	24,67
11	‹*manière*›	24,67
12	‹*décentralisé*›	24,67
13	‹*possédant*›	24,67
14	‹*répondant*›	24,67
15	‹*constitué*›	24,67
16	‹*échanger*›	24,57
17	‹*adresse*›	22,11
18	‹*IP*›	21,21
19	‹*principale*›	20,98
20	‹*fonction*›	20,98
21	‹*appelé*›	20,67
22	‹*chaque*›	20,38
23	‹*Internet*›	18,49
24	‹*certains*›	18,49
25	‹*trouver*›	18,47
26	‹*client*›	17,21
27	‹*connecte*›	16,54
28	‹*programme*›	13,51
29	‹*type*›	12,28
30	‹*pair*›	12,17
31	‹*fichiers*›	3,21
32	‹*monter*›	0,76
33	‹*suivant*›	0,05

Table 1: signature interne à l'itération 2.

References

[BKM+00] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proc. 9th International World Wide Web Conference*, pages 309–320, 2000.

[BLCL+94] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The world wide web. *Communications of ACM*, 37(8) :76–82, 1994.

[BMPW98] S. Brin, R. Motwani, L. Page, and T. Winograd. What can you do with a Web in your Pocket ? *Data Engineering Bulletin*, 21(2) :37–47, 1998.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7) :107–117, 1998.

[CGMP98] Junghoo Cho, Hector García-Molina, and Lawrence Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1–7) :161–172, 1998.

[CP95] Lara D. Catledge and James E. Pitkow. Characterizing browsing strategies in the world wide web. *Computer Networks and ISDN Systems*, 27(6) :1065–1073, 1995.

[Goo98] Google. http ://www.google.com/, 1998.

[Hav99] T. Haveliwala. Efficient computation of PageRank. Technical report, Computer Science

rang	terme	poids
1	‹*réseau*›	88,25
2	‹*clients*›	79,90
3	‹*précédent*›	78,78
4	‹*matières*›	78,78
5	‹*participant*›	74,89
6	‹*rapport*›	73,6
7	‹*fichier*›	73,50
8	‹*constitué*›	69,61
9	‹*interconnectés*›	69,61
10	‹*appelé*›	69,61
11	‹*généralement*›	69,61
12	‹*utilisé*›	69,61
13	‹*échanger*›	69,61
14	‹*décentralisé*›	69,61
15	‹*possédant*›	69,61
16	‹*critères*›	69,61
17	‹*manière*›	69,51
18	‹*fonction*›	68,09
19	‹*adresse*›	67,22
20	‹*IP*›	66,33
21	‹*principale*›	66,09
22	‹*chaque*›	65,46
23	‹*Internet*›	63,34
24	‹*trouver*›	63,34
25	‹*certains*›	63,34
26	‹*connecte*›	60,93
27	‹*répondant*›	59,51
28	‹*programme*›	58,53
29	‹*type*›	54,42
30	‹*pair*›	54,26
31	‹*client*›	51,79
32	‹*fichiers*›	25,33
33	‹*monter*›	0,91
34	‹*suivant*›	0,42

Table 2: signature interne à l'itération 3.

rang	terme	poids
1	‹*exploration*›	94,44
2	‹*protocole*›	93,57
3	‹*identification*›	93,55
4	‹*matières*›	89,78
5	‹*diagramme*›	87,22
6	‹*problématique*›	83,86
7	‹*fonctionnalités*›	83,66
8	‹*tête*›	80,02
9	‹*implémentation*›	79,97
10	‹*noeuds*›	79,24
11	‹*introduction*›	79,23
12	‹*decentralisé*›	79,19
13	‹*hybride*›	79,17
14	‹*modèle*›	78,45
15	‹*architectures*›	77,68
16	‹*centralisé*›	76,68
17	‹*architecture*›	75,99
18	‹*rapport*›	74,43
19	‹*système*›	74,14
20	‹*differentes*›	73,78
21	‹*table*›	71,81
22	‹*applications*›	71,36
23	‹*super*›	70,77
24	‹*définition*›	70,62
25	‹*réseaux*›	69,54
26	‹*réseau*›	68,85
27	‹*historique*›	68,25
28	‹*index*›	66,67
29	‹*pair*›	65,72
30	‹*suivant*›	54,33
31	‹*précédent*›	39,25
32	‹*monter*›	38,72

Table 3: signature externe à l'itération 2.

Department, Stanford University, 1999.

[MFJR+04] Natasa Milic-Frayling, Rachel Jones, Kerry Rodden, Gavin Smyth, Alan Blackwell, and Ralph Sommerer. Smartback : supporting users in back navigation. In *Proceedings of the 13th international conference on World Wide Web*, pages 63–71. ACM Press, 2004.

[PBMW98] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking : Bringing Order to the Web. Technical report, Computer Science Department, Stanford University, 1998.

[Sal89] G. Salton. Automatic text processing. Massachusetts, 1989.

[SC96] L. Saloff-Coste. Lectures on finite Markov chains. In G.R. Grimmet E. Giné and L. Saloff-Coste, editors, *Lecture Notes on Probability Theory and Statistics*, number 1665 in LNM, pages 301–413. Springer Verlag, 1996.

[WM04] Long Wang and Christoph Meinel. Behaviour recovery and complicated pattern definition in web usage mining. In *ICWE*, pages 531–544. LNCS, 2004.

[YLYL95] Budi Yuwono, Savio L. Lam, Jerry H. Ying, and Dik L. Lee. A world wide web resource discovery system. In *Proc. 4th International World Wide Web Conference*, December 1995.

Rang	terme	poids
1	‹*exploration*›	94,59
2	‹*protocole*›	83,97
3	‹*identification*›	83,96
4	‹*introduction*›	79,23
5	‹*decentralisé*›	79,23
6	‹*hybride*›	79,23
7	‹*noeuds*›	79,21
8	‹*architecture*›	77,71
9	‹*architectures*›	77,44
10	‹*matières*›	77,37
11	‹*table*›	77,00
12	‹*rapport*›	76,97
13	‹*suivant*›	76,92
14	‹*centralisé*›	76,71
15	‹*des*›	76,64
16	‹*index*›	75,60
17	‹*système*›	74,17
18	‹*differentes*›	73,11
19	‹*définition*›	70,55
20	‹*super*›	70,51
21	‹*applications*›	70,47
22	‹*réseau*›	69,98
23	‹*réseaux*›	68,19
24	‹*historique*›	66,12
25	‹*pair*›	64,50
26	‹*précédent*›	42,69
27	‹*monter*›	42,68

Table 4: signature externe à l'itération 3.

Artificial Ants for Natural Language Processing

Mathieu Lafourcade[1] , Frédéric Guinand[2]

[1] LIRMM (CNRS - Université Montpellier 2)

mathieu.lafourcade@lirmm.fr

[2] Laboratoire d'Informatique du Havre

frederic.guinand@univ-lehavre.fr

Abstract

The global sense of a text results from the interactions between the different meanings of the words that compose it. Classical approaches for applications like Word Sense Disambiguation (WSD) rely on the use of conceptual vectors that represent word meanings by concept activations. The process for determining the global sense of a text consists in propagating the different vectors within the tree that represents the syntactic structure of the text (called morphosyntactic analysis tree). Unfortunately, such strategies do not take into account interpretation trails, and constraints between word senses. We propose a new approach for this problem by considering a text in natural language as a complex system in which interacting entities are together the meanings of the words and the various elements stemmed from the text analysis. In order to express these interactions, artificial ants are used as conveyors of the meanings, besides able to modify the structure of the text by transforming the initial morphosyntactic analysis tree. The modifications are produced by ants according to both kind of stigmergy: sign-based and sematectonic stigmergies. Early experiments gave hints that beside being cognitively motivated on some aspects, this approach can perform in some difficult cases very well and is a generalization of the standard propagation.

1 Introduction

Whatever the considered domain, physics, biology, mathematics, social sciences, chemistry, computer science... there exist numerous examples of systems exhibiting global properties that emerge from the interactions between the entities that compose the system itself. Shoal of fishes, flocks of birds,bacteria colonies,sand piles [Bak 1996], cellular automata [Wolfram 1984], protein interaction networks [Evry Spring School 2004], city formation, human languages are some such examples. These systems are called *complex systems* [Bossomaier and Green 2000]. They are opened, crossed by various flows and their compounds are in interaction with themselves and/or with an environment that do not belong to the system itself. They exhibit a property of self-organization that can be defined as an holistic and dynamic process allowing such systems to adapt themselves to the static characteristics as well as dynamic changes of

the environment in which their compounds move and act. The work presented in this paper aims at exploiting that self-organization property for computing solutions for a natural language processing problem, for which it is unlikely to build a good global evaluation function. The main idea consists in the conception of a complex system based on artificial ants that move in and act on a graph (initially the morphosyntactic analysis tree) in which we are looking for structures of special interest: solutions of our original problem. In order to answer our expectations, the entities must leave some marks in the environment and these marks should define expected structures. In our case, two kinds of marks are left in the graph by artificial ants: pheromones for indicating the interest of certain paths, and structural marks materialized by edges (called bridges in the sequel) for building shortcuts between terms geographically distant but semantically close.

In the framework of the Word Sense Disambiguation (WSD) and lexical transfer in Machine Translation (MT), the representation of word meaning is one main issue. The conceptual vector model aims at representing thematic activations for chunks of text, lexical entries, locutions up to whole documents. Roughly speaking, vectors are supposed to encode *ideas* associated to words or expressions. The main applications of the model are thematic text analysis and lexical disambiguation [Lafourcade 2001]. Practically, [Lafourcade *et al.* 2002] presents a system, with automated learning capabilities, based on conceptual vectors and exploiting monolingual dictionaries (available on the web). So far, from French, the system learned 110000 lexical entries corresponding to roughly 430000 vectors (the average meaning number for polysemous word being 5.1). The same experiment is conducted for English. The issue addressed in this paper concerns the analysis process itself.

Understanding a text requires the comparison of the various meanings of any polysemous word with the context of its use. But, the context is precisely defined by the words themselves with all their meanings and their status in the text (verb, noun, adjective...). Given their status, there exist some grammatical relations between words. These relations can be represented by a tree-like structure called a morpho-syntactic analysis tree. This structure represents one kind of relations between words, but in no way their semantics relations. In order to represent this new relationnal structure, we consider words as the basic elements of an interaction network which implicit dynamics reveals the pregnancy of each meaning associated to any polysemous word. If we refer to the most commonly shared definition of a *complex system*, it states that it is a *network of entities which interactions lead to the emergence of structures that can be identified as higher-level organisations*. The action of one entity may affect subsequent actions of other entities in the network, so that *the action of the whole is more than the simple sum of the actions of its parts.*[1].

The *actions* in our context correspond to the *meanings* of the words constituting the text, and the sum of the actions results in the global meaning of the text, which is, for sure, much more than the simple sum of the meanings of the words. Then, in itself, a text constitutes a complex system. The computational problem is that the meanings are not strictly speaking active elements. In order to ensure the dynamicity of the whole system, an active framework made of ”meaning transporters” must be supplied

[1] [Langton 1996] *Why do we need artificial life ?* page 305.

to the text. These ”transporters” are intended to allow the interactions between text elements. They have to be both light (because of their possible large number) and independent (word meanings are intrinsic values). Moreover, when some meanings stemmed from different words are compatible (*engaged* with *job* for instance), the system has to keep a trace of this fact. This trace may be for instance a link between terms which are semantically close, whatever their relative distance in the text. This was the main motivation for using artificial ants, their ability to transform their local and indirect interactions in numerical as well as structural marks in their environment that is initially the morphosyntactic analysis tree.

Ants algorithms or variants of them have been classically used for classical optimization problems; traveling salesman problem (TSP) [Dorigo and Gambardella 1997], routing problems [Bruten *et al.* 1996, Di Caro and Dorigo 1998], dynamic load balancing [Bertelle *et al.* 2004], graph coloring [Costa and Hertz 1997], and for computational molecular biology problems; protein identification [Gras *et al.* 2002] or DNA--Sequencing using Sequencing-by-Hybridization [Bertelle *et al.* 2002], but they were never used in Natural Language Processing (NLP). Most probably because NLP was neither modeled as an optimization problem, nor explicitly modeled as a complex system. However, [Hofstadter 1995] with the COPYCAT project, presented an approach where the environment by itself contributed to solution computation and is modified by an agent population where roles and motivations varies. In [Gale 1992], Gales, Church and Yarowsky have used Naive-Bayes algorithm for WSD. Some properties of these models seem to be adequate for the task of semantic analysis and WSD, where word senses can be seen as competing for resources. We retain here some aspects that we consider as being crucial: (1) mutual information or semantic proximity is one key factor for lexical activation, (2) the syntactic structure of the text can be used to guide information propagation, (3) conceptual bridges can be dynamically constructed (or deleted) and could lead to *catastrophic events* (in the spirit of [Thom 1972]). These bridges are the instrumental part allowing mutual-information exchange beyond locality horizons. Finally, as pointed by [Hofstadter 1995], biased randomization (which doesn't mean chaos) plays a major role in the model.

In this paper, we first expose the conceptual vectors model and the notions of semantic distance and contextualization. Then, we detail the text analysis process coupled with conceptual vectors, named Standard Propagation (SP), which is used for text classification, thematic analysis and vector learning. After analyzing some drawback of the SP approach, we present the Colored Ant (CA) Algorithm. CA includes SP in its behavior but extends it with interpretation trail creation and prepositional phrases attachment. We show, that the overall process is by essence emergent.

2 Conceptual Vectors

Thematic aspects of textual segments (documents, paragraphs, syntagms, etc.) can be represented by conceptual vectors. Vectors have been used in information retrieval for long [Salton and MacGill 1983] and for meaning representation by the LSI model [Deerwester *et al.* 1990] from latent semantic analysis (LSA) studies in psycholinguistics. In computational linguistics, [Chauché 1990] proposes a formalism for the pro-

jection of the linguistic notion of semantic field in a vectorial space, from which this model is inspired. From a set of elementary notions, concepts, it is possible to build vectors (conceptual vectors) and to associate them to lexical items[2]. The hypothesis that considers a set of concepts as a generator to language has been long described in [Rodget 1852] (*thesaurus hypothesis*). Polysemous words combine the different vectors corresponding to the different meanings. This vector approach is based on well known mathematical properties, it is thus possible to undertake well founded formal manipulations attached to reasonable linguistic interpretations. Concepts are defined from a thesaurus (in the prototype applied to French, [Larousse 1992] has been chosen where 873 concepts are identified. This figure is comparable both quantitatively and qualitatively with the thousand defined in [Rodget 1852] for English). To be consistent with the thesaurus hypothesis, we consider that this set constitutes a generator space for the words and their meanings. This space is probably not free (no proper vectorial base) and as such, any word would project its meaning on this space.

2.1 Thematic Projection Principle

Let be $\mathcal{C}$ a finite set of n concepts, a conceptual vector V is a linear combination of elements c_i of $\mathcal{C}$. For a meaning A, a vector $V(A)$ is the description (in extension) of activations of all concepts of $\mathcal{C}$. For example, the different meanings of ‹*quotation*› could be projected on the following concepts (the CONCEPT[*intensity*] are ordered by decreasing values): V(‹*quotation*›) = *STOCK EXCHANGE*[*0.7*], *LANGUAGE*[*0.6*], *CLASSIFICATION*[*0.52*], *SYSTEM*[*0.33*], *GROUPING*[*0.32*], *ORGANIZATION*[*0.30*], *RANK*[*0.330*], *ABSTRACT*[*0.25*], ...

In practice, the largest $\mathcal{C}$ is, the finer the meaning descriptions are. In return, the computer manipulation is less easy. It is clear, that for dense vectors[3] the enumeration of the activated concepts is long and difficult to evaluate. We would generally prefer to select the thematically closest terms, i.e., the *neighborhood*. For instance, the closest terms ordered by increasing distance of ‹*quotation*› are: $\mathcal{V}$(‹*quotation*›) = ‹*management*›, ‹*stock*›, ‹*cash*›, ‹*coupon*›, ‹*investment*›, ‹*admission*›, ‹*index*›, ‹*abstract*›, ‹*stock-option*›, ‹*dilution*›, ...

2.2 Angular Distance

Let us define $Sim(A, B)$ as one of the *similarity* measures between two vectors A et B, often used in information retrieval. We can express this function as the scalar product of their vector divided by the product of their norm. We suppose here that vector components are positive or null. Then, we define an *angular distance* D_A between two vectors A and B as:

$$D_A(A, B) = \arccos(Sim(A, B))$$
$$\text{with} \quad Sim(A, B) = \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{1}$$

[2]Lexical items are words or expressions which constitute lexical entries. For instance, ‹*car*› or ‹*white ant*› are lexical items. In the following we will use sometimes *word* or *term* to speak about a *lexical item*.

[3]Dense vectors are those which have very few null coordinates. In practice, by construction, all vectors are dense.

Intuitively, this function constitutes an evaluation of the *thematic proximity* and is the measure of the angle between the two vectors. We would generally consider that, for a distance $D_A(A, B) \leq \frac{\pi}{4}$, (i.e. less than 45 degrees) A and B are thematically close and share many concepts. For $D_A(A, B) \geq \frac{\pi}{4}$, the thematic proximity between A and B would be considered as loose. Around $\frac{\pi}{2}$, they have no relation. D_A is a real distance function. It verifies the properties of reflexivity, symmetry and triangular inequality. We can have, for example, the following angles:

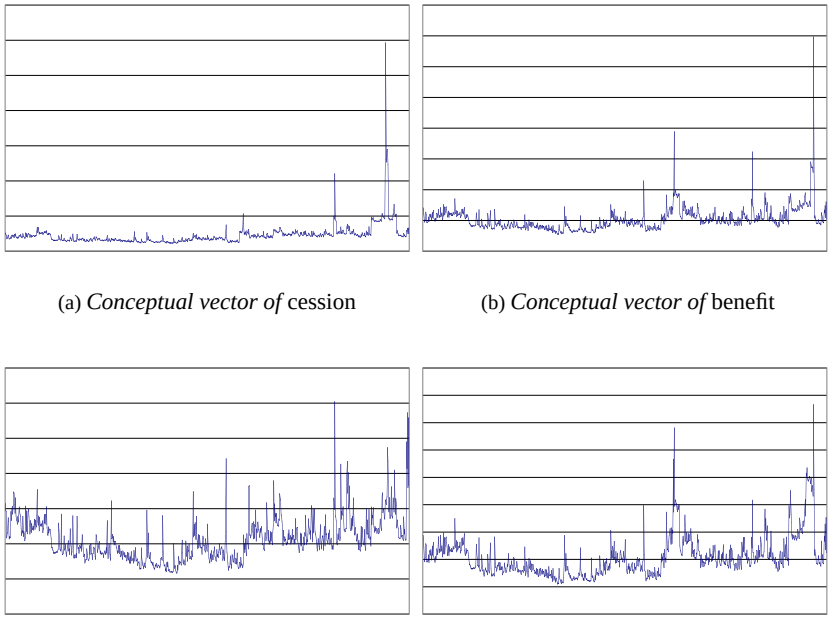
D_A(‹*profit*›, ‹*profit*›)=0°	D_A(‹*profit*›, ‹*product*›)=32°
D_A(‹*profit*›, ‹*benefit*›)=10°	D_A(‹*profit*›, ‹*goods*›)=31°
D_A(‹*profit*›, ‹*finance*›)=19°	D_A(‹*profit*›, ‹*sadness*›)=65°
D_A(‹*profit*›, ‹*market*›)=28°	D_A(‹*profit*›, ‹*joy*›)=39°

The first value has a straightforward interpretation, as ‹*profit*› cannot be closer to anything else than itself. The second and the third are not very surprising since a ‹*benefit*› is quite synonymous of ‹*profit*›, in the ‹*finance*› field. The words ‹*market*›, ‹*product*› and ‹*goods*› are less related, which explains a larger angle between them. The idea behind ‹*sadness*› is not much related to ‹*profit*›, contrary to its antonym ‹*joy*› which is thematically closer (either because of metaphorical meanings of ‹*profit*› or other semantic relations induced by the definitions). The thematic proximity is by no way an ontological distance but a measure of how strongly meanings may relate to each others.

The graphical representations of the vectors of ‹*exchange*› and ‹*profit*› shows that these terms are indeed quite polysemous. Two other terms (‹*cession*› and ‹*benefit*›) seems to be more focused on specific concepts. These vectors are the average of all possible meanings of their respective word in the general Thesaurus. It is possible to measure the level of *fuzziness* of a given vector as a clue of the number of semantic fields the word meaning is related to.

Because of the vagueness related either to polysemy or to lacks of precision (only 873 general concepts), vectors have to be *plunged* into a specialized semantic space. However, one cannot cut loose from the general vectors for two reasons. First, even non-specialized words may turn out to be pivotal in word sense disambiguation of specialized ones. Second, we cannot know beforehand if a given occurrence of a word should be understood in its specialized acception or a more general one.

One would certainly consider that the angle between two vectors can be regarded as a similarity measure, and *of course* the cosine between vectors (or $1 - \cos$ if a metric is required) is also a similarity measure, and that's everyone else uses. This remark is by itself correct, but we would like to stress that using the *arcos* metric lead to a more discriminant function at small angles. Precisely, generally we do consider the limit of the *cos* metric at 0.5, below this value there is no much common information. With *arcos*, we set this limit at $\pi/4$ which corresponds to a value of $\sqrt{2}/2$. What is really important is to be able to discriminate strongly between objects and certainly not in a linear way, as already distant objects are not to be scrutinized contrary to seemingly closed ones.



(a) *Conceptual vector of* cession

(b) *Conceptual vector of* benefit

(c) *Conceptual vector of* exchange

(d) *Conceptual vector of* profit

Figure 1: *Graphical representation of the vectors of four terms. It is noteworthy that bottom terms (*exchange and profit*) are more polysemous than top ones.*

2.3 Vector Operators

Vector Sum. Let X and Y be two vectors, we define their *normed sum* V as:

$$V = X \oplus Y \quad | \quad v_i = (x_i + y_i)/\|V\| \tag{2}$$

This operator is idempotent and we have $X \oplus X = X$. The null vector $\vec{\mathbf{0}}$ is by definition the neutral element of the vector sum. Thus we write down that $\vec{\mathbf{0}} \oplus \vec{\mathbf{0}} = \vec{\mathbf{0}}$.

Normed Term to Term Product. Let X and Y be two vectors, we define V as *their normed term to term product*:

$$V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{3}$$

This operator is idempotent and $\vec{\mathbf{0}}$ is absorbent. We have: $V = X \otimes X = X$ and $V = X \otimes \vec{\mathbf{0}} = \vec{\mathbf{0}}$.

Contextualization. When two terms are in presence of each other, some of the meanings of each of them are thus selected by the presence of the other, acting as a context. This phenomenon is called *contextualization*. It consists in emphasizing common features of every meaning. Let X and Y be two vectors, we define $\Gamma(X, Y)$ as the contextualization of X by Y as:

$$\Gamma(X,Y) = X \oplus (X \otimes Y) \tag{4}$$

These functions are not symmetrical. The operator Γ is idempotent ($\Gamma(X,X) = X$) and the null vector is the neutral element. ($\Gamma(X,\vec{\mathbf{0}}) = X \oplus \vec{\mathbf{0}} = X$). We will notice, without demonstration, that we have thus the following properties of *closeness* and of *distance*):

$$\begin{aligned} & D_A(\Gamma(X,Y),\Gamma(Y,X)) \\ \leq \{D_A(X,\Gamma(Y,X)), & D_A(\Gamma(X,Y),Y)\} \leq D_A(X,Y) \end{aligned} \tag{5}$$

The function $\Gamma(X,Y)$ brings the vector X closer to Y proportionally to their intersection. The contextualization is a low-cost way of amplifying properties that are salient in a given context. For a polysemous word vector, if the context vector is relevant, one of the possible meanings is *activated* through contextualization. For example, *bank* by itself is ambiguous and its vector is pointing somewhere between those of *river bank* and *money institution*. If the vector of *bank* is contextualized by *river*, then concepts related to finance would considerably be dimmed.

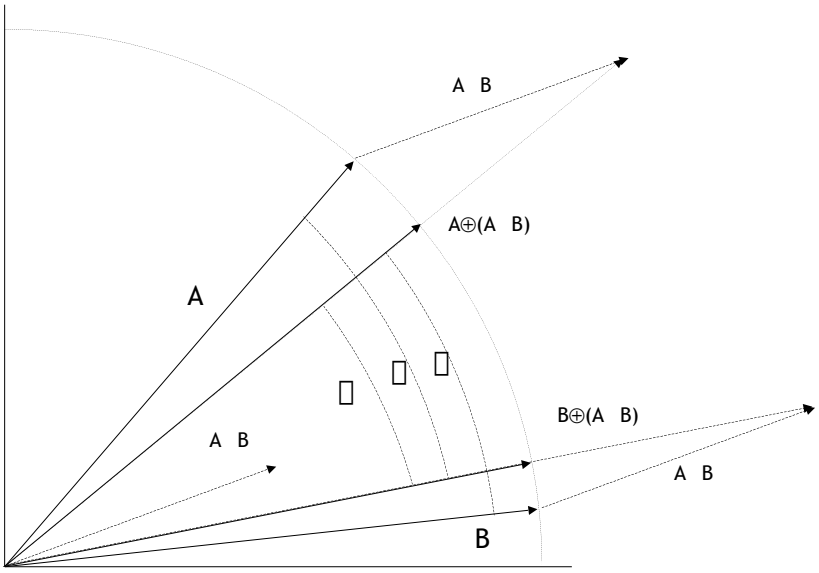


Figure 2: Geometric representation (in 2D) of the contextualization function. The α angle represents the distance between A and B contextualized by each other.

2.4 From Text to Vectors by Standard Propagation

How do we build a conceptual vector from a given text? From this text, we first compute a morphosyntactic analysis tree. This is a derivation tree from where leaves (roughly) reconstitute the original sentence. A leaf refers to a word on which are associated one or several definitions (as found in dictionaries) and a conceptual vector. For simplicity, we only consider contents words, that is nouns, verbs, adjectives and adverbs. After filtering according to agreement on morphosyntactic attributes, is attached to the leaf an *uncontextualized global* conceptual vector computed from the vectors of its k definitions. The most straightforward way (not the best) to do so is to compute the average vector: $V(w) = V(w.1) \oplus \cdots \oplus V(w.k)$. If the word is unknown (i.e. it is not in the dictionary), the null vector is taken instead.

Vectors are then propagated upward. Consider a tree node N with p dependents $N_i (1 \leq ip)$. The newly computed vector of N is the weighted sum of all the vectors of N_i: $V(N) = \alpha_i N_1 \oplus \cdots \oplus \alpha_p N_p$. The weights α depend of the syntactic functions of the node. For instance a governor would be given a higher weight ($\alpha = 2$) than a regular node ($\alpha = 1$) so that, for example, the vectors computed for *a boat sail* and for *a sail boat* would not be identical. Once the vector of the tree root is determined a downward propagation is performed. A node vector is contextualized by its parent: $V'(N_i) = V(N_i) \oplus \Gamma(N_i, N)$. This descent is done recursively until reaching a leaf. At the leaf level an implicit WSD process is undertaken as the new *contextualized global* vector is then a weighted sum of the vector of the definitions where weights are non-linearly related to the amount of mutual information between the context (node N) and a given meaning:

$$\begin{aligned} V'(w) &= \beta_i V(w.1) \oplus \cdots \oplus \beta_i V(w.k) \\ \text{with} \qquad \beta_i &= \cot(D_A(V(N), V(w.i)) \end{aligned} \tag{6}$$

If the vector context $V(N)$ is very close to $w.i$, then the global vector $V(w)$ for the word w is almost equal to $V(w.i)$ (we recall that *cot* refers to the *cotangent* function, with $\cot(0) = +\infty \quad$ and $\quad \cot(\pi/2) = 0$).

The processes of upward and downward propagation are iterated until either a maximum number of cycles is reached or when the root vector stabilizes (it is proved that in all generality there is no convergence as sometimes oscillations happen with strongly ambiguous sentences).

One major drawback of this analysis model is that the various interpretations are merged together, and constraints between selected word senses are not structurally represented.

2.5 Conceptual Vector Learning

Before processing texts, a lexicon associating terms and vectors should be constructed. Beside full manual indexing, which is difficult and time consuming, a supervised learning can be devised.

The learning process is an *ever-going* task that consists in picking randomly a word to be learned (or revised). The vector of each definition of this word is then computed

weevil : *n* a small beetle that spoils grain.

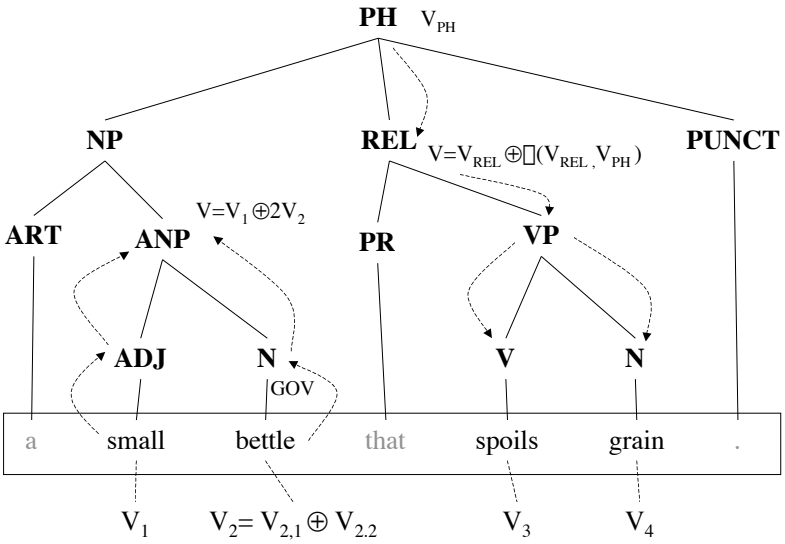


Figure 3: Simplified graphical representation of conceptual vector upward and downward propagation.

as described above. At the beginning the dictionary is empty, and a bootstrapping is done by manually indexing a small set of common words

3 Colored Ants for Word Sense Disambiguation

Ant algorithms are a class of meta-heuristics based on a population of individuals exhibiting a cooperative behavior[Langton 1987]. Ants continuously forage their territories to find food[Gordon 1995] visiting paths, creating bridges, constructing nests, etc. A fundamental principle in the emergence of coordinated system-level behavior from the local interactions of ants is stigmergy. This concept was introduced by Grassé in the 1950s from the interpretation of the behavior of social insects [Grassé 1959]. The idea of stigmergy is that a collaborative task (clustering, nest building, food search...) is implicitly coordinated through the elements (signs and/or modifications) resulting from individuals activities. For instance, ants perform indirect communications using chemical signals called *pheromones*. The larger the quantity of pheromones on a path, the larger the number of ants visiting this path. Thus, signs left on paths by some ants influence choices of next ants. This characteristic was successfully exploited for processing various combinatorial optimization problems like TSP or routing in networks [Dorigo and Gambardella 1997, Di Caro and Dorigo 1998]. However, two gen-

eral forms of stigmergy are identified. The first is sign-based stigmergy. The elements left by ants in the environment don't directly contribute to the achievement of the collaborative task. Pheromones fall into this category. The second form is sematectonic stigmergy. It generally involves a change in the physical characteristics of the environment. Elements of sematectonic stigmergy may be environmental modifications that directly concern the collaborative task. The application of clustering described in [Lumer and Faieta 1994] is based on that kind of stigmergy. Our method for WSD relies on both kind of stigmergy. Sign-based stigmergy plays a role in ant behaviors. Sematectonic stigmergy is used for modifying nodes characteristics and for creating new paths between vertices. In the sequel, these new paths will be called bridges.

In its simplest version, an ant algorithm follows three simple rules. In order to simplify, let us consider that the algorithm operates on a graph: (1) each ant drops a small quantity of pheromone along the path it uses; (2) at a crossroads, the choice of the ant is partially determined by the quantity of pheromones on each outgoing edge: the choice is probabilistic and the larger the quantity of pheromones on one edge, the larger the probability to choose this edge; (3) pheromones evaporate with the time. During the process, some edges are more frequently visited than other ones, thus, at the graph level some paths appear resulting from the local and indirect interactions of ants.

3.1 Motivation for Colored Ants

The "binary bridge" is an experiment developed by [Pasteels *et al.*]. As reported in [Dorigo *et al.* 1999] *in this experiment, a food source is separated from the nest by a bridge with two equally long branches A and B.*Initially, both paths are visited and after some iterations, one path is selected by the ants, whereas the second, although as good as the first one, is deserted. This experiment interests us for two reasons. It first shows that ants have the ability of organizing themselves in order to determine a global solution from local interactions, thus, it is likely to obtain an emergent solution for a problem submitted to an ant-based method. This point is crucial for our problem, since we expect the emergence of a meaning for the analyzed text. But, the experiment also shows the inability of such method, in its classical formulation, to provide a set of simultaneous and distinct solutions instead of only one at a time. As these methods are based on the reinforcement of the current best solution, they are not directly suitable for our situation. Indeed, if several meanings are possible for a text, all these meanings should emerge. In this work we present an ant-based method implementing several colonies competing for promoting their meaning and collaborating for the building of global meanings. These colonies are distinguished by colors: one color is associated to each sense of each term. A similar approach was already successfully implemented for performing dynamic load balancing in the context of simulations [Bertelle *et al.* 2004]. These competing colonies provide their own solution constrained by the senses of the terms forming the text. Usually the emergence of a global solution results from local interactions, and in most cases local interactions are limited to geographic proximity. In our case, local interactions should also concern semantic proximity of words that may be very distant in the text and in the morphosyntactic tree. For that reason, in our model ants are allowed to modify their environment by building bridges between

"friends words" as it will be described in the sequel.[4] Text analysis results in computing a global conceptual vector at each level of the analysis tree. These vectors represent the *ideas* expressed at various text granularity: terms, groups, sentences, paragraphs... At the root level, we want to get the global concept activations of the entire text. These goals are extended to get some explicit interpretation representation, in the form of transversal trails between selected word senses. In the French sentence *L'avocat est véreux.* (see figure 4) we have only two reasonable interpretations (out of four). Generally, not all combinations of word senses are possible and the Standard Propagation Analysis only accounts for conceptual activation not *interpretation trails*.

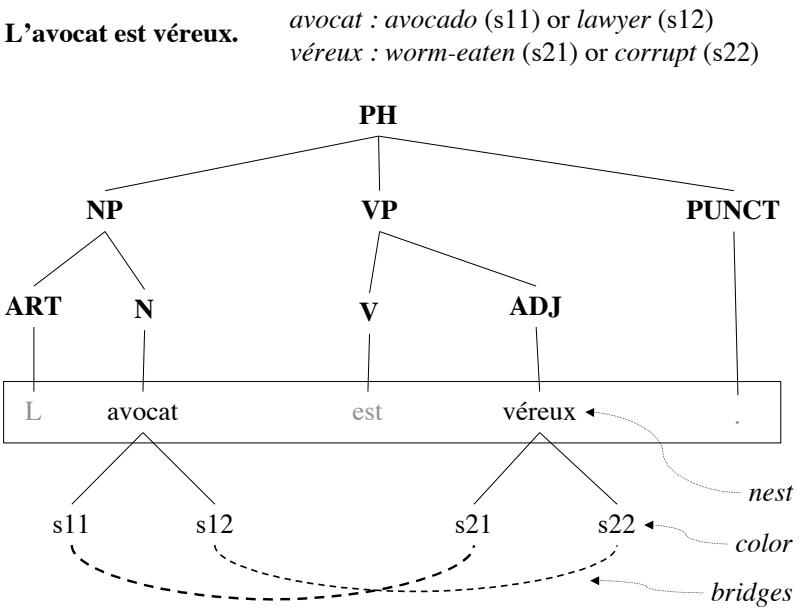


Figure 4: For the french sentence *L'avocat est véreux.* There are two interpretations (*The avocado is worm-eaten* or *The lawyer is corrupt.* An interpretation trails is a strongly excited bridge (or bridge sequence) between activated word senses.

3.2 Environment

As before, the underlying structure of the environment is the morphosyntactic analysis tree of the text to be analyzed. Each content word is a node. This node has as many children in the tree as senses. To each child associated to a sense corresponds a unique color: the conceptual vector of the sense. A child is also a *nest* and all children of a node associated to a content word are *competing nests*. In figure 4, both nodes are

[4] friend senses ? Senses which conceptual vectors are close in term of angular distance will be called friends in the sequel

directly linked to two nests. An ant can walk through graph edges and, under some circumstances, can build new ones (called bridges). Each node contains the following attributes beside the morphosyntactic information computed by the analyzer: (1) a resource level R, and (2) a conceptual vector V. Each edge contains (1) a pheromone level. The main purpose of pheromone is to evaluate how popular a given edge is. The environment by itself is evolving in various aspects:

1. the conceptual vector of a node is slightly modified each time a new ant arrives. Only vectors of nests are invariant (they cannot be modified). A nest node is initialized with the conceptual vector of its word sense, other nodes with the null vector.

2. resources tend to be redistributed toward and between nests which *reinvest* them in ants production. Nodes have an initial amount of resources of 1.

3. the pheromone level of edges are modified by ant moves. There is a factor decay δ (the evaporation factor) which ensures that with time pheromone level tends to decrease toward zero if no ant are passing through. Only bridges (edges created by ants) would disappear if their pheromone level reaches zero.

The environment has an impact on an ant and in return ants continuously modify the environment. The results of a simulation run are decoded thanks to the pheromone level of bridges and the resource level of nests.

3.3 Nests, Ant Life and Death

A nest (word sense) has some resources which are used for producing new ants. The level of resources denoted R $\in [-\infty, +\infty]$ may be negative. However, a nest with a negative level of resources may still produce new ants. At each cycle, among the set of nests having the same parent node (content word), only one is allowed to produced a new ant. The color of this ant is the one of the selected nest. In all generality, a content word has n children (nests), and the nest chosen for producing the next ant is probalistically selected according to the level of resources.

There is a cost ϵ for producing an ant, which is deducted from the nest resources. Resource levels of nests are modified by ants.

The probability of producing an ant, is related to a sigmoid function (see figure 6) applied to the resource level of the nest. The definition of this function ensures that a nest has always the possibility to produce a new ant although the chances are low when the node is inhibited (resources below zero). A nest can still borrow resources and thus a word meaning has still a chance to express itself even if the environment is very unfriendly.

The ant cost can be related to the ant life span λ which is the number of cycles the ant can forage before dying. Below, we discuss the effect of setting a high or low λ on the overall process. When an ant dies, it gives back all the resources it carries plus its cost, to the currently visited node. For instance, if the ant carrying 0.5 dies on node N, the new level of resources of this node is increased by $\epsilon + 0.5$. This approach leads to a very important property of the system, that the total level of resources is constant.

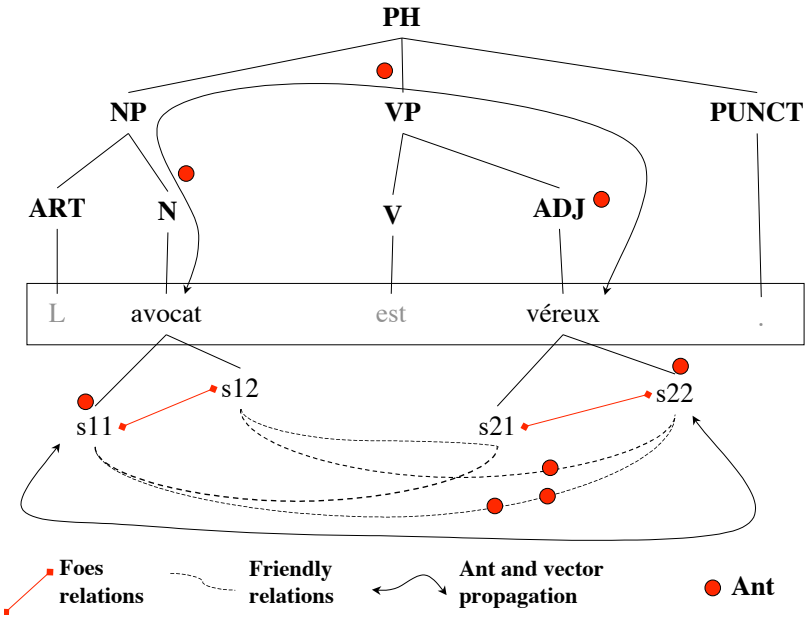


Figure 5: Propagation schema. Ants by their foraging activities establish Friend and Foes relations between nests and are redistributing resources and propagating conceptual vectors among the nodes of the tree. The tree is becoming a graph through the creation of new relations (bridges). Mostly activated relations lead to interpretation trails.

The resources can be unevenly distributed among nodes and ants and this distribution changes over time, sometimes leading to some stabilization and sometimes leading to periodic configurations. This is this *transfer of resources* that reflects the lexical selection, through word senses activation and inhibition.

The ant population (precisely the color distribution) is then evolving in a different way of classical approaches ([Dorigo and Gambardella 1997]) where ants are all similar and their number fixed in advance. However, at any time (greater than λ), the environment contains at most λ ants that have been produced by the nests of a given content word. It means that the global ant population size depends on the number of content words of the text to be analyzed, but not on the number of word meanings. To our views, this is a very strong point that reflects the fact some meanings will express more than others, and that, for a very polysemic word, the ant struggle will be intense. A monosemic word will often serve as a pivot to other meanings. Moreover, this characteristic allows us to evaluate the computing requirements needed for computing the analysis of a given text since the number of ants depends only on the number of words.

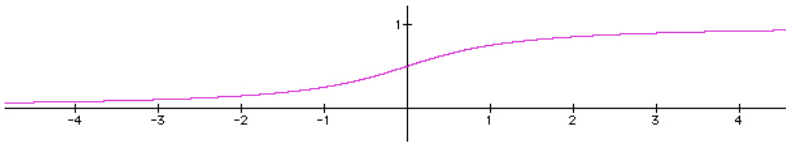


Figure 6: Sigmoid function: $\text{Sig}(x) = \frac{1}{\pi}\arctan(x) + 0.5$. Some values are: $\text{Sig}(0) = 1/2, \text{Sig}(1) = 0.75, \text{Sig}(2) = 0.852, \text{Sig}(-1) = 0.25, \text{Sig}(-2) = 0.147$.

3.4 Ant Population

An ant has only one motivation: foraging and bringing back resources to its nest. To this purpose, an ant has two kinds of behavior (called modes), (1) searching and foraging and (2) returning resources back to the nest. An ant a has a resource storage capacity $\text{R}(a) \in [0, 1]$. At each cycle, the ant will decide between both modes as a linear function of its storage. For example, if the $\text{R}(a) = 0.75$, there is a 75% chance that this ant a is in *bringing back* mode.

Each time an ant visits a (non-nest) node, it modifies the node color by adding a small amount of its own color. This modification of the environment is one factor of the sematectonic stigmergy previously mentioned and is the means for an ant to find its way back home. The new value of the color is computed as follows: $C(N) = C(N) + \alpha C(a)$ with $0 < \alpha < 1$. In our application, colors are conceptual vectors and the "+" operation is a normalized vector addition ($V(N) = V(N) \oplus \alpha V(a)$). We found heuristically, that $\alpha = 1/\lambda$ constitutes a good trade-off between a static and a versatile environment.

3.5 Searching Behavior

Given a node N_i. N_j is a neighbor of N_i if and only if there exists an edge E_{ij} linking both nodes. A node N_i is characterized by a resource level noted as $\text{R}(N_i)$. An edge E_{ij} is characterized by a pheromone level noted as $\text{Ph}(E_{ij})$. A searching ant will move according to the resource level of each neighboring node (its own nest excepted) and to the level of pheromones of the outgoing edges. More precisely an attraction value is computed for each neighbor. This value is proportional to the resource level and inversely proportional to the pheromone level:

$$\text{attract}_S(N_x) = \frac{\max\left(\text{R}(N_x), \eta\right)}{\text{Ph}(E_{ix}) + 1} \tag{7}$$

Where η is a tiny constant avoiding null values for attraction. The motivation for considering an attraction value proportional to the inverse of the pheromone level is to

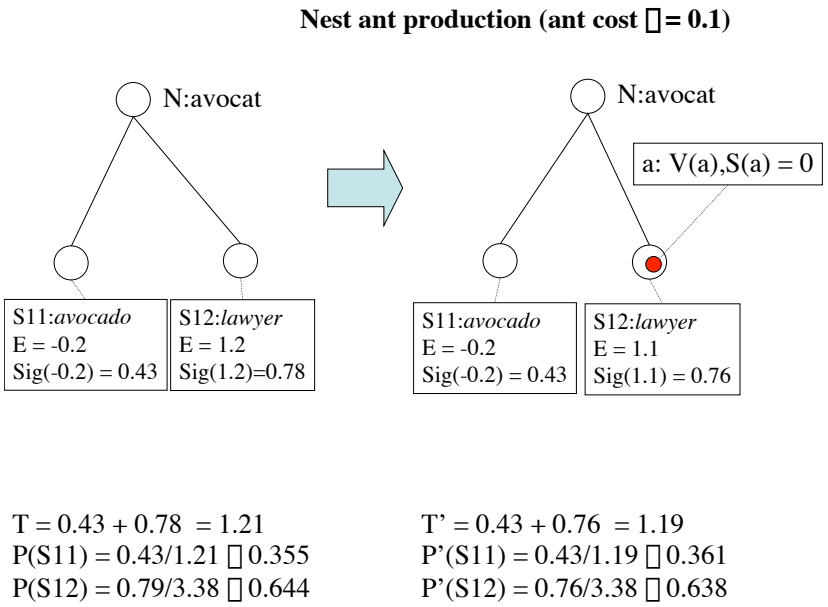


Figure 7: Example of nest ant production. Nest *S12* has been selected to produce a new ant, thus decreasing its total amount of resources by ϵ. For the next cycle, the respective probability of ant production between *s11* and *S12* have been adjusted.

encourage ants to move to non visited parts of the graph. If an ant is at node N_i with p neighbors $N_k(k = 1 \cdots p)$, the probability $P_S(N_x)$ for this ant to choose node N_x in *searching* mode is:

$$P_S(N_x) = \frac{\text{attract}_S(N_x)}{\sum_{1 \leq j \leq p} \text{attract}_S(N_j)} \tag{8}$$

Then, if all neighbors of a node N_i have the same level of resources (including zero), then the probability for an ant visiting N_i to move to a neighbor N_x depends only on the pheromone level of the edge E_{ix}.

An ant is attracted by node with a large supply of resources, and will take as much as it can hold (possibly all node resources, see figure 8). A depleted node does not attract searching ants. The principle here, is a simple greedy algorithm.

3.6 Bringing Back Behavior

When an ant has found enough resources, it tends to bring them back to its nest. The ant will try to find its way back thanks to the color trail left back during previous moves.

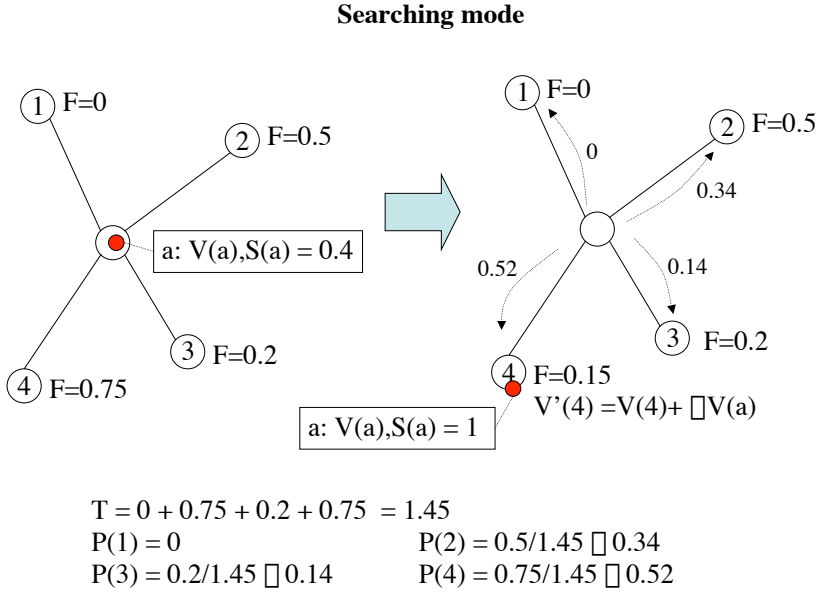


Figure 8: Example of node selection for a searching ant. The node 4 have been (randomly) chosen inducing a color propagation of the ant color.

This trail could have been reinforced by ants of the same color, or inversely blurred by ants of other colors.

An ant a returning back and visiting N_i will move according to the color similarity of each neighboring node N_x with its own color and according to the level of pheromones of the outgoing edges. More precisely an attraction value is computed for each neighbor. This value is proportional to the similarity of colors and to the pheromone level:

$$\text{attract}_R(N_x) = \max(\text{sim}(\text{colorOf}(N_x), \text{colorOf}(a)), \eta) \times (\text{Ph}(E_{ix}) + 1) \quad (9)$$

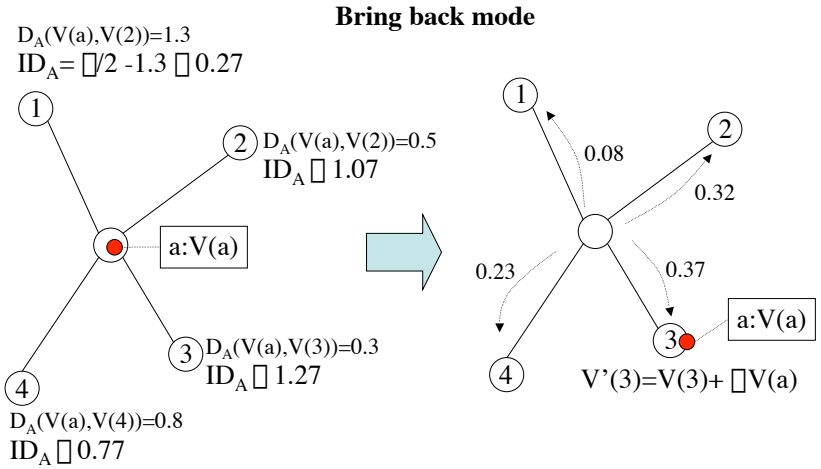
Where η is a tiny constant avoiding null values for attraction.

If an ant is at node N_i with p neighbors $N_k (k = 1 \cdots p)$, the probability $P_B(N_x)$ for this ant to choose node N_x in *returning* mode is:

$$P_R(N_x) = \frac{\text{attract}_R(N_x)}{\sum_{1 \leq j \leq p} \text{attract}_R(N_j)} \quad (10)$$

In our case where colors are represented through conceptual vectors, the similarity function is called the mutual information denoted as mi and defined as follows:

$$\text{mi}(N_x, a) = 1 - \frac{2 \times D_A(\text{vectorOf}(N_x), \text{vectorOf}(a))}{\pi} \quad (11)$$



T = 0.27 + 1.07 + 0.77 + 1.27 = 3.38
P(1) = 0.27/3.38 0.08 P(2) = 1.07/3.38 0.32
P(3) = 1.27/3.38 0.37 P(4) = 0.77/3.38 0.23

Figure 9: Example of node selection for a bringing ant. The node 3 have been (randomly) chosen inducing a vector propagation of the ant vector.

All considered nodes are those connected by edges in the graph. Thus, the syntactic relations, projected into geometric neighborhood on the tree, dictate constraints on the ant possible moves. However, when an ant is at a friendly nest, it can create a shortcut (called a *bridge*) directly to its home nest. That way, the graph is modified and this new arc can be used by other ants. These arcs are evanescent and might disappear when the pheromone level becomes null.

From an ant point of view, there are two kinds of nests: friend and foe. Foe nests correspond to alternative word senses and ants stemmed from these nests are competing for resources. Friendly nests are all nests of other words. Friends can fool ants by inciting them to give resource. Foe nests instead are eligible as resource sources, that is to say an ant can steal resources from an enemy nest as soon as the resource level of the latter is positive.

3.7 Stigmergy

The stigmergy principle is expressed through color propagation and pheromone deposit, both induced by the ant wandering, and through bridge creation. If ants of a given color tend to be largely present in some part of the tree, then this color will be strongly present in nodes of that region. Ants of other colors will mitigate the colors of such nodes.

During its way back home, an ant may arrive to a friendly nest N. In such a case, the ant gives some part of the resources it carries and may build a bridge from this nest to its own nest. Created bridges constitute one manifestation of sematectonic stigmergy. These bridges play a crucial part in interpretation trails detection since these trails are materialized by a couple of nests linked by a bridge.

The part of the resources left in the friendly nest is proportional to $\mathrm{mi}(N, a)$. For instance, if a is carrying 0.5 and reaches mistakenly a node with $\mathrm{mi}(N, a) = 0.6$, then the ant will give $0.5 * 0.6 = 0.3$ and will have 0.2 resource left. We notice that, when the friendly nest is the home, all resource is given since $\mathrm{mi}(\mathrm{home}, a) = 1$.

Bridges creation can induce *catastrophic events* (in the sense given by [Thom 1972]). Indeed, once created, a bridge allows some ants to reach parts of the tree unreachable otherwise. Note that the creation of a new bridge may change dramatically ant circulation in a very short time, and may ruin all structures established so far.

4 Discussion

4.1 WSD and Interpretation Trails

Correct word senses are selected among the most activated ones. Furthermore, a trail (sequence of nests linked by bridges) between word senses should be present. In rare cases, we may have conflicting results here but this is, most of the time, very significant on the semantic structure, that is, several interpretations are possible for the sentence. Very ambiguous or even humorous sentences, lead to such conflicts and can be detected.

With such a sentence *The old musician donated his organ to the hospital*, we have a strong ambiguity with the word *organ*. Quickly (after 100 cycles in our experiments), the musical instrument interpretation is supported by *musician*, but the presence of *hospital* gives credit to the body part. Both senses of *organ* are activated, but there is not a continuous trail of interpretation for the whole sentence. But a third actor comes into play with *donate*. It doesn't interact much with other words but only slightly with *organ:body part*. After some time (around 400 cycles) a bridge between *donate* and *organ:body part* is able to maintain itself, forcing the interpretation *organ:music* to slowly steps back.

4.2 Prepositional Phrase Attachment

Without adding much to the model, our approach can solve some prepositional phrase (PP) attachment. Consider the very classical example: *He saw the girl in the park*

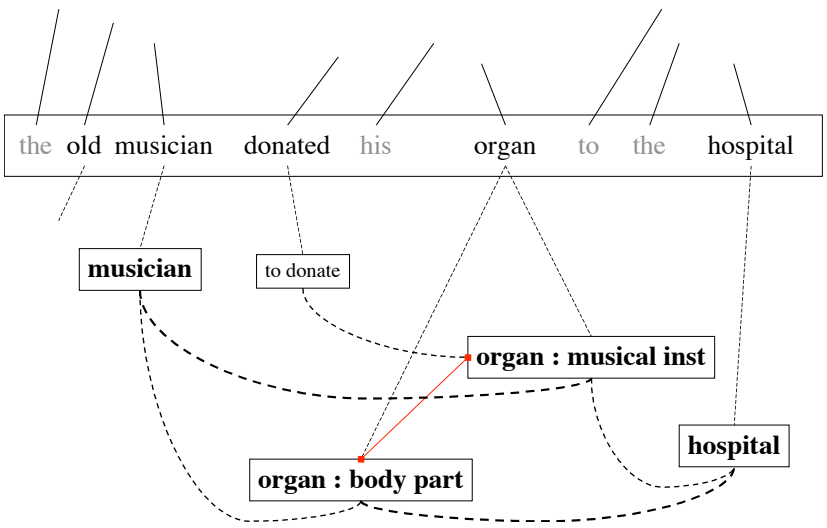


Figure 10: Example of discontinuous trail partially solved by an external weak interaction (originating from the *donate* nest).

with a telescope. First, a strong trail between *saw:see* will be created will *telescope* inducing a strong activation of this sense compared to *saw:saw* (see figure 11).

The only requirement is to enumerate all syntactically acceptable attachments for a PP. Ants and vector propagation will *semantically* choose those which maximize mutual-information sharing. In the sentence *They hit the man with a cane*, the syntagm *with a cane* will be preferably attached to *hit*. Note here, that we are not pretending to actually *solve* any ambiguity, but instead the system computes preferences. These results emerge by the mutual interaction of ants over the environment.

4.3 Results

The evaluation of our model in terms of linguistic analysis is by itself challenging. Manually examining the ant population and node activation on a given text is time consuming. To have a larger scale assessment of our system, we prefer to evaluate it through a Word Sense Disambiguation task (WSD).

A set of 100 small texts have been constituted and each term (noun, adjective, adverb and verb) has been manually tagged. A tag is a term that names one particular meaning. For example, the term *bank* could be annotated as *bank/river*, *bank/money institution* or *bank/building* assuming we restrict ourselves to three meanings. In the conceptual vector database, each word meaning is associated to at least one tag (in

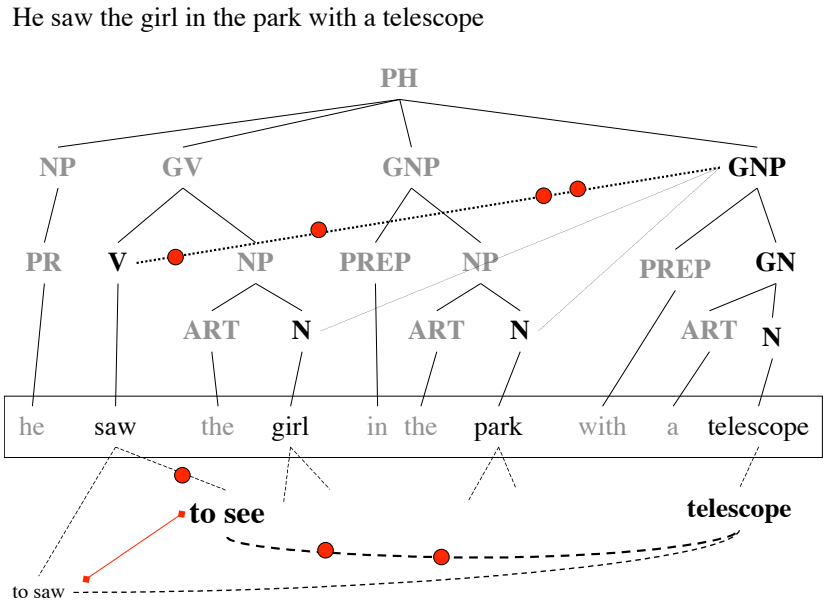


Figure 11: Example of induced PP attachment. Syntactically possible attachments of *with a telescope* are enumerated. The ant population dynamically chooses the shortest path according to conceptual vector mutual information. The strongest trail links *with a telescope* to *saw*. The process is entirely emergent.

the spirit of [Jalabert and Lafourcade 2002]). Using tag is generally much easier than sense number especially for human annotators.

The basic procedure is quite straightforward. The unannotated text is submitted to the system which annotates each term with the guessed meaning. This output is compared to the human annotated text. For a given term, the annotation available to the human annotator are those provided by the conceptual vector lexicon (i.e. for bank the human annotator should choose between *bank/river*, *bank/money institution* or *bank/building*). It is allowed for the human annotator to add several tags, in case several meanings are equally acceptable. For instance, we can have *The frigate/*{*modern ship/ancient ship*} *sunk in the harbor.*, indicating that both meanings are acceptable, but excluding *frigate/bird*. Thereafter, we call *gold standard* the annotated text. We should note that only annotated words of the gold standard are target words and used for the scoring.

When the system annotates a text, it tags the term with all meanings which activation level is above 0. That is to say that inhibited meanings are ignored. The system associates to each tag the activation level in percent. Suppose, we have in the sentence *The frigate sunk in the harbor.* an activation level of respectively 1.5 , 1 and -0.2 for respectively *frigate/modern ship*, *frigate/ancient ship* and *frigate/bird*. Then, the output produced by the system is:

The frigate/{modern ship:0.6/ancient ship:0.4}.

Precisely, we have conducted two experiments with two different ranking methods.

A *Fine Grained* approach, for which only the first best meaning proposed by the system is chosen. If the meaning is one of the gold standard tag, the answer is considered as valid and the system scores 1. Otherwise, it is considered as erroneous and the system scores 0.

A *Coarse Grained* approach, more lenient, gives room to closely related meanings. If the first meaning is the good one, then the system scores 1. Otherwise, the system scores the percent value of a good answer if present. For example, say the system mixed up completely and produced:

The frigate/{bird:0.8/ancient ship:0.2}.

the system still gets a 0.2 score.

Scoring scheme	**All terms**	**Nouns**	**Adjectives**	**Verbs**	**Adverbs**
Fine Grain Scoring	0.68	0.76	0.78	0.61	0.85
Coarse Grain Scoring	0.85	0.88	0.9	0.72	0.94

These results compare quite favorably to other WSD systems as evaluated in SENSEVAL campaign [Senseval 2000]. However, our experiment is applied to French which figures are not available in Senseval-2 [Senseval 2 2001].

As expected, verbs are the most challenging as they are highly polysemous with quite often subtle meanings. Adverbs are on the contrary quite easy to figure when polysemous.

We have manually analyzed failure cases. Typically, in most cases the information that allows a proper meaning selection are not of thematic value. Other criteria are more prevalent. Lexical functions, like hyperonymy (is-a) or meronymy (part-of) quite often play a major role. Also, meaning frequency distribution can be relevant. Very frequent meanings can have a boost compared to rare ones (for example with a proportional distribution of initial resources). Only if the context is strong, then could rare meanings emerge.

All those criteria were not modeled and included in our experiments. However, the global architecture we propose is suitable to be extended to ants of other *caste*. In the prototype we have developed so far, only one caste of ants exists, dealing with thematic information under the form of conceptual vectors. Some early assessments seem to show that only with a semantic network restricted *part-of* and *is-a* relations, a 10% gain could be expected (roughly a of gain 12% and a lost of 2%).

5 Conclusion

The conceptual vector model constitutes a numerical approach to lexical semantic representation that is applied to WSD. Contrary to traditional vector models, components

refer to ideas or concepts and not to lexical items. More specifically, the methodology includes an autonomous learning of the vectors by the system. Learning is done through the analysis of various lexical information with a strong focus on human usage dictionary definitions.

We stressed on the question of the analysis process. After a first evaluation of a vector propagation over a morphosyntactic analysis tree, it appears that such a simple strategy was falling short in many cases. Mainly, there is a need to explicitly represent connections between selected word senses, thus leading to the creation of interpretation trails. These trails are based on bridges which *solidity* is directly related on their utility for ants.

The ant approach is computationally intensive, but easily parallelizable. The main point is to maintain a (real or simulated) asynchronous parallelism. This parallelism induces partly a biased randomness, which is mandatory for building improbable bridges that may turn out to be very successful. From another spotlight, randomness and bridges may help crossing potential barriers that Standard Propagation cannot cope with. Globally, the model leads to an any-time process that is robust and adaptive. It is possible to add a new sentence next an old one, and watch the previous equilibrium shifting to a new interpretation.

We strongly believe that our approach, in its principles, is potentially very fruitful for semantic analysis. The simplified model presented here only retains the most profound aspects and some extensions have to be done in the way to a very efficient WSD. For instance, all ants are equally competent, their differences being only the color and the position of their nest. It seems clear, that several types (or castes to refer to [Bertelle *et al.* 2002]) of ants with different linguistic competencies are desirable. Potentially, other phenomena, like anaphoric relations, could then be concurrently tackled by specialized ants.

References

[Langton 1987] C. G. Langton Artificial Life. Addison Wesley.

[Gordon 1995] D. M. Gordon The expandable network of ant exploration *Animal Behaviour* 50:995-1007, 1995.

[Bertelle *et al.* 2002] C. Bertelle, A. Dutot, F. Guinand, and D. Olivier. DIMANTS: a Distributed Multi-Castes Ant System. In proceedings of Bixmas (Workshop of AAMAS 2002), pages 1-6. Bologna (Italy), July 12-14, 2002.

[Bertelle *et al.* 2004] C. Bertelle, A. Dutot, F. Guinand, D. Olivier Colored ants for distributed simulations. In *ANTS 2004*, **LNCS** vol. 3172, pages 326-333. Brussels (Belgium), September 5-8, 2004.

[Bruten *et al.* 1996] J. Bruten R. Schoonderwoerd, O. Holland and L. Rothkrantz. Ant-based load balancing in telecommunications networks. *Adaptative behavior*, 5:169–207, 1996.

[Di Caro and Dorigo 1998] G. Di Caro and M. Dorigo. AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9:317–365, 1998.

[Costa and Hertz 1997] D. Costa and A. Hertz. Ants can color graphs. *Journal of Operation Research Society*, 48:105–128, 1997.

[Chauché 1990] Chauché J., "Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance", *TA Information*, vol. 31, n° 1, p. 17-24.

[Deerwester *et al.* 1990] Deerwester S., S. Dumais , T. Landauer, G., R. Furnas, Harshman, "Indexing by latent semantic analysis", *Journal of the American Society of Information Science*, 416(6), p. 391-407, 1990.

[Dorigo and Gambardella 1997] Dorigo M., and L. Gambardella, "Ant colony system : A cooperative learning approach to the travelling salesman problem.", *IEEE Transactions on Evolutionary Computation*, 1(1), p. 114-128, 1997.

[Dorigo *et al.* 1999] M. Dorigo E. Bonabeau and G. Theraulaz. *Swarm Intelligence: from natural to artificial systems*. Oxford University Press Inc., 1999. ISBN: 0-19-513158-4.

[Gale 1992] Gale W., K. W. Church, and D. Yarowsky, "A Method for Disambiguating Word Senses in a Large Corpus.", Computers and the Humanities, 26:415-439, 1992.

[Gras *et al.* 2002] R. Gras P. Hernandez and R. D. Appel. Ant colony optimization metaheuristic applied to automated protein identification from tandem mass spectrometry data. In *Proceedings of NETTAB 2002 (Network Tools and Applications in Biology*, Bologna, Italy, July 12th - 14th 2002. http://www.nettab.org.

[Grassé 1959] P.-P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez Belicositermes natalensis et Cubitermes S.P. La théorie de la Stigmergie : essai d'interprétation du comportement des termites constructeurs. In Insectes Sociaux, vol. 6, pp. 41-80. 1959.

[Hofstadter 1995] Hofstadter, D. R., "Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought (together with the Fluid Analogies Research Group)", NY: Basic Books, 1995.

[Lafourcade *et al.* 2002] Lafourcade M., V. Prince and D. Schwab, "Vecteurs conceptuels et structuration émergente de terminologie", *TAL*, vol 43 - n° 1, p. 43-72, 2002.

[Lafourcade 2001] Lafourcade M., "Lexical sorting and lexical transfer by conceptual vectors", *First International Workshop on MultiMedia Annotation* (MMA'2001), Tokyo, 6 p, January 2001.

[Larousse 1992] Larousse, *Thésaurus Larousse - des idées aux mots - des mots aux idées.* Larousse, 1992, ISBN 1264-4242.

[Langton 1996] C. G. Langton, editor. *Artificial Life: an overview*. MIT, 2nd edition, 1996. ISBN:0-262-12189-1.

[Lumer and Faieta 1994] E. Lumer and B. Faieta, Diversity and Adaptation in Populations of Clustering Ants. Proceedings of the Conference on Simulation of Adaptive Behaviour: from animals to animats 3, pp. 501-508, MIT Press Cambridge, 1994.

[Pasteels *et al.*] J.M. Pasteels J.L. Deneubourg, S. Goss, D. Fresneau, and J.P. Lachaud. Self-organization mechanisms in ant societies (ii): learning in foraging and division of labour. *Experientia Supplementa*, 54:177–196, 1987.

[Rodget 1852] *Thesaurus of English Words and Phrases.* Longman, London, 1852.

[Salton and MacGill 1983] Salton G., MacGill M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

[Thom 1972] Thom R., *Stabilité structurelle et Morphogénèse*, InterEditions, Paris, 1972.

[Senseval 2000] http://www.itri.brighton.ac.uk/events/senseval/

[Senseval 2 2001] http://www.sle.sharp.co.uk/senseval2/

[Jalabert and Lafourcade 2002] From sense naming to vocabulary augmentation in Papillon. In proc. of PAPILLON-2003, Sapporo, Japan, July 2002, 12 p.

[Bak 1996] P. Bak, *How Nature works: the science of self-organized criticality*, Springer-Verlag, 1996.

[Wolfram 1984] S. Wolfram, "Universality and complexity in cellular automata," *Physica D*, vol. 10, pp. 91–125, 1984.

[Evry Spring School 2004] P. Amar, J.-P. Cornet, F. Képès, and V. Norris, Eds., *Proceedings of the Evry Spring School on "Modelling and Simulation of Biological Processes in the Context of Genomics"*, 2004.

[Bossomaier and Green 2000] , T. R. J. Bossomaier, D. G. Green editors, "Complex Systems", Cambridge University Press, 2000.

CHAPITRE 5

Applications et perspectives

*Ce chapitre présente quelques applications directes de ce qui précède et s'ouvre sur des perspectives de recherche. Certaines d'entre-elles ont déjà fait l'objet de travaux, mais ont été, pour le moment, peu ou pas publiées. En particulier, les techniques liées aux réseaux lexicaux et aux vecteurs d'idées ont été appliquées à l'*ingénierie des modèles *avec les travaux de Jean-Rémy Falleri. La question de l'évaluation du réseau lexical produit via JeuxDeMots est également abordée par l'entremise d'un outil de recherche de mot sur le bout de la langue. En effet, évaluer une telle ressource reste difficile d'une part, face à l'impossibilité de trouver des données de référence et de même nature, et d'autre part, face à la complexité à mettre en œuvre des évaluations en vraie grandeur (*a priori *sans échantillonnage ni des entrées ni des relations). La question de la consolidation du réseau est reprise (suite à PtiClic) en vue de l'identification d'autres modèles de jeux ou d'activités contributives. Enfin, un retour sur l'analyse sémantique est faite avec la présentation d'un modèle holistique généralisant les algorithmes bioinspirés que nous avons déjà abordés.*

Articles joints

Falleri J.-R., Prince V., Lafourcade M., Dao M., Huchard M., Nebut C. *Using Natural Language to Improve the Generation of Model Transformation in Software Design* International Multi-Conference on Computer Science and Information Technology, Pologne (2009), 8 p.

Joubert A., Lafourcade M., Schwab D., Zock M. *Évaluation et consolidation d'un réseau lexical via un outil pour retrouver le mot sur le bout de la langue* TALN'2011, Montpellier (2011), 12 p.

G. Artignan, M. Hascoët, M. Lafourcade *Multiscale Visual Analysis of Lexical Networks.* Proc. of 13th International Conference Information Visualisation IV 09, (2009), 12 p.

Encadrement - Jean-Rémy Falleri (co-encadrement Doctorat), *plusieurs groupes de TER M1 informatique ainsi que L2 et L3 informatique.*

5.1 Vers une analyse en ingénierie des modèles

Le Génie Logiciel et plus particulièrement l'*ingénierie des modèles* peut être l'objet d'une application des connaissances lexicales et ontologiques ([Falleri *et al.*, 2010], [Falleri *et al.*, 2009] inclus). Ces travaux ont eu comme prémisse la question de la discrimination entre les relations onto-

logiques, et en particulier l'hyperonymie et la méronymie [Lafourcade, 2003a]. Ces deux relations sont constitutives des objets manipulés en modélisation, et être capable de les identifier correctement dans des spécifications textuelles est un prérequis à la génération automatique de modèles. D'une façon générale, les structures syntaxiques de surface pouvant les représenter sont particulièrement ambigües, et une connaissance du monde (partielle) peut servir de base à des processus d'induction permettant une sélection appropriée.

Les travaux menés avec J.R. Falleri ont été l'occasion d'évaluer la faisabilité de l'utilisation d'un réseau lexical de connaissances générales en vue de la factorisation de classes. L'approche exploite la forme des identificateurs (similarité entre chaînes), leur statut (noms de classes, d'attributs etc.) et leur similarité thématique. La *ressemblance* entre modèles est une valeur calculée par composition des trois critères ci-dessus, et elle peut être fondée sur le réseau implicite reliant les différents objets d'un modèle. Le réseau s'enrichit incrémentalement dans le temps selon deux échelles. En premier lieu, le réseau évoluera selon une échelle locale lors du traitement (les éléments déduits sont mémorisés), et ensuite, selon une échelle globale, la mémorisation des objets d'un modèle pouvant servir ultérieurement à l'analyse ou à la factorisation de modèles ultérieurs.

Cette approche peut être comparée à celle de l'analyse sémantique de texte fondée sur des graphes, à la différence qu'ici les objets du graphe sont les objets de modèles, et non pas des termes ou des concepts du texte (comme, par exemple, pour nos travaux sur le projet UNL [Lafourcade, 2003b], [Lafourcade, 2003b] et [Lafourcade & Boitet, 2002]). Les objets disposant d'une similarité suffisante sont fusionnés au sein du réseau.

> Se reporter en particulier à l'article inclus : Falleri J.-R., Prince V., Lafourcade M., Dao M., Huchard M., Nebut C. *Using Natural Language to Improve the Generation of Model Transformation in Software Design* International Multi-Conference on Computer Science and Information Technology, Pologne (2009), 8 p.

5.2 Évaluation et consolidation d'un réseau lexical

Comment évaluer qualitativement un réseau lexical de la nature de celui produit avec JeuxDeMots ? Depuis le début du projet JeuxDeMots, s'est posée la question de l'évaluation de la ressource produite. Les évaluations que nous avons proposées jusque-là concernaient la qualité des vecteurs au travers de ressemblance aux réponses des joueurs (chapitre 2). Apprécier si la ressource est de qualité adéquate peut être envisagé à grande échelle, c'est-à-dire sur un grand nombre d'entrées, dans la durée, et de façon multiple pour chaque relation et terme. L'approche proposée ici est une alternative crédible à celle de la comparaison à une ressource de référence, qui, si elle existait, invaliderait le besoin d'en construire une nouvelle.

5.2.1 AKI : un oracle lexical

Un outil permettant de retrouver un terme qu'une personne a *sur le bout de la langue (tip of the tongue problem)* est intéressant en soi en tant qu'*assistant lexical*, mais peut également constituer une approche pour l'évaluation de données lexicales. En particulier, nous avons mis au point un tel prototype (nommé AKI [1]) dans le but d'évaluer le réseau lexical obtenu par JeuxDeMots.

Le principe de fonctionnement est le suivant. L'utilisateur entre un terme indice et AKI fait une proposition. Si la proposition est le terme recherché, l'utilisateur peut l'indiquer au système, sinon il peut introduire un nouvel indice. Le processus indice-réponse se poursuit jusqu'à deux issues possibles : AKI trouve le terme *(◇ trouvé)* ou abandonne *(× échec)* et demande la solution *(▷ solution)*. AKI peut être vu comme un outil ou comme un jeu (où l'utilisateur essayerait de mettre en défaut sa

1. http://jeuxdemots.org/AKI.php

capacité à trouver les termes à partir d'indices).

Par exemple, nous avons obtenu les parties suivantes :
Notation sous la forme *(indice joueur)* ⇒ *(proposition d'*AKI*)*

Exemple 1
cheval ⇒ équitation
pied ⇒ patte
chaussure ⇒ sabot
♢ *(trouvé)*.

Exemple 2
vin ⇒ bouteille
rouge ⇒ blanc
Bordeaux ⇒ vin rouge
cabernet sauvignon ⇒ Médoc
AOC sauvignon ⇒ médoc (vin)
♢ *(trouvé)*.

Exemple 3
plage ⇒ sable
construction ⇒ bateau
grain ⇒ caillou
pelle ⇒ lieu
tour ⇒ château
creuser ⇒ × *(échec)*
▷ château de sable *(solution donnée)*.

Exemple 4
caillou ⇒ pierre
oiseau ⇒ hibou
garçon ⇒ petit
bûcheron ⇒ Le Petit Poucet
♢ *(trouvé)*.

FIGURE 5.1 – Exemple de session sous AKI. À chaque indice entré par l'utilisateur, AKI propose une réponse.

Nous avons constaté que dans l'immense majorité des cas, l'utilisateur est de bonne foi, c'est-à-dire qu'il fournit des indices effectivement en rapport avec le terme qu'il veut faire deviner. Ceci s'explique sans doute parce que berner une machine avec des indices tels qu'aucun être humain n'aurait pu trouver la solution n'a aucun intérêt. Toutefois, afin de vérifier cela dans la suite nous proposons plusieurs expériences, où nous tentons de comparer les performances dans des tâches identiques. Lors de la partie en exemple 4, l'utilisateur a fourni des indices relativement indirects, mais qui restent pour l'essentiel corrects.

Jouer avec des relations

L'utilisateur peut, s'il le souhaite, faire précéder l'indice du nom d'une relation spécifique. Par exemple, l'entrée *:carac blanc* aura comme interprétation que le terme recherché a comme caractéristique *blanc*. Les principales relations du réseau de JeuxDeMots sont disponibles. Préciser la relation peut fortement accélérer la découverte du mot recherché, ou à l'inverse lorsque le réseau

lexical est incomplet perdre AKI. Il est possible de mélanger des indices sous forme générale (la relation n'est pas spécifiée) et des indices sous formes précises. Nous avons, par exemple :

Exemple 5
:part dent ⇒ bouche
:loc rivière ⇒ piranha
:make barrage ⇒ castor
♢ *(trouvé)*.

Exemple 6
:tout plat ⇒ lentille
:carac blanc ⇒ oeuf
:part peau ⇒ lapin
:do faire pleurer ⇒ oignon (plante potagère)
♢ *(trouvé)*.

Enfin, le joueur peut proposer un terme raffiné comme indice, en faisant suivre l'indice de $>$ *glose*. Comme a priori, le joueur ne sait pas forcément quelle serait la bonne glose, il peut choisir pour cela le terme qu'il souhaite, le raffinement sera sélectionné en fonction de sa similarité au terme choisi. Par exemple, pour la proposition *poisson>bête* (qui n'existe pas sous cette forme dans le réseau), le système considère qu'il s'agit de *poisson>animal*. L'utilisateur peut détruire ses indices, s'il juge qu'ils ne sont pas bien interprétés, et en remettre d'autres.

Exemple 7
:isa plat (spécialité) ⇒ couscous (plat)
:part saucisse (charcuterie) ⇒ choucroute garnie
:loc Sud-Ouest ⇒ cassoulet
♢ *(trouvé)*.

Exemple 8
:isa plante (botanique) ⇒ lierre
:part tubercule ⇒ pomme de terre
:syn pomme de terre ⇒ solanum tuberosum
:hypo charlotte (pomme de terre) ⇒ patate
♢ *(trouvé)*.

Principe de l'algorithme

À partir du premier indice i_1 une signature lexicale $S(i_1) = S_1 = t_1, t_2, \cdots$ est calculée, où les t_i sont triés par activations décroissantes. Autrement dit, t_1 est le terme le plus activé de la signature. D'une façon générale, nous noterons ce terme $max(S)$. De cette signature, i_1 est retiré (en effet s'il est présent, renvoyer l'indice fourni ne serait pas pertinent). Ce terme t_1 est proposé comme première réponse R_1 au premier indice, et est retiré de la signature. Donc à ce stade, la signature courante est :

$$S'_1 = S_1 \setminus t_1 \setminus i_1$$

À partir du second indice t_2, une intersection entre la signature courante et celle de second indice est réalisée. D'une façon générale, à l'étape n, nous avons :

$$S_n = (S'_{n-1} \cap S(i_n)) \setminus i_n \quad \text{et} \quad S'_n = S_n \setminus max(S_n)$$
$$R_n = max(S_n)$$

Au fur est à mesure de l'insertion d'indices, la signature diminue et finit par devenir vide (sauf si le terme cherché a été trouvé entre temps, auquel cas le processus s'arrête). Une fois la signature vide, AKI n'est plus en mesure de proposer quoi que ce soit. Le processus pourrait s'arrêter là, mais il s'avère qu'une procédure de rattrapage peut être utilisée avec profit. Elle consiste non plus à effectuer des intersections de signatures, mais des sommes.

$$S_n = (S'_{n-1} \oplus S(i_n)) \setminus i_n \quad \text{et} \quad S'_n = S_n \setminus max(S_n)$$

$$R_n = max(S_n)$$

Cette procédure de rattrapage doit être bornée en nombre de coups. En pratique, deux itérations de cette procédure semblent suffisantes. En effet, au-delà, les propositions du système s'éloignent considérablement de la thématique initialement majoritaire, pour aller piocher dans des termes très génériques.

Comme nous pouvons le constater, le principe général de l'algorithme est extrêmement simple. Il s'agit d'un choix délibéré afin que la procédure globale d'évaluation reste la plus proche possible des données. Ce sont bien les données que nous cherchons à évaluer, pas spécifiquement l'algorithme ci-dessus. La procédure de rattrapage a également pour fonction de faire découvrir occasionnellement au système de nouvelles relations, qui sont manuellement validées par la suite.

Résultats sur un vocabulaire fermé

Nous avons mené une évaluation informelle des performances d'AKI à partir du *jeu tabou inversé*. Le principe du jeu tabou est de faire deviner un terme à d'autre personnes à l'aide d'indices à l'exclusion d'une liste réduite de termes dits *tabou*. La version commerciale de ce jeu fournit une collection de 500 fiches comprenant un *terme cible* et cinq termes ne pouvant être utilisés comme indices. La version inversée de ce jeu consiste à faire deviner le mot cible en énumérant ces termes tabous. L'hypothèse est que ces indices sont un ensemble particulièrement évocateurs du terme à trouver. Afin de comparer les performance du système et d'un utilisateur, nous avons donc soumis cette collection de 500 fiches à AKI et à trois personnes. AKI a retrouvé le terme cible au plus tard au bout des 5 indices dans 494 cas (soit 98,8% de réussite). Les personnes prises comme références ont globalement trouvé (dans les mêmes conditions) 402 fois (soit 80,4% de réussite).

FIGURE 5.2 – AKI : exemples de fiches du jeu *Tabou*

Résultats sur un vocabulaire ouvert

Une expérience de plus grande ampleur, dont le compte est rendu précisément dans [Joubert *et al.*, 2011] (article inclus), est menée sur AKI depuis décembre 2010. Nous avons mesuré les données relatives aux parties (mots cibles, indices, réponses et résultats), afin non seulement d'en voir l'évolution, mais également d'arriver à caractériser aussi bien les comportements moyens d'AKI que celui des joueurs. La courbe [2] de la figure 5.3 représente l'évolution du taux de réussite d'AKI. Ce taux est fluctuant aux alentours de 70-75% et présente des irrégularités manifestes. Il semble y avoir des périodes où le système aurait des performances basses aux alentours de 70%, et d'autres périodes où il avoisinerait 80%. Nous ne savons pas précisément si cela est fonction de l'activité et des comportements des utilisateurs, ou s'il s'agit d'une propriété intrinsèque de ce type de système à apprentissage.

2. http ://www.lirmm.fr/jeuxdemots/AKI.php ?action=graphtailleseg=autostyle=step

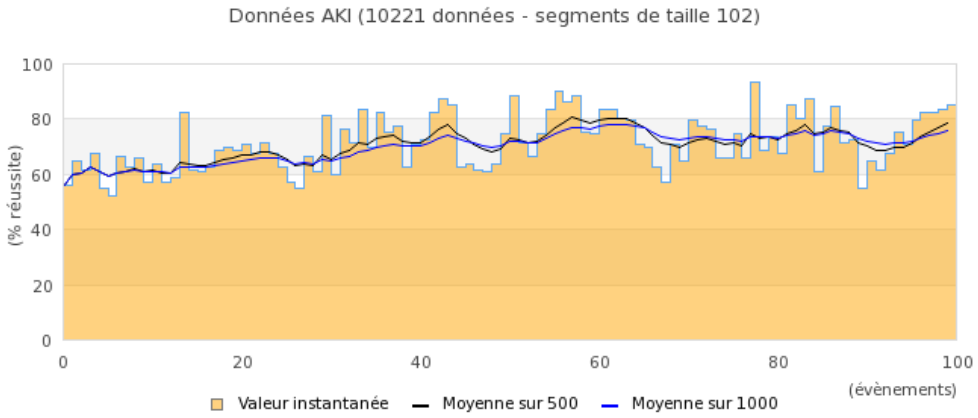


FIGURE 5.3 – AKI : graphe d'évolution du taux de réussite.

En général, les joueurs cherchent à mettre le système en défaut (pour environ 4 parties sur 5), et cela selon deux stratégies distinctes. La première consiste à faire deviner un terme rare, de spécialité ou récent (lié à l'actualité), en espérant avec raison qu'AKI ne le connaitra pas. Dans ce cas, les indices sont directs, c'est-à-dire fortement associés au terme à trouver (par exemple, *félin*, *miauler*, *souris* pour faire deviner *chat*). La seconde stratégie consiste à faire deviner un terme courant mais en donnant des indices indirects (par exemple, *canapé*, *poils*, *croquettes* pour faire deviner *chat*). Dans environ un cas sur trois, une partie porte sur un terme courant.

		partie directe	partie indirecte
Terme	courant	7%	26%
	rare	66%	$\leqslant 1\%$

AKI : distribution des types de parties et des termes

En moyenne, AKI trouve la solution correcte entre 70% et 75% des cas (sur environ 10000 parties). Il est encore difficile de dire si ce résultat est stable dans le temps.

À l'aide d'un groupe de 22 étudiants (M1 module TALN niveau 1), nous avons mené une contre-évaluation des parties de AKI proposées. Nous avons demandé à chaque étudiant de faire au moins une dizaine de parties, de noter les indices proposés et le résultat, puis ensuite de refaire la même partie cette fois avec comme partenaire une personne. L'objectif est, sur le même ensemble de parties et sur des termes librement choisis par les étudiants, de comparer les performances d'AKI et d'individus, mais également d'évaluer la difficulté d'une partie. Nous supposerons (peut-être naïvement) qu'une partie échouée pour un individu est difficile.

		AKI échec	AKI réussite	total
Humain	**échec**	55	74	129
	réussite	18	113	131
	total	73	187	260

AKI : distribution des réussites et des échecs pour AKI et les joueurs

Nous avons ainsi obtenu 260 parties. Globalement, AKI trouve dans 72% des cas (187 sur 260) alors que les joueurs, dans cette expérience, dépassent à peine 50%. Lorsque AKI échoue, les joueurs trouvent la bonne réponse dans 25% des cas (18 sur 73). Alors que lorsque AKI réussit, les joueurs humains ne réussissent que dans 60% des cas (113 sur 187). AKI réussit dans 57% des cas d'échec (74 sur 129) des personnes sollicitées. AKI réussit dans 86% des cas (113 sur 131) où les personnes réussissent. À l'inverse, dans 14% des cas, AKI échoue là où des personnes réussissent. Il s'agit dans ce cas clairement de termes trop peu renseignés ou inexistants dans le réseau lexical. La plupart du temps, ce sont des termes récents ou des termes auxquels l'actualité a donné une coloration nouvelle.

Que pouvons-nous en conclure ? Que AKI obtient des résultats honorables, certes, mais encore ? Nous pouvons sans doute considérer que la connaissance contenue dans le réseau est une somme de connaissances de plusieurs personnes (les joueurs de JeuxDeMots ou de AKI, etc.). Les individus ont une connaissance ayant une forte composante commune, mais possèdent également des savoirs moins partagés, qui eux se retrouvent dans le réseau. De ce point de vue là, AKI est un autre jeu dual de JeuxDeMots, car les joueurs vont piocher majoritairement dans des domaines spécifiques (mais qui restent de l'ordre de ce qui est partageable, cela excluant les savoirs strictement personnels). Parmi les quelques perles que nous avons relevées, nous avons :

Exemple 9
calcaire ⇒ roche
ovale ⇒ terrain
Christophe Colomb ⇒ oeuf
♢ *(trouvé)*.

Exemple 10
:carac vert ⇒ oeil
:carac poilu ⇒ pelouse
s'amouracher ⇒ coeur d'artichaut
♢ *(trouvé)*.

Échantillonnage par les utilisateurs ?

Un échantillonnage est effectivement réalisé sur les entrées du réseau lexical, mais par les utilisateurs eux-mêmes. En effet, ce sont eux qui choisissent les termes à faire deviner. Est-ce que cet échantillonnage est de bonne qualité ? Si nous considérons qu'il doit être représentatif des termes intéressant les individus, alors il est possible de répondre par l'affirmative. Ceci étant, il est vraisemblable qu'un échantillon pertinent est fortement dépendant de l'application. Par exemple, il n'y pas de raison de croire que les termes choisis par les joueurs d'AKI constituent un ensemble totalement pertinent dans le cadre de la traduction automatique.

Un tiers des termes joués relève du vocabulaire courant, vocabulaire devant être correctement renseigné dans le réseau quelles que soient les applications visées. Cette proportion est en fait bien supérieure à la quote-part du vocabulaire courant sur l'ensemble des termes. Les termes courants sont donc plutôt bien renseignés (ou testés) via AKI.

Un test sévère ?

Comment interpréter ces résultats ? Nous pourrions considérer l'ensemble des résultats obtenus par AKI comme relativement satisfaisants, l'outil dépassant les performances des utilisateurs. Cette

remarque est à nuancer dans la mesure où AKI ne procède que par croisements sur les données qu'il possède, et ne se fonde sur aucun processus déductif ou de décomposition morphologique de l'entrée (ce qu'un humain fait sans hésiter). Par exemple, à partir de l'indice *pinson de Darwin*, il sera vraisemblable que les co-indices *pinson* et *Darwin* seront activés. Avec AKI ce ne sera le cas que si les termes sont reliés dans le réseau. L'algorithme utilisé dans AKI n'est finalement fondé que sur la force brute et la quantité de données disponibles dans le réseau. Par ailleurs, nous pensons également que les conditions d'évaluation sont particulièrement sévères (ou défavorables) pour le contenu du réseau lexical. En effet, les joueurs ont des comportements relativement déviants, et visent à mettre en défaut le système, avec souvent des indices très indirects. Dans le cas de l'analyse sémantique, les termes du texte constituant le contexte sont le plus souvent fortement en relation avec les termes à désambiguïser. Une estimation empirique serait d'augmenter d'environ 10 à 15% les performances d'un contexte thématique en situation réelle d'analyse de texte. Non seulement le contexte serait plus favorable, mais de plus la sélection lexicale se ferait parmi les acceptions possibles du terme cible. Il est possible d'espérer avec une longue traîne importante, approcher les 90% de désambiguïsation lexicale (au moins pour les substantifs). Ces résultats semblent cohérents avec ceux présentés par [Navigli & Lapata, 2010] sur l'utilisation de Wikipedia comme source d'amélioration de WordNet.

5.2.2 Vers d'autres activités pour l'acquisition de données lexicales

Quelles alternatives à JeuxDeMots ? Les modèles de JeuxDeMots et de PtiClic présentent des biais et des limites qu'il est possible de dépasser en partie avec d'autres approches.

Limites du jeu associatif

Nous avons déjà présenté (au chapitre 3) certaines limites du jeu associatif pour la construction d'un réseau lexical. Nous les exposons de nouveau brièvement ici :

- jouabilité : les parties proposées doivent être jouables, c'est-à-dire en particulier faire intervenir des relations intuitives aux joueurs et raisonnablement lexicalisées (il doit y avoir un nombre de réponses suffisant pour alimenter l'aspect ludique) ;
- nature du vocabulaire : JeuxDeMots est un jeu ouvert sollicitant en grande partie le vocabulaire actif. PtiClic permet de compenser au moins partiellement ce biais, en proposant un nuage de mots issu d'un vocabulaire potentiellement plus large ;
- relation positive : l'ensemble des modèles de jeu que nous avons présentés ne permet de générer que des relations positives (dites d'*activation*), or il semblerait intéressant de pouvoir disposer également de relations négatives (dites d'*inhibition*) ;
- contributeurs : il est vraisemblable qu'il existe une population non-joueuse qui serait potentiellement intéressée de contribuer à la construction d'une ressource lexicale comme le réseau JeuxDeMots. Que serait-il possible de leur proposer ?

Devinettes et fonctions peu lexicalisées

Ce qui suit est un résumé succinct des travaux rapportés dans [Lahrizi et al., 2008] et [Veyssier et al., 2009] ainsi que d'un stage d'été effectué par L. Guizol. Un prototype de jeu de devinettes (GuessIt ![3]) a été conçu de façon à favoriser l'acquisition de relations non standard dans un réseau lexical. Le jeu est à double sens, à savoir il est aussi bien possible de poser une devinette au système (comme pour AKI) que de tenter d'en trouver une posée par le système (image 5.4). Les devinettes posées au système sont mémorisées et soumises à d'autres joueurs, dans le but d'une validation pondérée.

Jeu à questions fermées

La question de la pertinence de la consolidation du réseau lexical par des mécanismes d'inférence peut se poser. En particulier quelles sont les conditions à respecter afin de ne pas dénaturer son aspect

3. http://www.lirmm.fr/~lafourcade/guessit_project/guessit/

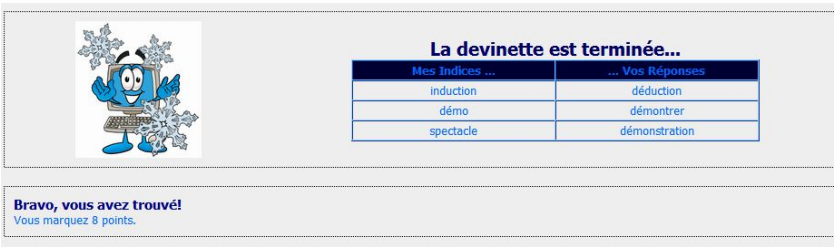


FIGURE 5.4 – GuessIt : partie type ou le joueur doit deviner ce qui est proposé par le système. Le mot à trouver était *démonstration*.

populaire ? Une fois de plus, il est possible de concevoir une activité d'inférence avec validation par les joueurs. L'invalidation par ces derniers de certaines propositions faites par le système permet également d'obtenir des relations à valeurs négatives, qui sont utiles (voire nécessaires) pour réaliser des phénomènes d'inhibition dans le cadre d'une analyse sémantique. L'inhibition permet des choix par élimination plutôt que par sélection, ou de combiner les deux. Par exemple, pour la phrase :

L'avocat mange une pomme.

La sélection du raffinement *avocat>justice* plutôt que *avocat>fruit* n'est pas triviale. En effet, si le réseau lexical contient bien (directement ou indirectement) la relation avocat>justice $\xrightarrow{agent}$ *manger* (avocat>justice est un *agent* possible de *manger*) en faveur de la bonne interprétation, l'activation thématique a tendance à favoriser le choix de la mauvaise. La présence d'une relation inhibitrice (avocat>fruit $\xrightarrow{agent:<0}$ *manger*) permet non seulement de conforter l'émergence de *avocat>justice*, mais d'accélérer l'analyse.

ASKIT est un jeu [4] à réponse *oui* - *non*, visant à renforcer le réseau lexical de JeuxDeMots, notamment avec des relations à valeur négative. Le principe du jeu est très simple (et est illustré par la figure 5.5). Une question est posée à l'utilisateur, qui a la possibilité de répondre par l'affirmative, la négative ou de s'abstenir (en cliquant sur les boutons ad-hoc, ou laisser passer la question). Un exemple de question pourrait être : "Est-ce que *fric* peut se trouver dans *banque* ?" ou encore (*) "Est-ce que voiture (automobile) a comme partie pot (chance)".

À l'issue de son choix (s'il en fait un), le joueur gagne des points, s'il a fourni une réponse conforme à la majorité des réponses fournies par les joueurs pour cette question. S'il est le premier à y répondre une heuristique fondée sur la mesure d'activation entre les termes permet de récompenser ou non le joueur. Le joueur ne sait pas combien de personnes ont déjà répondu à cette question.

Répondre négativement à une question décrémente légèrement la valeur d'une relation, dont la valeur peut être négative. Si la relation n'est pas dans le réseau, alors elle est créée (avec une valeur négative). Ainsi, il est possible d'obtenir des inhibitions entre deux termes pour chaque type de relation, qui ont une influence sur la sélection ultérieure des questions, ou des types de parties dans JeuxDeMots.

Quelle stratégie adopter pour la sélection des questions ? Il est nécessaire de définir en premier lieu le statut d'une relation négative. Il s'agit intuitivement d'une occurrence de relation où les deux termes n'entretiennent pas la relation, *et pour laquelle il est pertinent de le représenter*. Ce dernier point est important, car il ne s'agit pas d'explorer et de représenter toute la combinatoire des termes du réseau, ce qui serait non seulement fastidieux et inutile, mais probablement générateur de bruit. Dans deux cas, il semble approprié d'avoir à tester de telles relations potentiellement négatives :

– les *exceptions*, où la relation a une valeur contraire de celle qui serait obtenue par inférence (l'exemple classique est * *"autruche agent voler"*) ;

4. http ://www.lirmm.fr/jeuxdemots/askit.php

FIGURE 5.5 – Partie de ASKIT : le joueur pour répondre *oui* ou *non*, et éventuellement passer.

– les *raffinements*, où le terme général dispose d'une relation qui ne s'applique qu'à certains des usages de ce terme.

Nous avons donc défini un certain nombre de schémas d'inférence. Par exemple, les schémas déductifs sont de la forme :

$$x \xrightarrow{hyper:>0} y \quad \wedge \quad y \xrightarrow{R:>0} z \qquad \Longrightarrow \qquad x \xrightarrow{R:?} z$$

avec R $\in \{$hyper, partie, tout, obj-loc, obj-car, $\cdots\}$

Il s'agit du schéma classique où nous cherchons à *transférer* une relation du cas général vers le cas particulier. Nous testons donc l'existence de la relation d'hyperonymie entre le terme à consolider et un de ses hyperonymes ainsi qu'une relation (ontologique) entre l'hyperonyme et un troisième terme. Par exemple :

$$\begin{array}{c} \textit{lit} \xrightarrow{hyper} \textit{meuble} \quad \wedge \quad \textit{meuble} \xrightarrow{matière} \textit{bois} \\ \Longrightarrow \qquad \textit{lit} \xrightarrow{matière:?} \textit{bois} \end{array}$$

$\Longrightarrow$ Est-ce que *lit* peut être fait de *bois* ?

Le type de schéma qui suit est similaire au précédent, mais porte sur le raffinement de termes. Une relation du terme général existe-t-elle pour un de ses raffinements ?

$$x \xrightarrow{raff:>0} y \quad \wedge \quad x \xrightarrow{R:>0} z \qquad \Longrightarrow \qquad y \xrightarrow{R:?} z$$

avec R $\in \{$hyper, partie, tout, obj-loc, obj-car, $\cdots\}$

Le terme y est un raffinement de x, et x a une propriété $R(z)$. Le terme y vérifie-t-il cette propriété ?

Nous pouvons avoir par exemple :

$$\begin{array}{c} \textit{bateau} \xrightarrow{raff} \text{bateau>trottoir} \quad \wedge \quad \textit{bateau} \xrightarrow{partie} \textit{mât} \\ \Longrightarrow \qquad \text{bateau>trottoir} \xrightarrow{partie:?} \textit{mât} \end{array}$$

$\Longrightarrow$ Est-ce que bateau>trottoir a comme partie *mât* ?

Nous avons également des schémas inductifs (du particulier au général) de la forme :

$$x \xrightarrow{hyper:>0} y \quad \wedge \quad x \xrightarrow{R:>0} z \qquad \Longrightarrow \qquad y \xrightarrow{R:?} z$$

avec R $\in$ {hyper, partie, tout, obj-loc, obj-car, $\cdots$}

Parmi les relations négatives que nous avons obtenues, voici par exemple :

- * avocat>fruit $\xrightarrow{agent}$ *plaider*
- * avocat>fruit $\xrightarrow{obj-loc}$ *palais de justice*
- * avocat>justice $\xrightarrow{agent}$ *mûrir*
- * avocat>justice $\xrightarrow{partie}$ *noyau*

Nous remarquerons qu'il s'agit de situations typiques et non pas d'interdictions formelles. En effet, nous pouvons très bien imaginer que *avocat>fruit* puisse se trouver dans un *palais de justice*, cependant la situation non seulement est anecdotique mais elle sera de plus spontanément considérée comme inadéquate. Nous ne faisons pas de distinction entre l'impossibilité et l'improbabilité, et nous laissons pour l'instant cette question ouverte.

Les relations positives produites sont celles qui sont correctes par transitivité, et un mécanisme de cadeaux permet une seconde validation par les joueurs. La correction dépend évidement des réponses des joueurs, cependant la mécanique du jeu pousse à répondre honnêtement, car comme pour AKI l'activité n'offre aucune valorisation à la contourner. Par exemple, nous avons :

$$\begin{aligned} &\textit{cicindèle champêtre} \xrightarrow{hyper} \textit{coléoptère} \\ \wedge \quad &\textit{coléoptère} \xrightarrow{hyper} \textit{insecte} \\ \Longrightarrow \quad &\textit{cicindèle champêtre} \xrightarrow{hyper} \textit{insecte} \end{aligned}$$

Une partie de JeuxDeMots portant sur les hyperonymes possibles de *cicindèle champêtre* sera proposée, afin de renforcer cette relation et de la diversifier. De ce point de vue là, ASKIT est une activité permettant non seulement la consolidation du réseau, mais également la détection de relations positives potentielles.

La sélection des candidats se fait, soit si la relation n'existe pas, soit si elle a une valeur absolue faible (poids < 10). La recherche de relations à tester se fait en tâche de fond et alimente un stock de relations en cours d'évaluation. Une fois la relation suffisamment activée (positivement ou négativement), elle est retirée du stock.

Depuis environ 18 mois, ASKIT a été à l'origine de la création d'environ 15 000 relations positives et 8 000 relations négatives. Les schémas déductifs sont généralement plus productifs que les schémas inductifs. Cela peut être expliqué par la nature globalement descendante (par raffinement) de la construction du réseau lexical de JeuxDeMots. De plus, il y a bien davantage de termes, et donc de relations potentielles, sous l'horizon conceptuel [Lafourcade *et al.*, 2002b] qu'au dessus.

Vers un dictionnaire contributif

Pourquoi ne pas rendre le réseau contributif, à l'image du projet Papillon ? À quelles conditions un dictionnaire associatif et contributif pourrait-il rencontrer un certain succès (c'est-à-dire que les gens contribuent vraiment) et être efficace quant à la constitution de la ressource ?

La mise en place d'un système consultatif et contributif pour le réseau JeuxdeMots a pris la forme d'un dictionnaire d'associations : Diko [5]. L'interface présentée aux utilisateurs reste classique, et se

5. http://www.lirmm.fr/jeuxdemots/diko.php

situe dans la ligne de nos expérimentations passées (projet Fe* et serveur de dictionnaires multilingues, cf. chapitre 1). Ici, au multilinguisme, se substitue la multiplicité des relations accessibles (figure 5.6).

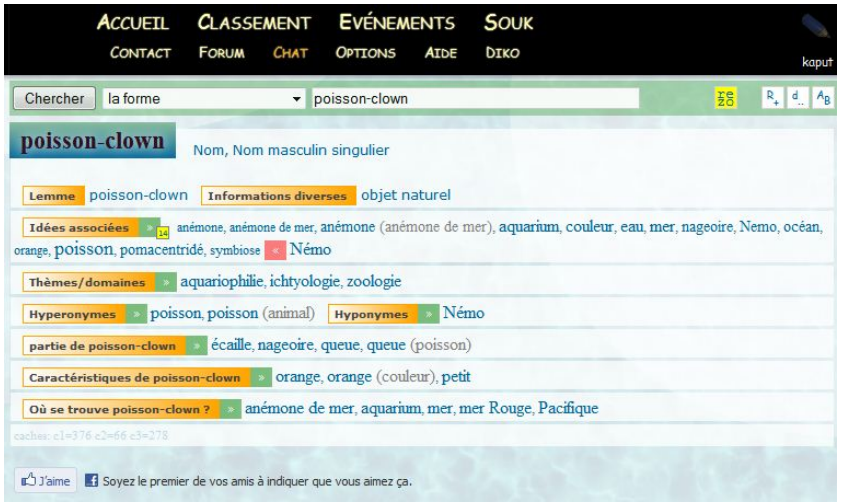


FIGURE 5.6 – Diko : affichage de l'entrée *poisson-clown* en mode consultation.

Nous souhaitons rendre ce dictionnaire contributif (dans l'esprit du projet Papillon) car certains types de relation ne sont pas adaptés à une approche ludique. En effet, elles peuvent être soit trop compliquées à jouer, soit trop peu lexicalisées. L'interface propose à un joueur (clairement identifié) une interface d'édition (figure 5.7) lui permettant de saisir pour une relation donnée les termes cibles. Ces propositions non encore validées sont soumises à l'acceptation ou refus (par vote) des autres contributeurs. La validation par un modérateur permet son inclusion effective dans le réseau, ce modérateur étant sollicité lorsque le nombre de votes pour une proposition dépasse un certain seuil.

FIGURE 5.7 – Diko : affichage de l'entrée *poisson-clown* en mode édition.

De façon similaire à JeuxDeMots, le contributeur dont le moteur est l'estime de soi, se voit confronté à deux contraintes contradictoires dont les extrêmes sont les suivants : contribuer de façon

originale au risque de ne pas être suivi et donc de ne voir jamais ses contributions validées, ou n'être que suiviste de façon à valider les contributions des autres mais se retrouver distancé dans l'affichage public des contributions de chacun.

Un système de point est mis en place afin de mesurer le niveau de contribution de chacun. Être à l'origine d'une contribution a plus de valeur (3 points) que d'être *suiviste* et de voter en faveur d'une contribution. Par contre, voter contre une contribution qui se trouve invalidée rapporte autant de points. Les contributions ne sont validées qu'à partir d'un certain nombre de votes (pour ou contre). Les points ne sont distribués qu'au moment de la validation.

La saisie d'un terme pour la recherche ou la contribution est assistée par un *système d'autocomplétion tolérant*. Suite à l'entrée de quelques caractères, un menu déroulant affiche les entrées existantes les plus probables. La tolérance du système permet à l'utilisateur une certaine marge d'erreur quant à la casse, les diacritiques, l'insertion de caractères surnuméraires ou leur suppression, ainsi qu'une mauvaise voyellisation (*ai* pour *é*, par exemple, figure 5.8a). De plus, l'autocomplétion donne accès aux raffinements du terme saisi, s'ils existent (figure 5.8b). Enfin, la liste de propositions inclut les termes sémantiquement équivalents (figure 5.8c, il s'agit, dans le réseau JeuxDeMots, d'une relation différente de la synonymie).

FIGURE 5.8 – Diko : autocomplétion tolérante.

Ce retour (boucle) vers certains aspects du projet Papillon, avec la mise en place d'un système contributif, nous permet d'identifier quelques conditions semblant nécessaires si nous souhaitons que des gens participent :

- un volume conséquent de données doit déjà être présent avant de demander de contribuer. En effet, la contribution est *sans doute* une conséquence de la consultation et ce faisant, des entrées déjà largement renseignées seront potentiellement motivantes à compléter.
- les joueurs ne vont contribuer que sur ce qui les intéresse, à savoir soit des relations non standard, soit des termes spécifiques.

L'activité contributive reste modeste dans la mesure où il s'agit encore très largement d'un prototype, et où seules des personnes sollicitées à des fins de tests ont contribué. Le nombre de relations introduites s'élève à environ 15 000, fin août 2011.

Une version alternative à Diko [6] a été développée par C. Vidal sous Facebook.

5.2.3 Visualisation globale

Quel type de visualisation globale du réseau lexical pouvons-nous envisager et dans quel but ? Idéalement, celle-ci doit permettre de découvrir les structures sous-jacentes de ce réseau qui échapperaient à l'observation locale (au niveau de chaque entrée). G. Artigant (sous la direction de M. Hascoët, [Artignan *et al.*, 2009]) a développé un mode de visualisation arborescente rappelant

6. http://apps.facebook.com/jdm_diko

l'inclusion d'ensembles. Un prototype a été développé et permet d'effectuer une plongée en profondeur avec un effet de zoom potentiellement infini (figure (5.9). Cette visualisation présuppose une classification arborescente des termes du lexique. L'arbre construit est donc dépendant de la fonction de similarité choisie.

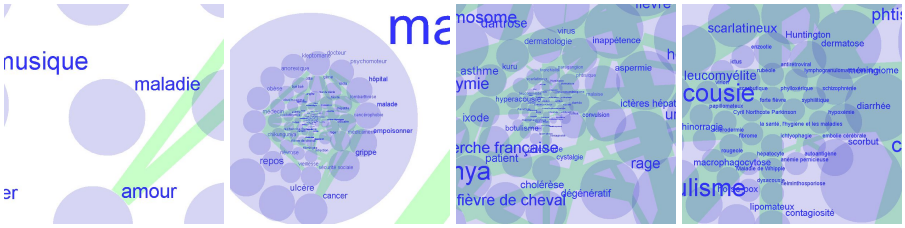


FIGURE 5.9 – Affichage global arborescent du réseau de JeuxDeMots et exploration par effet de zoom.

Par ailleurs, M. Hascoët utilise les données du réseau JeuxDeMots afin de tester des algorithmes de visualisation de graphes (figures 5.15 à 5.18, visualisation du voisinage lexical du terme *bateau*, en annexe de ce chapitre) avec l'outil Donatien [Hascoët & Dragicevic, 2011]. Les algorithmes Spring et Kamada-Kawai sont basés sur la métaphore des ressorts avec des contraintes/modèles légèrement différents. Kamada-Kawai prend explicitement en compte les distances dans le graphe pour calculer l'état optimal. Spring, plus basique, ne tient compte que de la connexité.

En plus de confronter ces algorithmes de visualisation à des données réelles de grande taille, il s'agit à nouveau de tenter de caractériser globalement le réseau de JeuxDeMots pour chaque type de relation.

5.3 Vers une analyse holistique de textes

L'analyse sémantique est clairement une analyse multicritères où l'importance relative de chaque critère est elle-même fonction des éléments du texte, de la base de connaissances et d'un ensemble de contextes. L'approche par propagation réduite à des activations peut s'avérer limitante, notamment dans de nombreux cas où la sélection des acceptions des termes peut être effectuée par élimination. Nous penserons par exemple, à une phrase comme ≪ *L'avocat grimpe dans l'arbre.* ≫, où la sélection du bon sens d'*avocat* se fait principalement par élimination. Dans ce cas, la propagation d'inhibitions semble être une possibilité intéressante pour aboutir à une opération de filtrage. Enfin, les objets manipulés au cours de l'analyse peuvent avoir un statut conceptuel (il s'agit d'idées) ou être constitutifs du texte (des syntagmes ou plus généralement des segments textuels) et se combiner afin d'établir des relations entre eux ou construire de nouveaux objets. Parmi, les relations, celles de dépendances sont à la fois fonction des constituants, de leur position syntaxique mais aussi de connaissances *a priori*.

Le modèle que nous esquissons ici, se veut aussi générique que possible et tente d'apporter des réponses opérationnelles aux points suivants :

- **suppression du contrôle** : à l'instar de nos expériences précédentes avec les algorithmes à fourmis, la construction n'est pas l'objet d'un contrôle supervisé, mais résulte d'un effet émergent ;
- **suppression des phases d'analyse** : les phases syntaxique et sémantique sont décomposées en faveur de microtâches répétitives qui se déroulent en fonction des objets présents dans la structure de calcul ;

- **approche *anytime*** : le calcul peut être suspendu à chaque instant pour une lecture de la structure qui sera la réponse du système à ce moment précis ; nous n'avons toutefois pas la garantie qu'à chaque instant la solution courante se rapproche de la solution optimale, dont la définition même n'est pas assurée ;
- **robustesse aux entrées dégradées** : nous nous situons clairement dans un processus d'analyse (et non de génération) qui cherche à créer des structures sémantiques à partir des objets présents dans l'environnement quel qu'il soit.

L'analyse sémantique, et dans une certaine mesure l'analyse syntaxique, se heurtent à la disponibilité des informations nécessaires à leur réalisation. L'approche par phases fait que souvent ces informations ne sont pas disponibles au bon moment, que certains choix sont faits ou trop tôt ou trop tard.

L'utilisation ou non des relations d'inhibition offre des possibilités non seulement d'accélération du processus d'analyse, mais d'amélioration qualitative quand le contexte n'est pas particulièrement *sélectionnant* mais plutôt *filtrant*.

La métaphore biologique des colonies de fourmis peut être conservée pour ce qui concerne la partie exploratoire du modèle. En effet, d'une part le principe de communication indirecte par modification de l'environnement est toujours utilisé, et d'autre part des stratégies diverses d'exploration du réseau se prêtent bien à une *agentification* du modèle. Les travaux de M. Minsky [Minsky, 1988] constituent une source d'inspiration, car en effet dans notre modèle les agents sont les acteurs d'une partie de la modification de l'environnent de travail. L'approche proposée par R. Hudson avec les *Word Grammars* [Hudson, 2007] où le postulat est que le langage et en particulier un énoncé est un réseau (lexical et conceptuel) est également relativement convergente avec ce que nous proposons ici. De façon plus récente, les travaux de V. Fomichov [Fomichov, 2010] ont aussi été une source d'inspiration.

5.3.1 Principe général

Le calcul vise à produire un réseau (dit de travail) à partir du texte d'entrée et d'un réseau lexical servant de base de connaissance. Dans nos expérimentations, le réseau produit par le projet JeuxDeMots a été utilisé. Nous définissons deux types de nœuds :

- les nœuds *physiques* correspondant à des segments du texte. Nous notons ce type de nœud entre crochets, par exemple [chat].
- les noeuds *conceptuels* correspondant à des idées associées aux termes du texte. Nous notons ces nœuds entre chevrons, par exemple, ⟨félin⟩.

Les nœuds disposent d'un ancrage dans l'environnement. Il s'agit en pratique d'un intervalle de mots dans le texte (en toute généralité, une liste d'intervalles, ces nœuds n'étant pas nécessairement connexes).

L'entrée pour l'analyse holistique de texte est une chaîne de termes. Par exemple, la phrase *le chat boit du lait de chèvre* prendra la forme :

$$[\top] \leftrightarrows [\text{le}] \leftrightarrows [\text{chat}] \leftrightarrows [\text{boit}] \leftrightarrows [\text{du}] \leftrightarrows [\text{lait}] \leftrightarrows [\text{de}] \leftrightarrows [\text{chèvre}] \leftrightarrows [\bot]$$

Le symbole $\leftrightarrows$ indique qu'il existe deux liens *successeur* et *prédécesseur* entre les nœuds (liens notés `s/p` par la suite, notamment dans les figures). Ce type de liens permet non seulement de représenter la succession des éléments pour les constituants, mais est également nécessaire pour rendre possible la propagation d'activations. Quand cela est nécessaire, nous noterons explicitement les liens et leur type, par exemple sous la forme : *chat* $\xrightarrow{succ}$ *boit*.

Ce que nous souhaitons obtenir en particulier comme objets et relations serait par exemple :

$[\text{du}] \leftrightarrows [\text{lait de chèvre}]$

$[GN1] \xrightarrow{struc} le\ chat$

$[du] \xrightarrow{struc} de\ le$

$[GN2] \xrightarrow{struc} [lait\ de\ chèvre]$

$[boit] \xrightarrow{lemme} [boire]$

$[GNP1] \xrightarrow{struc} [du\ GN2]$

Il s'agit là de constituants et de relations de dépendances. Des relations correspondant aux rôles sémantiques, à la désambiguïsation lexicales seraient :

$[chat] \xrightarrow{acception} \langle félin \rangle$

$[boire] \xrightarrow{agent} [chat]$

$[boire] \xrightarrow{patient} [lait\ de\ chèvre]$

$[lait\ de\ chèvre] \xrightarrow{isa} \langle produit\ laitier \rangle$

$[boire] \xrightarrow{syn} \langle laper \rangle$

Avec éventuellement quelques relations supplémentaires d'ordre explicatif :

$[boire] \xrightarrow{instr} \langle langue \rangle$

$[chat] \xrightarrow{carac} \langle assoiffé \rangle$

$\langle assoiffé \rangle \xrightarrow{cause} [boire]$

Cycle de calcul

L'analyse se fait selon une itération (potentiellement non finie) de cycles. Un cycle de calcul consiste en la réalisation des tâches suivantes :

- augmentation de l'activation des nœuds actifs ;
- production d'agents par les nœuds ;
- modification du réseau de calcul par exploration et recopie du réseau lexical ;
- nettoyage du réseau de calcul (destruction des arcs inactifs et des nœuds isolés, et fusion de nœuds conceptuels voisins) ;

Les nœuds actifs sont les nœuds correspondants aux termes du texte à analyser. Ce sont les seuls nœuds dont l'activation est augmentée d'un quantum q_{n+} d'énergie à chaque cycle. De façon identique à [Gui2010] un nœud a une probabilité covariante à son niveau d'activation de produire un agent (il s'agit encore une fois de la fonction sigmoïde, voir annexe du chapitre 4). Il existe plusieurs types d'agent ayant chacun un comportement particulier. La sélection du type d'agent lors de sa création par un nœud est faite aléatoirement. La création d'un agent coûte au nœud un quantum d'énergie q_{n-} (c'est-à-dire que son activation diminue de cette quantité). Quand un agent disparait, il restitue au nœud *où il se trouve* ce quantum q_{n-}. Ce nœud n'est pas nécessairement celui qui a créé l'agent, la disparition de l'agent en cas d'échec réalisant ainsi une forme de redistribution de l'énergie.

Vie et mort des arcs et des nœuds

Lors du déplacement d'un agent, un dépôt d'un signal est effectué sur l'arc emprunté (qui existait auparavant ou non) d'un certain quantum q_{a+}. Ce signal est métaphoriquement une phéromone dont l'intensité diminue mécaniquement à chaque cycle (d'un certain quantum q_{a-}). Si un arc voit son niveau de phéromone à zéro à la fin d'un cycle, cet arc disparait. Un nœud qui n'a plus d'arc (ni entrant ni sortant) disparait. Si le paramétrage du système est tel que $q_{a+} = q_{a-}$ alors un arc se maintient si en moyenne un agent l'emprunte à chaque cycle.

Nous faisons remarquer que ce mécanisme de disparition des nœuds isolés induit une non conservation de l'énergie totale dans le système. Sans rentrer dans le détail, les rapports qu'entretiennent les deux paires de paramètres q_{n+} (quantum d'énergie ajouté à chaque tour pour les nœuds actifs) et q_{n-} (coût d'un agent) d'une part, et q_{a+} (dépôt sur un arc) et q_{a-} (évaporation des arcs) d'autre part sont critiques quand à la dynamique du système.

Construction du réseau et agents explorateurs

La construction du réseau se fait par l'entremise de l'activité d'agents créés par les nœuds. Plusieurs types d'agents correspondent chacun à un mode de construction particulier :

agent d'exploration ;

agent de consolidation ;

agent d'agrégation ;

agent d'inférence ;

Le comportement des agents dépend de leur type, toutefois ils ont les propriétés suivantes en commun. Un agent coûte de l'énergie au nœud qui le produit. Un agent, à chaque cycle se déplace au plus d'un arc. La durée de vie des agents est variable selon leur type, toutefois à chaque cycle il y a un petit pourcentage de chance qu'un agent meure (fixé empiriquement à 5% dans nos expérimentations). Donc, aucun agent n'est en mesure de durer très longtemps quel que soit son type.

La construction du réseau est faite en premier lieu par recopies partielles du réseau lexical vers l'environnement de calcul. Cette tâche est dévolue aux *agents explorateurs* qui parcourent l'espace de travail et le réseau lexical comme s'il s'agissait d'une même entité.

À partir d'un nœud de l'environnement, un agent explorateur choisit de façon pseudo-aléatoire un nœud de destination parmi ceux pouvant être atteints dans le réseau lexical ou l'environnement courant. Si l'arc choisi appartient au réseau lexical mais n'existe pas dans l'environnement de travail, il est alors créé (avec un signal fixé à q_{a+}). Si le nœud destination n'existe pas il est également créé avec un niveau d'énergie de 0. Un agent explorateur meurt une fois le nœud destination atteint (il ne se déplace donc qu'une seule fois). En définitive, le nœud nouvellement créé dispose d'un niveau d'énergie (ou activation) à $0 + q_{n+} = q_{n+}$.

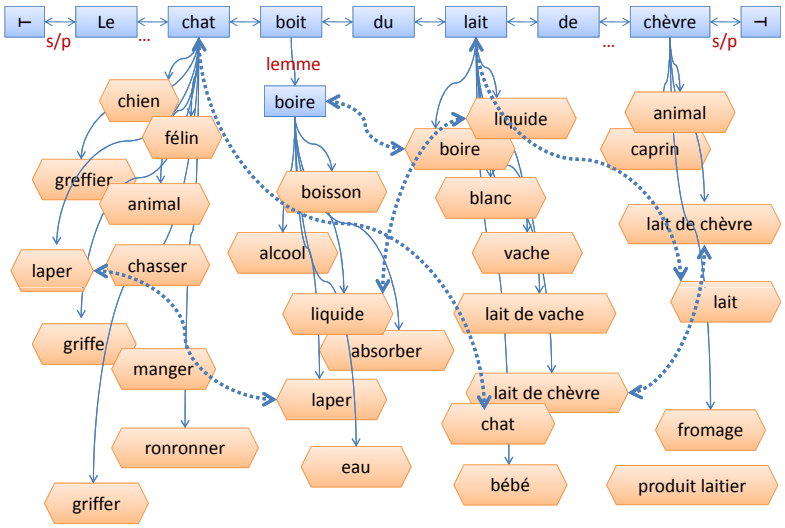


FIGURE 5.10 – L'activité des agents explorateurs fait émerger des nœuds conceptuels au sein de l'espace de travail.

Nous ferons remarquer que le modèle ne fait pas intervenir explicitement de vecteur d'idées et de calcul de similarité pour le choix de la destination. Cependant, le réseau lexical traduit implicitement cette information, avec des termes plus ou moins fortement reliés et donc plus ou moins similaires.

Fusion de nœuds et agents de consolidation

Si deux *nœuds conceptuels équivalents* (portant la même étiquette) sont situés sur des intervalles distincts et relativement voisins, alors ils sont susceptibles d'être fusionnés, par l'entremise des *agents de consolidation*, qui réalisent une forme de diffusion de l'information à travers le réseau. Ce processus de fusion proprement dit a lieu lors de la phase de nettoyage du réseau. Un agent de consolidation est issu d'un nœud conceptuel et cherche dans son voisinage (intervalles voisins) si un nœud équivalent existe, si tel est le cas, un arc d'équivalence est créé. Si au moment de la fusion, un arc d'équivalence dépasse un certain seuil d'activation, alors la fusion s'opère.

La fusion des nœuds est un élément déterminant du modèle car il permet de construire des ponts entre les éléments conceptuels du réseau de travail. Ces ponts sont souvent de nature thématique, ce qui représente environ 75% de l'information nécessaire à une désambiguïsation lexicale. Si un concept est commun à deux nœuds peu éloignés alors, la fusion permet de réaliser en pratique ce partage. Par exemple, considérons le segment textuel suivant : *carambolage sur l'A7.* (figure 5.11).

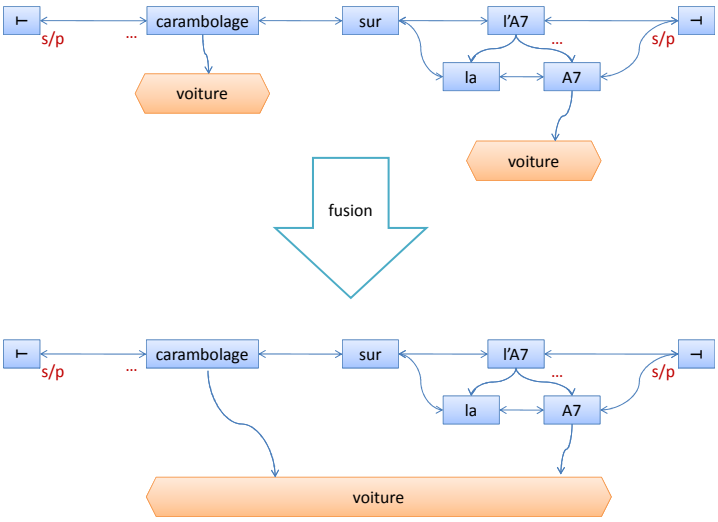


FIGURE 5.11 – Fusion de nœuds

Les nœuds concrets *carambolage* et *A7* ont été relié chacun à un nœud conceptuel *voiture* via l'activité d'agents propagateurs. Ces deux nœuds conceptuels ayant des étiquettes équivalentes et n'étant pas très éloignés, ont une probabilité d'être fusionnés lors de la phase de nettoyage du réseau. La fonction de probabilité est de la forme $a/1000$ où a est l'activation du lien entre les deux nœuds.

5.3.2 Découverte de constituants et de dépendances

Identifier les constituants a non seulement un intérêt en soi, mais est surtout une étape vers l'identification des dépendances entre les éléments du texte.

Agents d'agrégation et multi-termes

Un *agent d'agrégation* peut être produit par un nœud concret et cherche à construire dans l'environnement de travail des séquences de termes (ou de constituants) existant dans le réseau lexical. une fois créé, il se déplace de la gauche vers la droite, et dès qu'une séquence est reconnue, le nœud physique matérialisant cette sous-séquence est créé et l'agent meurt. Si aucune sous-séquence n'est trouvée au terme d'une certaine fenêtre (empiriquement fixé à 10 mots), l'agent meurt sur le dernier nœud où il se trouve. Les éléments de la séquence sont reliés au segment crée par des relations de dépendance, et plus précisément de *constituance*. Comme pour tous les nœuds physiques, les liens successeurs et prédécesseurs sont également introduits.

Par exemple, considérons l'entrée *lait de chèvre* présente dans le réseau lexical. Dans l'environnement de travail, nous avons (entre autres) le sous-graphe :

$$\cdots \leftrightarrows [\text{lait}] \leftrightarrows [\text{de}] \leftrightarrows [\text{chèvre}] \leftrightarrows \cdots$$

Un agent d'agrégation créé par le nœud [lait] va se déplacer vers la droite en suivant les arcs successeurs ($\leftrightarrows$) et atteindre successivement les nœuds [de] et [lait]. L'agent meurt sur le nœud qu'il a créé, à savoir [lait de chèvre].

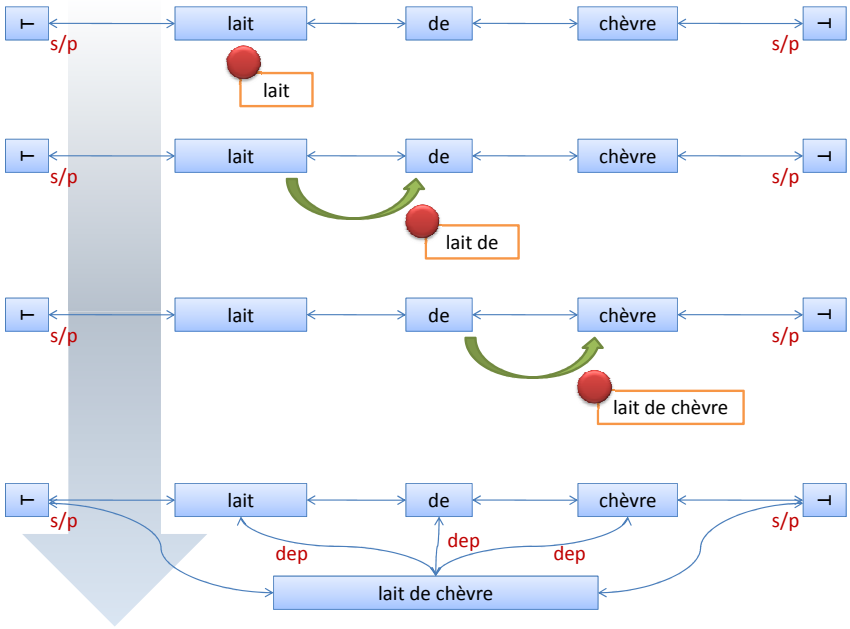


FIGURE 5.12 – Reconnaissance d'un multi-terme. Les liens notés `s/p` sont les liens *successeurs* et *prédécesseurs*. Les liens `dep` sont des liens de dépendance/constituance.

Un segment préfixe d'un autre sera construit en premier, puis à partir de ce segment construit, le second sera introduit. Par exemple, dans le réseau lexical nous avons les deux termes composés *pièce de machine* et *pièce de machine à laver*. Supposons que dans le texte se trouve le segment ≪ ...pièce de machine à laver... ≫. Durant le calcul, le segment court sera d'abord trouvé à partir du nœud *pièce*, et le segment long sera trouvé ensuite à partir du nœud *pièce de machine*. Le processus fonctionne de façon similaire pour tout segment infixe non vide. Globalement, le processus décrit ici rappelle un peu les algorithmes d'analyse tabulaire, mais sans pour autant systématiquement représenter toutes les analyses possibles.

Agents d'agrégation et constituants

La présence dans le réseau lexical de nœuds correspondant à des constituants permet leur construction à l'aide des agents d'agrégation. Ces nœuds sont éventuellement reliés par une relation de renommage qui en fait est la relation de dépendance *dep*. L'ensemble forme une (pseudo) grammaire hors-contexte. Non seulement, les nœuds, par leur présence, définissent les segments de constituants acceptables, mais par la relation de renommage indiquent ce à quoi ils sont équivalents. Par exemple, nous avons de façon non exhaustive :

- $[ADJ : N :] \xrightarrow{dep} [GN :]$
- $[N : ADJ :] \xrightarrow{dep} [GN :]$
- $[DET : N :] \xrightarrow{dep} [GN :]$
- $[V : GN :] \xrightarrow{dep} [GV :]$
- $[V : du\ GN :] \xrightarrow{dep} [GV :]$

Dans la liste ci-dessus, le segment [ADJ N] est acceptable et pourra être créé par un agent d'agrégation. Par la suite, un agent d'exploration pourra éventuellement introduire le nœud [GN] via la relation *dep* et en faire une copie locale. De façon similaire, si dans l'environnement de travail nous avons $[DET] \xrightarrow{succ} [N]$ et que dans le réseau lexical, il existe un nœud [DET N] (ce qui est donc le cas), alors une copie locale de ce nœud sera construite par l'agent d'agrégation.

En ce qui concerne le processus, cette construction est exactement de même nature que celle des multi-termes où les mêmes agents sont à l'œuvre. Un agent d'exploration ne pourra parcourir la relation *dep* vers un des nœuds destination que si auparavant la structure a été construite. Il n'y a pas de communication directe entre les agents, mais indirecte par modification de l'environnement.

Le renommage permet donc une spécification aisée de règle de construction de dépendance, par une factorisation implicite des symboles non-terminaux (rien de nouveau ici, c'est le principe même d'une grammaire). Un autre usage en est l'*effacement* qui permet de *mettre entre parenthèse* certains constituants :

- $[ADJ :] \xrightarrow{eff} []$
- $[ADV :] \xrightarrow{eff} []$
- $[CIRC :] \xrightarrow{eff} []$

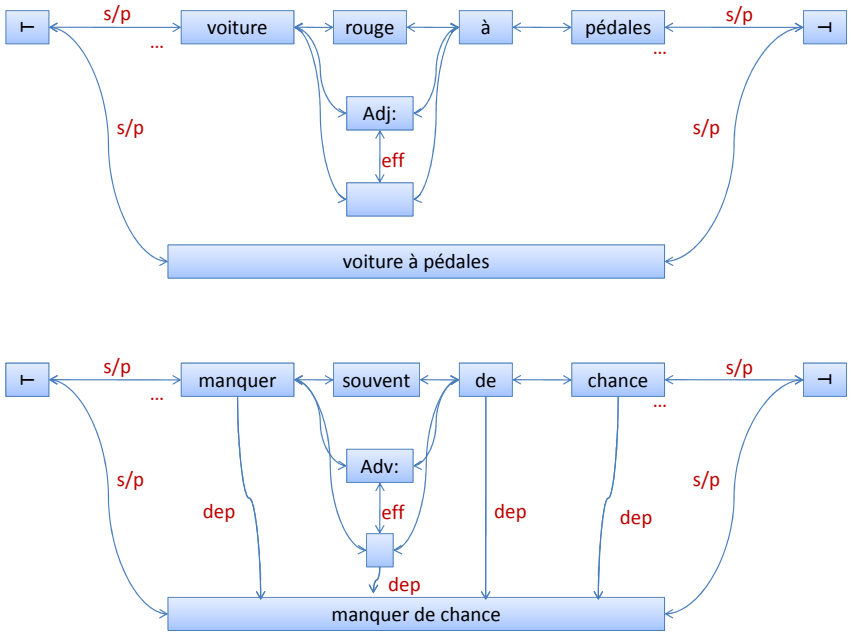


FIGURE 5.13 – Effacement et reconstitution de multi-termes. Le multi-terme identifié garde une trace de sa construction à travers des relations de dépendance.

Les trois relations effacements ci-dessus signifient qu'un adverbe, un adjectif ou un groupe circonstanciel peuvent être ignorés. Dans la phrase ≪ *La 4L roule lentement sur la nationale.* ≫, l'ef-

facement de *lentement* rend possible le rapprochement entre *roule* et *nationale*, et augmente les chances de découverte dans le réseau lexical de la relation $[rouler] \xrightarrow{loc} [nationale]$. De plus, l'effacement permet aux agents d'agrégation de trouver des entrées existantes dans le réseau sous une forme non connexe dans le texte. Par exemple, le syntagme *riz blanc parfumé* via l'effacement de *blanc* va conduire à la création du segment *riz parfumé* (qui existe dans le réseau lexical) en plus du segment *riz blanc* (le *segment riz blanc parfumé* n'existant pas dans le réseau lexical ne sera pas retrouvé).

Le nœud physique vide [], n'existant pas dans le réseau lexical, est donc complètement stérile pour les agents explorateurs ou agrégateurs. La ponctuation n'étant *a priori* jamais effacée, les risques d'effacements inconsidérés par propagation sont réduits.

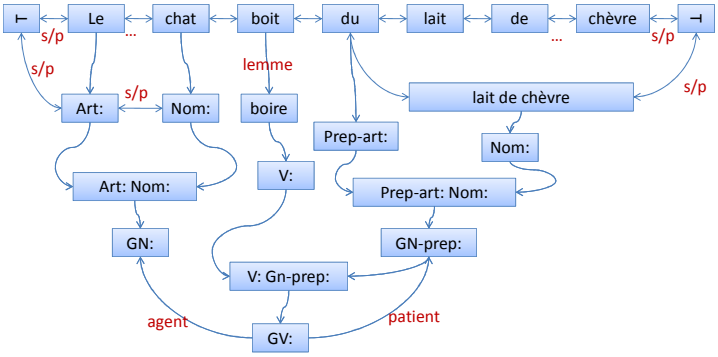
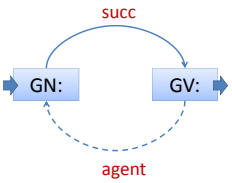


FIGURE 5.14 – Construction de constituants et des rôles syntaxiques

Construction de dépendances et de rôles

La construction des dépendants (et des relations syntaxiques associées) se fait par un mécanisme de règles appliquées par des *agents d'inférence*. Ces règles sont des séquences présentes dans le réseau lexical sous une forme particulière, et correspondent à des automates à états finis sans cycle. Par exemple, nous avons (en notation textuelle à gauche et graphiquement à droite) :

$$[\mathrm{GN}_1 :] \xrightarrow{succ} [\mathrm{GV} :] \xrightarrow{agent} [GN_1 :]$$



(dans la notation textuelle les indices identiques indiquent l'identité de nœud dans le réseau lexical.) Cette règle stipule que si un groupe nominal est suivi d'un groupe verbal, alors l'agent du groupe verbal est le groupe nominal.

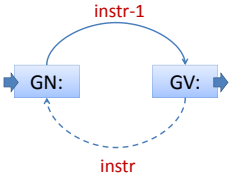
L'agent d'inférence choisit aléatoirement à partir du nœud initial l'automate qu'il va prendre. Il suit le chemin dans l'espace de travail. S'il parvient à atteindre l'état de sortie, alors la relation induite est construite ou renforcée (la relation peut déjà être présente dans l'espace de travail). L'agent meurt sur l'état de sortie. Donc dans l'exemple ci-dessus, si un agent propagateur passe d'un GN à un GV via la relation de succession, alors comme GV est l'état final, il créé ou renforce la relation *agent* de GV vers GN.

Quelques précisions sur le formalisme et les notations s'imposent :

1. une relation *r*-1 est la relation inverse de *r*. Elle peut être présente explicitement dans l'espace de travail ou non. Si la relation *r* entrante est présente, alors l'agent peut parcourir cet arc à l'envers si l'automate a comme transition *r*-1. Si *r* entrant et *r*-1 sortant sont tous les deux présents, alors l'agent prend un des deux arcs au hasard. Nous rappelons qu'emprunter un arc renforce son activation ;
2. Nous présentons ici quelques automates séparément, mais en pratique pour l'implémentation, les automates sont unis en un seul pour chaque état initial (ceci ne change rien en soit mais à la fois simplifie fortement l'implémentation et est beaucoup plus efficace) ;
3. les automates sont simples et forment un chemin unique que parcourt l'agent. Le chemin peut être aussi long que nécessaire, mais un automate bien formé doit disposer d'un état de sortie. Nous ferons remarquer qu'il peut être intéressant de définir des règles sans état de sortie, ainsi les agents le prenant n'en sortiront jamais et parcourront le chemin en question en les renforçant ;
4. nous notons un nœud quelconque `:x` et une relation quelconque `:r`. Il s'agit de variables qui seront associées à ce que l'agent parcourt dans l'espace de travail. Si nous avons besoin de plusieurs variables, nous les indiçons.

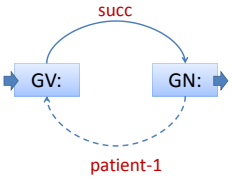
Les relations inverses peuvent être complétées par leur relation inverse (donc normale) avec une règle de la forme (pour instrument) :

$$[\text{GN}_1 :] \xrightarrow{instr-1} [\text{GV} :] \xrightarrow{instr} [GN_1 :]$$



qui permet donc de produire la relation inverse d'instrument, ce qui est particulièrement utile pour *redresser* la relation. Nous rappelons qu'établir des boucles est intéressant pour l'auto-renforcement des structures produites, cependant de telles boucles ne se maintiennent que si elles trouvent de l'énergie par ailleurs. De façon similaire, nous avons :

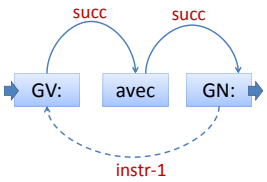
$$[\text{GV}_1 :] \xrightarrow{succ} [\text{GN} :] \xrightarrow{instr-1} [GV_1 :]$$



la présence des boucles d'auto-renforcement dans l'espace de travail est une des clés de la réussite de l'analyse. Par exemple, il ne suffit pas de savoir que *chat* est l'agent de *boire*, mais également que *boire* a pour agent *chat*. Les deux nœuds *chat* et *boire* vont se renforcer mutuellement et avoir une activité de charge et décharge importante, qui est non seulement un critère important pour la lecture du résultat, mais induit un niveau d'activation élevé des relations correspondantes. Par contre, il serait faut de croire qu'une boucle une fois créée va nécessairement perdurer. En effet, une boucle isolée ou peu alimentée va finir par disparaitre, d'abord par attrition de nœuds qui produisent des agents, puis par évaporation de l'activation des arc amenant à leur suppression, et enfin la destruction des nœuds non reliés.

Les fonctions syntaxiques induites par les prépositions peuvent être également calculées (de façon ambigüe) avec des règles. Par exemple, pour la préposition *avec* :

$$[GN_1 :] \xrightarrow{succ} [avec] \xrightarrow{succ} [\mathrm{GN}_2 :] \quad \xrightarrow{instr-1} [GN_1 :]$$



Les prépositions sont généralement ambigües, et plusieurs règles peuvent les régir.

5.3.3 Inférence, inhibition et lecture du résultat

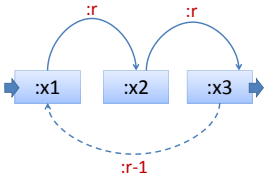
Pour l'analyse sémantique, il semble intéressant de rajouter deux mécanismes génériques d'inférence et d'inhibition. L'inférence consistera à partir d'une certaine configuration à supposer la validité d'une relation particulière. L'inhibition consistera à étendre l'exploitation du réseau lexical aux relations à valeur négative.

Inférence de relations

Les règles d'inférence que nous utilisons sont similaires à celles présentées dans ce chapitre avec le jeu à questions fermées ASKIT où des propositions de relation sont présentées à l'utilisateur, et en présente une forme de généralisation. Il s'agit de règles de la même forme que les règles présentées ci-dessus et qui sont exploitées de la même manière par les agents d'inférence. La première forme de règle consiste à effectuer un transfert d'information (par création d'un arc) sous la forme de triangulations dont voici la forme générique :

Forme générale de la règle de triangulation :

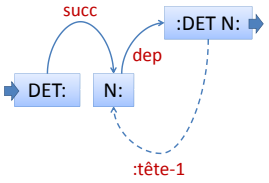
$$[\ :x1] \xrightarrow{:r} [\ :x2] \xrightarrow{:r} [\ :\mathrm{x3}] \xrightarrow{:r-1} [\ :x1]$$



Nous n'intégrons pas cette forme là dans le système, car elle semble trop générique et productrice de bruit potentiel. Mais certaines formes particulières sont intéressantes, comme notamment la propagation de dépendances, qui est la règle ci-dessous avec $r =$ dep. L'identification de la tête est importante pour aider aux basculement des rôles syntaxiques des constituants vers les termes (une descente, donc). Pour les noms, nous avons donc deux règles :

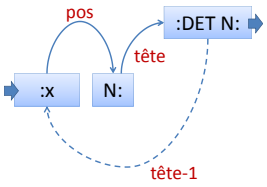
(1) une règle de redescente de la tête vers le la partie du discours (N :) pour les formes [DET : N :] :

$$[DET :] \xrightarrow{succ} [N :] \xrightarrow{dep} [\mathrm{DET} : \mathrm{N} :] \xrightarrow{tete-1} [N :]$$



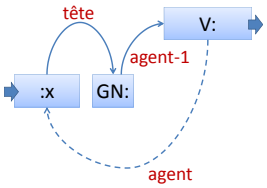
et (2) une règle de redescente de la tête de la partie du discours vers le terme :

$$[\,:x] \xrightarrow{pos} [N:] \xrightarrow{tete} [\text{GN}:] \xrightarrow{tete-1} [\,:x]$$



Enfin, s'il s'agissait du sujet, alors la règle suivante permet l'héritage de la relation *agent* pour le terme :

$$[\,:x] \xrightarrow{tête} [GN:] \xrightarrow{agent-1} [\text{V}:] \xrightarrow{agent} [\,:x]$$



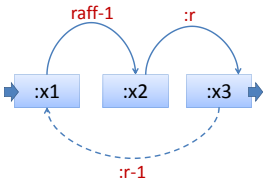
Il est possible de faire de même pour les groupes verbaux (la tête est le verbe) pour les formes [V : GN :] :

$$[V:] \xrightarrow{succ} [GN:] \xrightarrow{dep} [\text{V} : \text{GN}:] \xrightarrow{tete-1} [V:]$$
$$[\,:x] \xrightarrow{pos} [V:] \xrightarrow{tete} [\text{V} : \text{GN}:] \xrightarrow{tete-1} [\,:x]$$
$$[\,:x] \xrightarrow{tête} [GV:] \xrightarrow{patient} [\text{GN}:] \xrightarrow{patient-1} [GN:]$$

Nous définissons également une règle d'héritage pour les raffinements (acceptions) d'un terme :

Héritage de relation pour un raffinement :

$$[\,:x1] \xrightarrow{raff-1} [\,:x2] \xrightarrow{:r} [\,:\text{x3}] \xrightarrow{:r-1} [\,:x1]$$



Si un terme :x1 est un raffinement de :x2 et que :x2 est relié à :x3 par :r, alors le raffinement :x1 est relié à :x3 par r. Cette relation n'est pas toujours vraie mais constitue une heuristique intéressante. À nouveau, cette relation potentielle de découverte via le texte sera ajoutée aux stocks de questions de ASKIT afin d'être validée par la suite. Si elle existe dans le réseau lexical, elle sera alors simplement copiée.

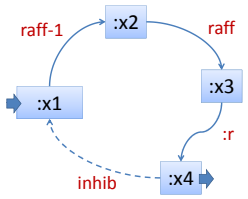
Inhibition

Le modèle permet au prix d'une modification mineure la prise en charge de relations à valuation négative. Lors de la sélection d'une relation, la pondération pour le choix pseudo-aléatoire sera la valeur absolue de son activation. Ainsi, des agents propagateurs peuvent sélectionner des relations inhibitrices, avec une probabilité covariante à leur force d'inhibition. Un agent atteignant un nœud via une relation à valeur négative diminue l'activation du nœud en question du quantum q_{n-}. Du point de vue du nœud ayant créé cet agent, il s'agit de *faire taire* le nœud de destination. En effet, ce dernier aura moins d'énergie disponible pour la création d'agents, et s'exprimera donc moins dans l'espace de travail.

L'inhibition se matérialise également par mise en concurrence des raffinements via une relation générique d'inhibition *inhib*. Cette relation a le même fonctionnement que les relations typées à valeur négative, à savoir de diminuer l'énergie du nœud destination lors du passage d'un agent d'exploration. Le schéma est le suivant :

Création d'une relation d'inhibition pour un raffinement :

$$[\,\text{:x1}\,] \xrightarrow{raff-1} [\,\text{:x2}\,] \xrightarrow{raff} [\,\textit{:x3}\,] \xrightarrow{:r} [\,\text{:x4}\,] \xrightarrow{inhib} \textit{:x1}$$



La relation d'inhibition n'est créée que s'il n'existe pas, soit dans l'espace de travail, soit dans le réseau lexical, de relation $[\,\textit{:x4}\,] \xrightarrow{:r} [\,\textit{:x1}\,]$, auquel cas, celle-ci est recopiée (et donc il n'y a pas inhibition). Cette règle va permettre ainsi aux raffinements d'un même terme de tenter de s'inhiber mutuellement par l'intermédiaire des termes qui l'activent. Cette mise en concurrence pourrait se résumer par l'adage suivant, du point de vue d'un raffinement : *si quelque chose m'active et pas toi, alors cela t'inhibe.*

Nos expériences préliminaires sur ce modèle ont semblé démontrer l'importance des relations d'inhibition à la fois dans la pertinence du résultat calculé mais également concernant le nombre de cycles nécessaires au calcul. Le contraste d'activation entre différentes acceptions s'en retrouve renforcé. Globalement, nous avons observé que la prise en compte de l'inhibition avait les effets suivants :

1. une augmentation de l'ordre de 10 % pour la sélection des acceptions ;
2. une augmentation de l'ordre de 10 % pour le contraste (la différence de valeur d'activité) entre les différentes acceptions ;
3. le niveau de participation des relations inhibitrices est de deux pour une relation activatrice (soit 66 % du total) ; c'est-à-dire que bien qu'elles soient bien moins nombreuses dans le réseau lexical, elles sont deux fois plus mises à contribution lors de l'analyse.

Lecture de l'analyse

Le modèle d'analyse que nous avons présenté produit un grand nombre de nœuds (dans la majorité conceptuels) et de relations. La lecture du résultat se fait par filtrage en fonction de l'activité des nœuds et de l'activation des arcs entre les nœuds sélectionnés. En effet, le niveau d'énergie d'un nœud à un instant t (ou en moyenne sur une fenêtre temporelle) n'est pas significatif, car il est en moyenne toujours plafonné aux alentours de 0. Ceci s'explique par la mécanique de création des agents, où chaque nœud produit autant d'agents qu'il le peut à chaque cycle de façon probabiliste jusqu'à rencontrer un échec. La probabilité de créer un agent à activation 0 est de $1/2$ ceci expliquant pourquoi les activations des nœuds en fin de chaque cycle sont basses. Une mesure de l'activité d'un nœud est clairement le nombre d'agents qu'il a produit.

Nous avons implémenté un prototype d'analyse holistique sur les principes énoncés ci-dessus. Les données sont contenues dans le réseau lexical JeuxDeMots, et nous avons défini une trentaine de règles simples (dont celle données en exemple). Si nous ne considérons que la désambiguïsation lexicale, nous obtenons des résultats qui se situent à environ 3 % de ceux du modèle biosinspiré. Le nombre de cycles pour arriver aux solutions est diminué d'environ 25 %, mais le temps de calcul est largement doublé (il y a bien plus d'agents dans l'environnement). Concernant les relations relevant des rôles sémantiques (réduits à *agent*, *patient*, *instrument*), elles sont correctement découvertes dans 95 % des cas.

Conclusion du chapitre 5

Le réseau JeuxDeMots fournit une base de connaissances utilisable dans d'autres domaines que le TALN. L'exploitation d'information lexico-sémantique a montré son utilité potentielle en ingénierie des modèles. Une approche relevant d'une triade propagation-similarité-fusion a été esquissée et semble à rapprocher du modèle d'analyse holistique de textes également présenté ici. Cette dernière semble une piste intéressante aussi bien au niveau du contrôle et de la modularité que des résultats obtenus.

La taille des données constitue également un défi pour les algorithmes de visualisation de graphes. En particulier est posée la question de la visualisation interactive multi-échelle d'un tel type de graphe. Cette approche nécessite comme traitement préalable un découpage en clusters hiérarchiques qui est à la fois coûteux et difficile à définir. La question du choix de la fonction de similarité permettant un découpage intuitif reste largement ouverte.

L'acquisition de relations *plus difficiles* peut finalement être réalisée via des approches contributives qui se révèlent complémentaires à l'acquisition via l'agrément par pairs, et peuvent relever elles aussi du consensus populaire. La multiplication des approches permet de réduire l'effet des biais de chacune, mais également de cibler l'acquisition d'informations nouvelles (comme par exemple la polarité moyenne associée aux termes du lexique).

Articles adjoints au chapitre 5

Falleri J.-R., Prince V., Lafourcade M., Dao M., Huchard M., Nebut C. *Using Natural Language to Improve the Generation of Model Transformation in Software Design.* International Multi-Conference on Computer Science and Information Technology, Pologne (2009), 8 p.

Joubert A., Lafourcade M.,Schwab D., Zock M. *Évaluation et consolidation d'un réseau lexical via un outil pour retrouver le mot sur le bout de la langue* TALN'2011, Montpellier (2011), 12 p.

G. Artignan, M. Hascoët, M. Lafourcade *Multiscale Visual Analysis of Lexical Networks.* Proc. of 13th International Conference Information Visualisation IV 09, (2009), 12 p.

Annexe : captures d'écran

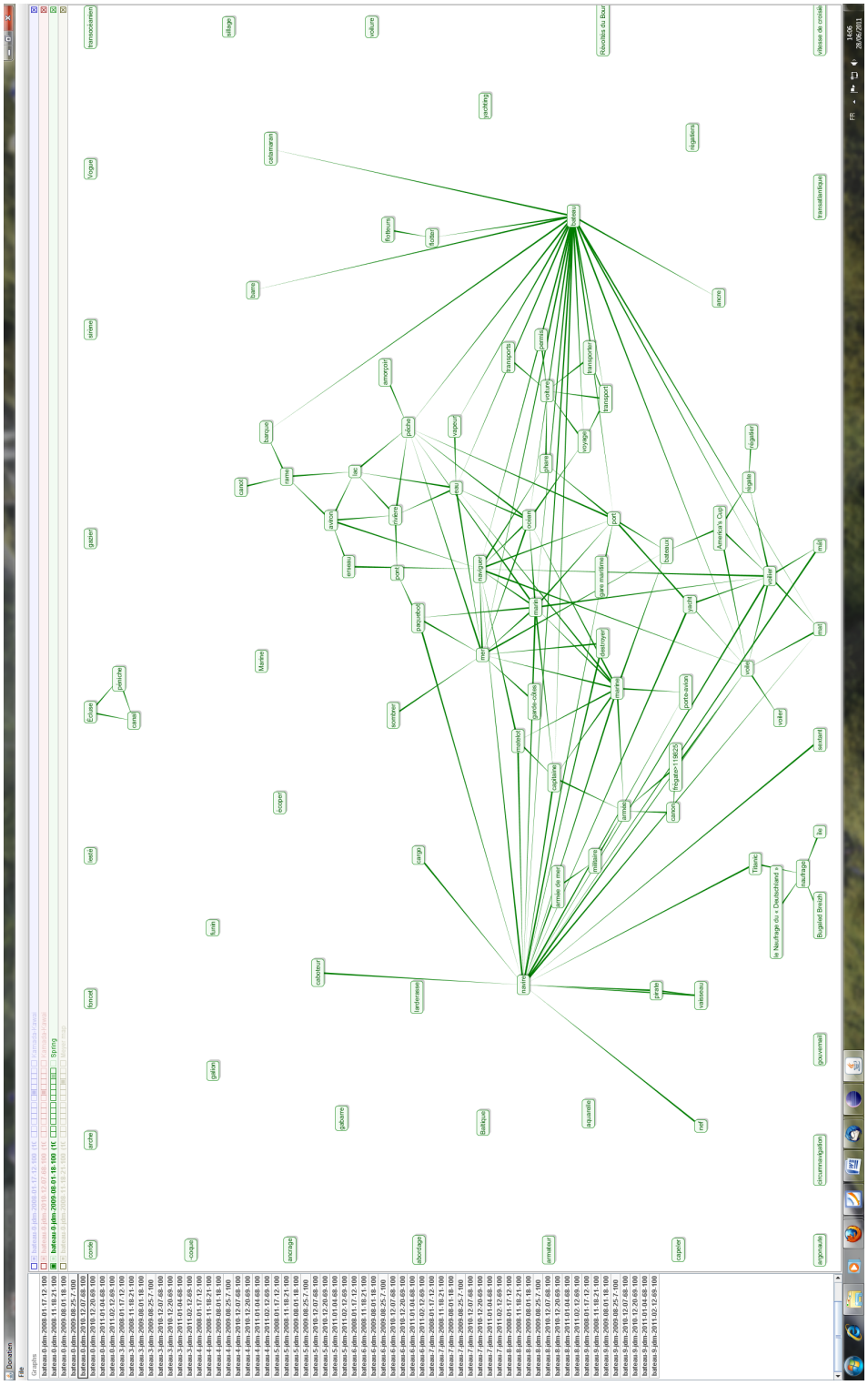


FIGURE 5.15 – Visualisation d'une sous-partie du réseau lexical de JeuxDeMots avec un algorithme à ressort (Spring) (travaux de M. Hascöet).

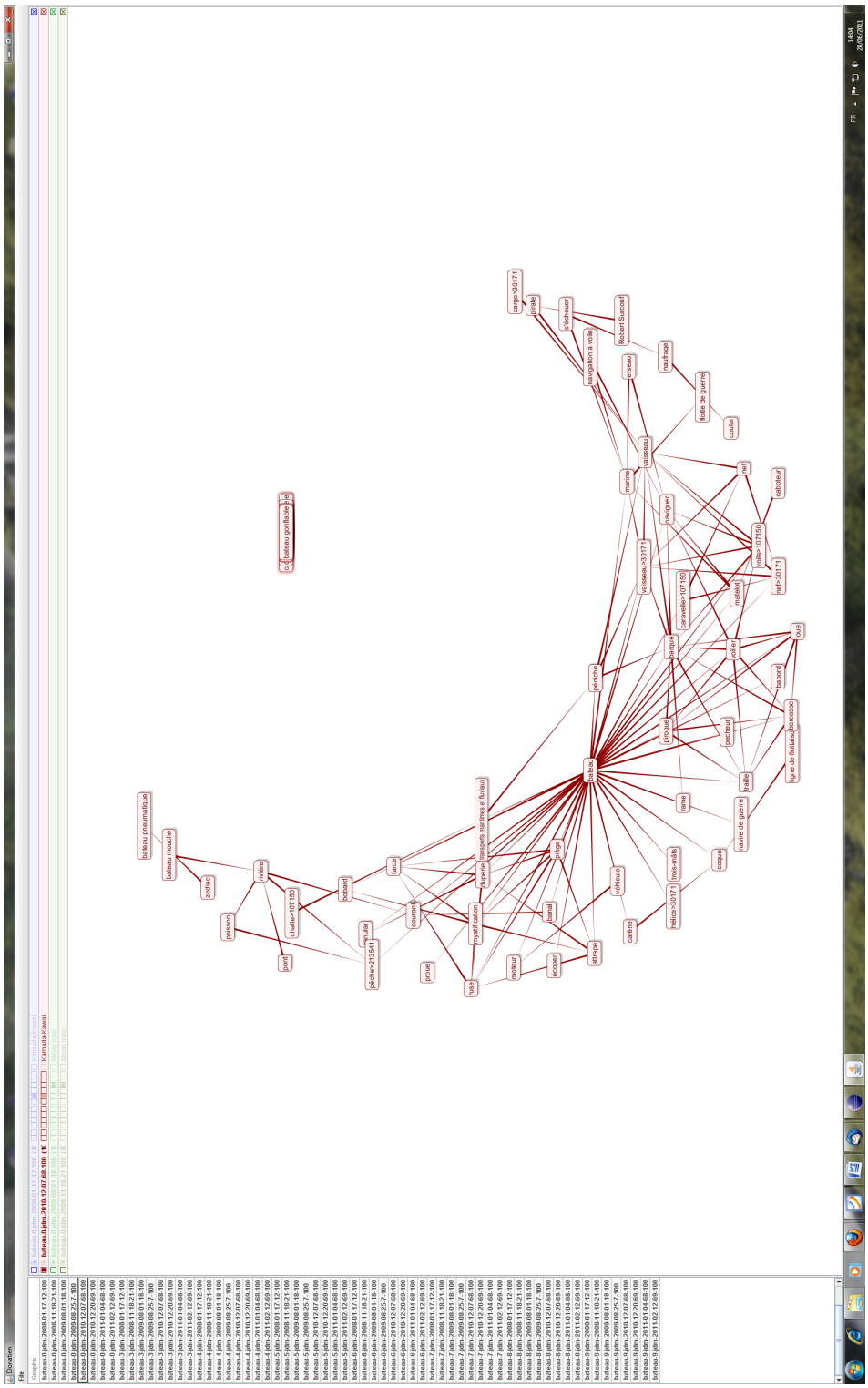


FIGURE 5.16 – Visualisation d'une sous-partie du réseau lexical de JeuxDeMots avec un algorithme Kamada-Kawai (travaux de M. Hascöet).

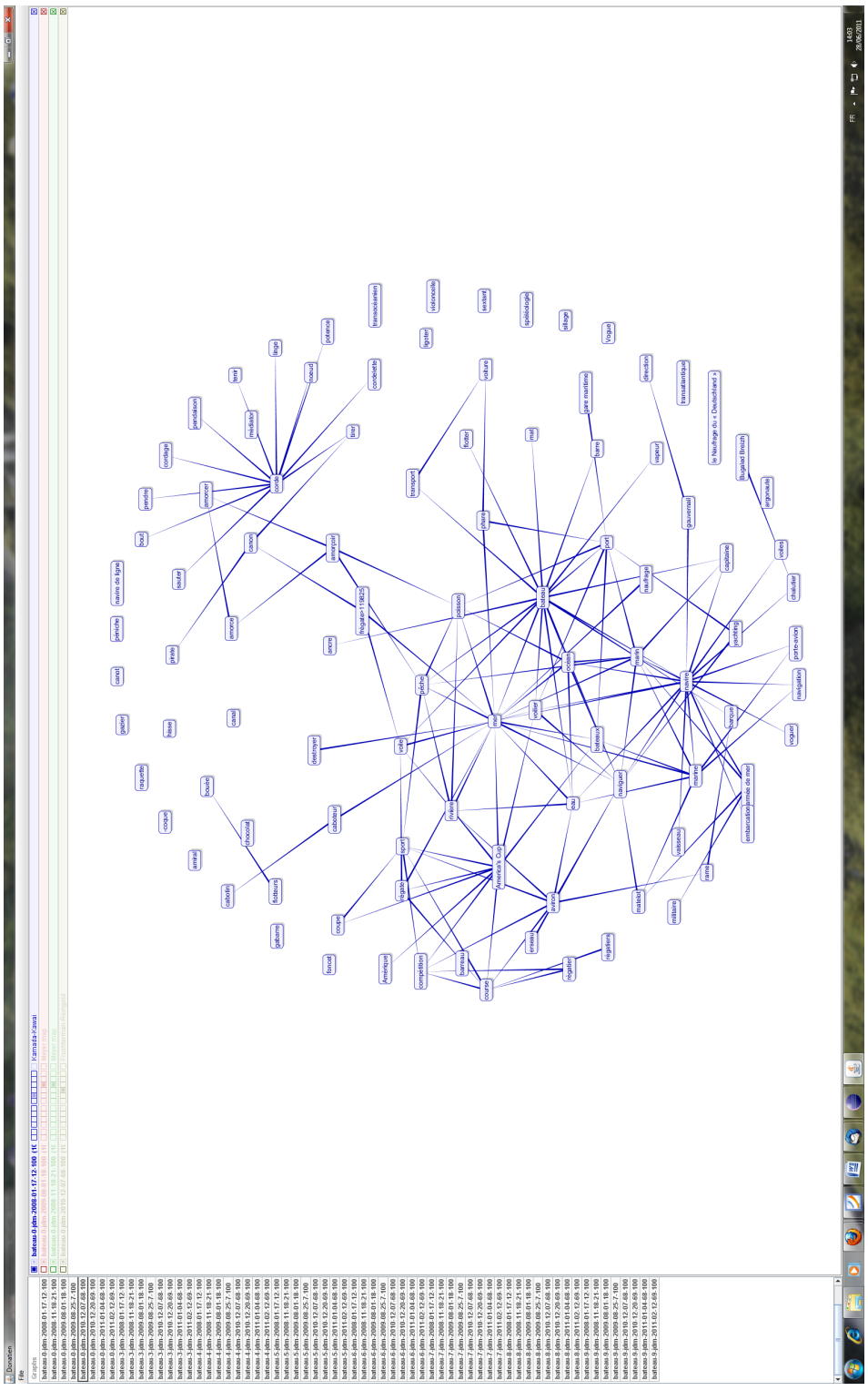


FIGURE 5.17 – Visualisation d'une sous-partie du réseau lexical de JeuxDeMots avec un algorithme Kamada-Kawai (bis) (travaux de M. Hascöet).

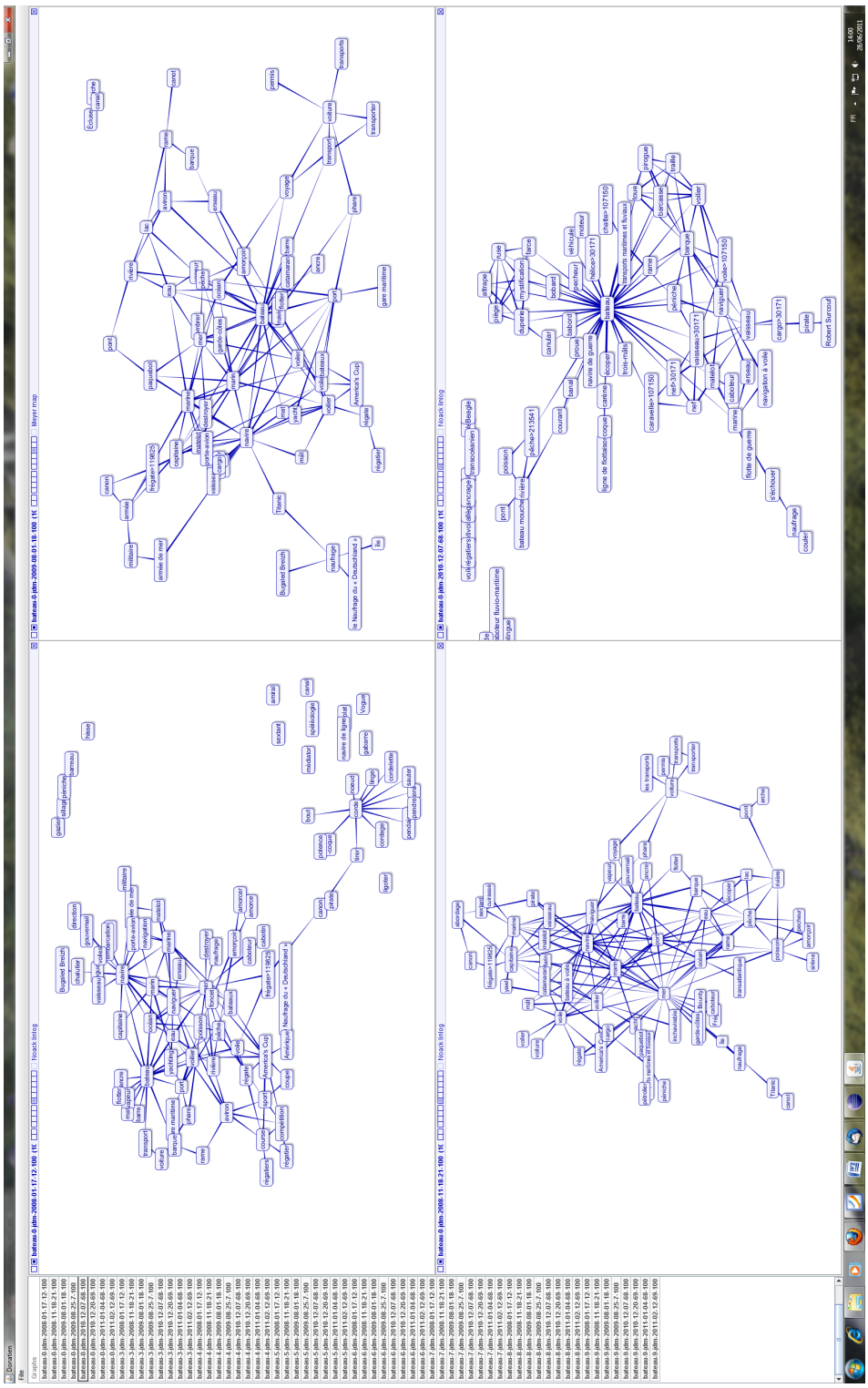


FIGURE 5.18 – Visualisation multiple (algorithme Spring) d'une sous-partie du réseau lexical de JeuxDeMots selon le type de relation (travaux de M. Hascöet)).

Using Natural Language to Improve the Generation of Model Transformation in Software Design

Jean-Remi Falleri
Univ. Montpellier 2
and LIRMM-CNRS
161 Ada Street F34392 Montpellier
France
falleri@lirmm.frl

Violaine Prince
Univ. Montpellier 2
and LIRMM-CNRS
161 Ada Street F34392 Montpellier
France prince@lirmm.fr

Mathieu Lafourcade
Univ. Montpellier 2
and LIRMM-CNRS
161 Ada Street F34392 Montpellier
France
lafourcade@lirmm.fr

Michel Dao
Orange Labs
Issy Les Moulineaux
France
michel.dao@orange-ftgroup.com

Marianne Huchard
Univ. Montpellier 2
and LIRMM-CNRS
161 Ada Street F34392 Montpellier
France
huchard@lirmm.fr

Clementine Nebut
Univ. Montpellier 2
and LIRMM-CNRS
161 Ada Street F34392 Montpellier
France
nebut@lirmm.fr

***Abstract*—Among the present crucial issues in UML Modeling, one of the most common is about the fusion of similar models coming from various sources. Several similar models are created in Software Engineering and it is of primary interest to compare them and, when possible, to craft a general model including a specific one, or just identify models that are in fact equivalent. Most present approaches are based on model structure comparison and alignment on strings for attributes and classe names. This contribution evaluates the added value of several combined NLP techniques based on lexical networks, POS tagging, and Dependency Rules application, and how they might improve the fusion of models. Topics : use of NLP techniques in practical applications.**

I. Introduction

Natural Language Processing (NLP) is more and more a topic of interest for Model Driven Engineering in Software Design. Software is designed worldwide for almost every type of task involving information, and has to be exchanged between different teams geographically and temporally distant. For the same kind of applications, one might find several 'metamodels' (*i.e.* abstract 'meta' specifications) independently developed, and also several versions of the same metamodel with different names and designations. Since those abstract structures generate various software specifications (called models), then compatibility needs to be ensured. Usually this problem is solved using manually written and *ad hoc* model transformations. The latter are not difficult to write per se, but are so numerous that they heavily impact the project work load. Members Sotfware Engineering community has thus suggested to NLP researchers to help them to find astute methods to automatically generate an alignment between two similar metamodels. Schema alignment already exists in domains such as semantic web, ontology integration, e-commerce and so forth. It takes as input two schemas and produces as output a set of relations (*e.g.* equivalence and subsumption) between the entities of the two input schemas. Concretely, a schema can be an XML Schema, an ontology, a database schema or an object-oriented class model. Despite the variation between these formats, the mechanisms involved to perform the match operation are highly similar. The NLP community has already contributed to facilitate Schema alignment [Rahm and Bernstein 2001]) whether for databases (*e.g.*, [Duchateau et al., 2007]), conceptual graphs (*e.g.*,[Montes y Gomez et al., 2007]), or domain specific ontologies (*e.g.*, [Fan et al. 2007] for medical ontologies). The meta-model alignment operation aims at finding a set of correspondences between elements (classes, attributes, references and enumerations) from a source meta-model and elements from a target metamodel. Those correspondences can be used later in several tasks:

- Automatic generation of a model transformation,
- Comparison of two models conforming to two different meta-models,
- Increasing efficiency of model merging or composition, as the last step after model transformation and model comparison.

This contribution focuses on the first step, as a necessary requirement for model merging. The goals our work is intended to achieve are the following:

- to discover possible relations between entity identifiers appearing in models: we would thus rely on the possible lexical or semantic relationships induced by names assigned to the model entities. For instance, if two class identifiers are synonyms in a thesaurus, this might suggest a possible redundancy between those two classes.
- if models elements are generated by meta-modeling tech-

niques without identifiers, to try to assign them names according to the semantics of the surrounding other elements (a topically driven name assignment).

Since both goals need an extensive description, this contribution sticks to the first of the preceding items. Lexical relations between identifiers are at the core of the added value of NLP to this task. In next section, the important lexical relations and their modeling are explained. Then the application is detailed in the following section: how compound identifiers are segmented, tagged with a POS tagger, and a dependency analysis assigns a function and induces ontological relations between terms. Experiments have been run on an existing UML modeling corpus, and their results show that NLP techniques have largely enhanced the automation of models transformation. Conclusion summarizes the benefits from such a cooperation and indicates the next tracks that are currently followed in order to succeed in this task.

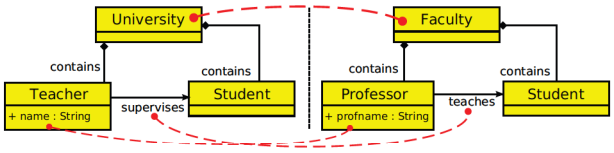


Fig. 1. Example of two models comparison under a general context. Correspondances are found, mainly synonymy, and most certainly those two models could be fusioned into one.

II. Modeling Lexical Relations Between Items Identifiers

Discovering lexical relations between identifiers (terms) appearing in models, seemed to be the first step for a semantic approach of models before transformation. We focused on possible ontological relations (synonymy, hyponymy, ...) between identifiers. Those relations are defined on the set of terms, but they are mostly context sensitive. For example, in a medical context *affection* is a synonym of *disease*, but in another context it may not be the case. Thus, to achieve a correct modeling, relations have to be context-dependent.

A. Basic Items

The **set of terms** is the set of correct identifiers in the models. A **context** is the formalization of a given domain where terms can hold specic relation occurences. Simply speaking, a context is a term (or more generally a set of terms) that specializes a given term. If the context is empty, then we assume we are in the most general domain of common knowledge. Let T be a set of terms. If c belongs to T then it can be a context. For example, *plane — aeronautics* is close to *aeroplane — aeronautics*. But, *plane — aeronautics* and *plane — mathematics* certainly do not refer to the same meaning. Terms are assumed to be available through a lexical network. In its most general definition, a lexical network is a set of words (with or without specific context) linked together with relations. Relations can be of various types [Budanitsky and Hirst 2006], ontological (in that case, we speak more often of ontology) and/or lexical (like synonyms, part-of-speech, lemma, etc.). Wordnet ([Miller 1994]) and EuroWordnet ([Vossen 1998]) are typical examples of lexical networks.

For our purpose relations have to be designed with a set of binary predicates describing their properties and rules, determining the nature of the relation. The useful properties are:

- Transitivity: does a relation propagate whenever true?
- Reflexivity: is the relation self-relevant?
- Symmetry: does the relation introduce an order or does it create a possible "similarity"?

The **basic relations** between terms on which we have most focused are:

- Synonymy, restricted to contextual synonymy, that is, when meanings are close according to a given context,
- Hyperonymy, also a contextual relationship, when a term seems to be the name of a "superclass" of a given class, in the model,
- Hyponymy, as the symmetric relation to hyperonymy,
- To a lesser extent, relations such as Meronymy / Holonymy (or part-whole relations), which might appear in some models and for which modeling has only awkward answers to provide.

Other derived relations, such as co-hyponymy or directy hyperonymy or hyponymy are also described hereafter, because of their usefulness to modeling in sofware design.

B. Relations Definitions For Model Transformation

Here follows the definitions of our relations:

1) Hyperonymy: A term $t1$ is a **hyperonym** of $t2$ if it is more general. For example, *vehicle* is a hyperonym of *car* in the context of transports. Here follow some properties if the binary relation$hyper$:

- *transitive*: if $hyper(t1, t2)$ and $hyper(t2, t3)$ then $hyper(t1, t3)$ (example: vehicle, car, fiat 500)
- *strongly antisymmetric*

We can thus consider $hyper$ as a strict partial order on $T|c$.

2) Hyponymy: A term $t1$ is a **hyponym** of $t2$ if it is more specific. For example, *dog* is an hyponym of *animal*. Mathematically, $hypo$ can be also modelized as a binary relation, then we have $hypo(dog, animal)$. Here, follow some properties of the binary relation $hypo$:

- *transitive*: (example: *labrador*, *dog*, *animal*)
- *strongly antisymmetric*

We can thus consider $hypo$ as a strict partial order on $T|c$.

The relations $hyper$ and $hypo$ are the inverse of each other, thus if $hyper(t1, t2)$, then $hypo(t2, t1)$.

3) Synonymy: A term $t1$ is a **synonym** of $t2$ under the context c if it is equivalent to $t2$. For example, *car* is a synonym *automobile* under the context of transports. Here follow some properties of the binary relation sym:

- *transitive*,
- *reflexive*,
- *symmetric*.

We can consider syn as a equivalence relation under $T|c$. One can notice that if, linguistically speaking, reflexivity is not relevant (a term being synonym of itself is not something to be considered as an interesting achievement), for software design purposes, this property introduces this equivalence relation that creates a *class of terms*, the use of which is quite obvious in model comparison.

4) Direct Hyperonymy: A term $t1$ is a **direct hyperonym** of $t2$ if it is directly more general. For example, *vehicle* is a direct hyperonym of *car* under the context of transports. Here follow some properties if the binary relation $dhyper$:

- *strongly antisymetric*

Moreover, we have $dhyper(a,b) \rightarrow hyper(a,b)$.

5) Direct Hyponymy: A term $t1$ is a **direct hyponym** of $t2$ if it is directly more specific. For example, *dog* is a direct hyponym of *animal*. Mathematicaly, $dhypo$ can be also modelized as a binary relation, then we have $dhypo(dog, animal)$. Here follow some properties if the binary relation $dhypo$:

- *strongly antisymmetric*

Morevover, we have $dhypo(a,b) \rightarrow hypo(a,b)$. The relations $dhyper$ and $dhypo$ are the inverse of each other, thus if $dhyper(t1,t2)$, then $dhypo(t2,t1)$.

C. Derived Relations and Less Frequent Relations

As explained before, modeling needs to provide horizontal relations between identifiers, and not only 'vertical' ones. Two terms are cohyponyms, if they are both hyponyms of a common term. Co-hyponymy frequently appears in models created by different teams, and is an issue in models comparison. However, as the notion of co-hyponymy depends strongly on the maximal distance of the common term we want to accept (all terms are cohyponyms of the most general one), we define several versions of this relation.

1) 1co-hyponymy: **1co-hyponymy** indicates that two terms are "children of the same direct parent'.
$1cohypo(a,b) \leftrightarrow dhypo(a,c) \wedge dhypo(b,c) \wedge a \neq b$. As $dhypo$ and $dhyper$ are inverse, we have: $dhypo(a,b) \leftrightarrow dhyper(b,a)$. Then, $dhypo(a,c) \wedge dhypo(b,c) \wedge a \neq b \leftrightarrow dhyper(c,a) \wedge dhyper(c,b) \wedge a \neq b$. Thus, $cohypo(a,b) \leftrightarrow hyper(c,a) \wedge dhyper(c,b) \wedge a \neq b$. Here follow some properties of the relation $1cohypo$:

- *symmetric.*

2) θco-hyponymy: The θco-hyponym is a formalization of a generalized co-hyponymy.
Two terms a and b are θco-hyponyms if a there is common hyperonym h between a and b such as $dmin(a,h) \leq \theta$ and $dmin(b,h) \leq \theta$, $dmin(x,y)$ being the shortest path between two comparable terms for the relation $dhypo$. We call h a θminor of a and b. Here follow some properties of the relation $\theta cohypo$:

- *symmetric.*

We can notice that $\theta cohypo(a,b) \rightarrow \varphi cohypo(a,b), \varphi \geq \theta$.

Here follow an extended version of the θco-hyponymy for a n-tuple $x_1, \ldots, x_n$ argument.

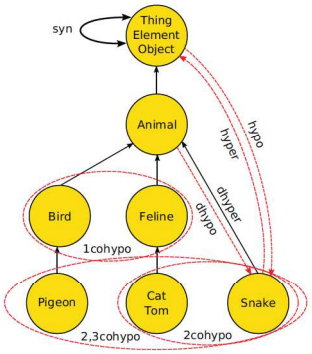


Fig. 2. Lexical relations between terms

3) $\theta_n$co-hyponymy: n terms $x_1, \ldots, x_n$ are $\theta_n$co-hyponymys if there is a common hyperonym h betwenn $x_1, \ldots, x_n$ such as $\forall i \in [1,n]$, $dmin(x_i,h) \leq \theta$, $dmin(x,y)$ being the shortest path between two comparable terms for the relation $dhypo$. We call h a $\theta_n$minor of $x_1, \ldots, x_n$. Here follow some properties of the relation $\theta_n cohypo$:

- *symmetric.*

We can notice that $\theta_n cohypo(x_1, \ldots, x_n) \rightarrow \varphi cohypo(x_1, \ldots, x_n), \varphi \geq \theta$. Moreover, if n terms are $\theta_n$co-hyponyms, then any set of cardinality larger than those n terms, contains also $\theta_n$co-hyponyms.

A graphical summary between the lexical relations are displayed in 2

4) Meronymy: A term $t1$ is a **meronym** of $t2$ if $t2$ is semantically of part of $t1$. For example, *wheel* is a meronym of *car* under the context of transports. Here follow some properties of the relation $mero$:

- *transitive* (car, wheel, rim),

5) Holonymy: A term $t1$ is a **holonym** of $t2$ if $t1$ contains semantically $t2$. For example, *body* is an holonym of *arm* under the context of anatomy. Here follow some properties of the relation $holo$:

- *transitive* (finger, hand, arm),

$mero$ and $holo$ are inverse relations, in effect if $mero(t1,t2)$, then $holo(t2,t1)$.

D. Composing relations

To summarize, we have now:

- One equivalence relation $T|c$ (syn),
- Two strict partial order relations $T|c$ ($hyper$, $hypo$),
- One symmetric relation $T|c$ ($\theta cohypo$),
- Two only transitive relations $T|c$ ($mero$ and $holo$).

Software designers have been interested in investigating if possible combinations of relations might occur, as a path to relate two items in their design. Therefore, we have worked on the properties of relation composition. More precisely, as syn is a equivalence relation on $T|c$ it is possible to define equivalence classes on $T|c$. We write $[x]$ the equivalence class of an element $x \in T|c$. A property of an equivalence class is as follows: $\forall y \in [x], \forall z \in [x], syn(y,z)$. Thus, we obtain the following property:

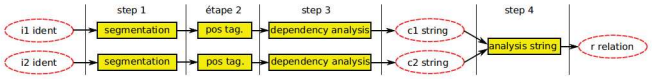


Fig. 3. Illustration of the overall process applied to identifier. At the end two identifiers could be compared for relation identification.

Relating Elements For Property Deduction

If there is a relation $r(x, y)$ (r can be any of: $hypo$, $hyper$, $dhyper$, $dhypo$, $cohypo$, $mero$ and $holo$) between x and an element y, then $\forall e \in [x], \forall f \in [y], r(e, f)$.

Here follow some examples to illustrate this property. We have two equivalence classes on the set of terms under the *anatomy* context: $(body)$ and $(skull, head)$. It is trival that $(body)$ is an equivalence class, and we do have $syn(skull, head)$. As we have moreover $mero(body, head)$, from the previous property, we deduce $mero(body, skull)$.

E. Importing Lexical Relations

In order to make good use of the rules we have defined above, it is necessary to construct several *initial relation occurences* between the identifiers of a model. In particular, this is necessary for initial syn and $dhypo$ occurences. In fact, other relations occurrences (for $dhyper$, $hyper$, $hypo$, $cohypo$) can be deduced from the former ones. Thus, we consider the initial set of occurrences of syn and $dhypo$ as a starting point under the set of identifiers. In practice, this set is available in any general use lexical network.

What follows is the description of some processing aiming at discovering lexical relations occurrences (synonymy, hyponymy, ...) between identifiers present in models. We define the $relids$ application.

- Let $T|c$ be the set of terms with a context c,
- Let $LEX = \{SYN, HYPO, HYPER, COHYPO\} \cup \emptyset$ be the set of the name of the lexical relations,

The application $relids$ is defined as follows:

$$relids : T|c \times T|c \rightarrow LEX \quad (1)$$

III. Extracting relations from identifiers: tools and rules

In order to compute the result of the application of $relids$ on an element $T|c \times T|c$, we setup the following process (figure 3) :

1) Segmentation: identifiers, which are generally a concatenation of terms, are cut into atomic terms (segments),
2) Labelling: a part of speech (POS) is given to each segment,
3) Dependency analysis: each segment is organized according to a dependency relation (which one is the head?),
4) Relation identification: the strings related to identifiers are analysed in order to determine the lexical relation between them.

A. Segmentation

In design (conceptual) model as well as in implementation models, the various elements (classes, attributes, operations, parameters, application programming interface, ...) are usually named with identifiers. As those identifiers are chosen by designers and/or programmers, they constitute (quite often) some clear information regarding the understanding of the model. It is largely accepted that identifiers of a given model are supposed to be easily associated (for a human being at least) to concepts or terms of the real world. Considering the naming restriction imposed on identifiers by programming languages or model syntax (for example, C or Java languages or UML), identifiers are often concatenations of various words, acronyms or word shortenings and abbreviations (like *getnextwarning*, *getimplementation* or *getimpl*). Those words are most of the time put into emphasis with a variation of case (upper and dowcase like in *getNextWarning*) or separators (*get_next_warning*).

The procedure of segmentation of an identifier into elements is thus the natural first step in order to retrieve possible lexical chains and match them with terms.

- Let $T|c$ be the set of valid terms under the context c,
- Let $W|c$ be the set of valid terms ($W|c \subseteq T|c$),

$$seg : T|c \rightarrow \bigcup_{i=1}^{\infty} (W|c)^i \quad (2)$$

Segmentation relies on clues. Here follows the different types of clues that can specify a segment end inside a given identifier:

- A case change (uppercase dans downcase) (*getNextWarning*),
- A separator (*get_next_warning*),
- Some numercial characters (*Block12*).

It occurs frequently that clues of several types appear for the same identifier (*stdlib_GetWarning123*). To take into account this fact, we specified several segmentation strategies, which are therafter combined into a global strategy. What follows is the description of the various strategies.

1) Separator based strategy: A separator list is given (term tagged as separator in the lexical network). An identifier is read from left to right character by character. As soon a separator is encountered, a segment is created (*date_label* becomes *(date,label)*).

2) Numerical character based strategy: We separate numerical characters from alphabetical ones (*Block122* becomes *(Block,122)*).

3) Case strategy: Here, the variation between uppercase and downcase in a given identifier is used as a clue for segmentation. For example, we have:

- *nextThing*: next, Thing.
- *ClassLoader*: Class, Loader.
- *RCAProcess*: RCA, Process.

It should be noted that it is possible to find the following type of segmentation *RCAprocess* (RCA, process), but this case is less common than the case *RCAProcess*. With a dictionary, it is possible to address this issue. The dictionary is by itself (a part of) the lexical network.

4) Dictionary based strategy: The dictionary based strategy is used in case there is no clue usable for segmentation with the previous strategies, like for example with `studentaddress`. This segmentation needs a dictionary to be effective, the dictionary being in fact some the terms contained in the lexical network (which has been added at bootstrap time with the *aspell* dictionary [Aspell 2008]). Segmenting with a dictionary is based on a prefix and a suffix approches.

Segmentation with prefixes. The identifier is segmented from left to right, reading the string character at a time. We extract the longuest existing substring. Example of segmentation of `studentadress`:

- string: `studentaddress`,
  - `s` exist? yes,
  - `st` exist? yes,
  - `stu` exist? no,
  - `stud` exist? yes,
  - `stude...` exist? no,
- First segment found `student`,
- string left `address`,
  - `a` exist? yes,
  - `ad` exist? yes,
  - `add` exist? yes,
  - `addr` exist? yes,
  - `addre` exist? no,
  - `addres` exist? no,
  - `address` exist? yes,
- Second segment found `address`,
- string left: empty
- Result: (student, address).

Some example of segmentations:

- `localname`: (local, name),
- `trytounderstandthat`: (try, to, understand, that),
- `taxis`: (tax,is).

In the last example, the proper segmentation is in fact (t,axis) because *t* is in this just a prefix of the identifier name. This strategy is not suitable when strings to be segmented are prefixed. To address this issue, we propose a suffix segmentation.

Segmentation with suffixes. This time, we scan the identifier from right to left finding longest suffixes. With the previous example, we obtain with the suffix segmentation the following results:

- `localname`: (local, name),
- `trytounderstandthat`: (try, to, understand, that),
- `taxis`: (t,axis).

Double segmentation. We combine both segmentations by choosing the result with the fewer number of segments. This is not an exact procedure, but in practice for identifiers, it is quite reliable.

B. Labelling

We aim at attaching a POS to each segment, for example for *get, next, warning* we have *verb, adj, noun*. We use *Tree Tagger* by H. Schmid ([Schmid 1994]) for this purpose. Here follows the definition of the tag function:

- Let $T|c$ be the set of valid identifiers,
- Let $W|c$ be the set of valid segments ($W|c \subseteq T|c$),
- Let POS be the set of POS,

$$tag : \bigcup_{i=1}^{\infty}(W|c)^i \rightarrow \bigcup_{i=1}^{\infty}(W|c \times POS)^i \qquad (3)$$

For example: tag: $tag((get, Next, Warning)) = (get, Verb)(Next, Adj)(Warning, Noun))$.

C. Dependency analysis

In practice, we used a simplified set of POS compared to those definied in Tree Tagger[Schmid 1994] for English:

- $Noun = NN \cup NNS \cup NP \cup NPS$,
- $Verb = VV \cup VVP \cup VVZ \cup VVG \cup VVD \cup VVN \cup VB \cup VBP \cup VBZ \cup VBG \cup VBD \cup VBN \cup VH \cup VHP \cup VHZ \cup VHG \cup VHD \cup VHN$,
- $Adj = JJ \cup JJR$,
- $Prep = IN \cup TO$,

After the labelling, we have a list of pairs $(segment, pos)$ for each indentifier. For example, for *getNextWarning* we obtain *[(get,Verb)(next,Adj)(Warning,Noun)]*. The goal of the dependency analysis is to reorganize the segment in function of the dominating order (i.e. finding heads). For example, *[(get,Verb)(next,Adj)(Warning,Noun)]* should be reorganized as *[(get,Verb)(Warning,Noun)(next,Adj)]*. The output is then a list of pairs $(segment, pos)$ ordered by dominating order.

- Let $T|c$ be the set of valid identifiers,
- Let $W|c$ be the set of valid segments ($W|c \subseteq T|c$),
- Let POS be the set of POS,

$$dep : \bigcup_{i=1}^{\infty}(W|c \times POS)^i \rightarrow \bigcup_{i=1}^{\infty}(W|c \times POS)^i \qquad (4)$$

This procedure is based on the POS given to the various segments of the identifier. For example, for *(Verb,Noun)* most probably the verb dominates the noun (example *compute sum*, *add number*, ...). The dependency analysis is done through a rulebased expert system. Rules are ordered by priority (for example, two nouns follow each other, an adjective follows a noun,...). For each rule, an action is defined and applied if the rule activates. Rules are applied iteratively on the pair list $(segment, pos)$, until all pairs are consumed. Generally speaking, such an algorithm becomes complicated to understand as the number of rules grows, making conflicts difficult to resolve, but in the case of identifiers, a small set of rules (between 5 and 10) is enough to compute properly dependency analysis.

The description of the set of rules follows.

1) English Rules Set: Let I the initial list of pairs $(segment, pos)$ and N the new reordered list, initialized to the empty list. Rules are the following (ordered from highest to lowest priority):

1) if I has size 0, then the procedure stops.
2) if I has size 1, then the element of I is added at the end of N and deleted from I.
3) if I has size 2 and the first element is a noun and the second is not a noun, then the first element is added at the end of N and deleted from I.
4) if the first element of l is a verb, it is added at the end of N and deleted from I.
5) if the first element of l is a preposition, it is added at the end of N and deleted from I.
6) if l is composed of elements that are not prepositions, followed by a preposition, followed by anything, l is divided into 3 segments (non prepositions, the preposition, the rest); the result of the application of the rule on the first part is added at the end of N, the preposition is added in N as well as the application of the rules on the rest.
7) if the last element of l is a number, then it is moved to the begining of l.
8) (default rule) the last element of l is inserted at the end of N and deleted form l.

Here follows an example of rule application for the identifier *putPersonInNicePlace*:

1) $I = (put, Verb)(person, Noun)(in, Prep)$ $(nice, Adj)(place, Noun)$, $N = \emptyset$
2) Rule 4 activates (verb in initial position)
3) $I = (person, Noun)(in, Prep)(nice, Adj)$ $(place, Noun)$, $N = (put, Verb)$
4) Rule 6 activates (there a preposition in the middle of the list)
5) I is cut in 3 pieces: $(person, Noun)$; $(in, Prep)$ and $(nice, Adj)(place, Noun)$
6) The result of the applications of the rules on $(person, Noun)$ is added at the end of N
7) Rule 2 activates on $(person, Noun)$ (list of size 1)
8) $N = (put, Verb)(person, Noun)$
9) The preposition is added to N
10) $N = (put, Verb)(person, Noun)(in, Prep)$
11) The result of the applications of the rules on $(nice, Adj)(place, Noun)$ is added at the end of N
12) Rule activates on $(nice, Adj)(place, Noun)$ (default rule), $(place, Noun)$ is added at the end of N
13) $N = (put, Verb)(person, Noun)(in, Prep)$ $(place, Noun)$
14) There is $(nice, Adj)$ left to place, rule 2 activates
15) $N = (put, Verb)(person, Noun)(in, Prep)$ $(place, Noun)(nice, Adj)$

Let us take a second example with the identifier *JavaBlock12*:

1) $I = (Java, Noun), (Block, Noun)(12, CD)$, $N = \emptyset$
2) Rule 7 activates (number in final position). The number is moved to the front.
3) $I = (12, CD)(Java, Noun)(Block, Noun)$, $N = \emptyset$
4) Rule 8 activate (default rule). The rightmost word is moved to the end of N.
5) $I = (12, CD)(Java, Noun)$, $N = (Block, Noun)$
6) Rule 68 activates again.
7) $I = (12, CD)$, $N = (Block, Noun)(Java, Noun)$
8) Rule 2 activates
9) $N - (Block, Noun)(Java, Noun)(12, CD)$

D. Identifying lexical relations

Now, given the strings computed at the previous stage, we try to identify if there is a proper lexical relation between two strings.

- Let $T|c$ be the set of valied identifiers,
- Let $W|c$ be the set of valid segments ($W|c \subseteq T|c$),
- Let POS be the set of POS,
- Let $LEX = \{SYN, HYPO, HYPER, COHYPO, MERO, HOLO\} \cup \emptyset$ the set of lexical relation types,

.

$$rel : (\bigcup_{i=1}^{\infty}(W|c \times POS)^i) \times (\bigcup_{i=1}^{\infty}(W|c \times POS)^i) \rightarrow LEX \quad (5)$$

This procedure looks for correspondences between two strings c_1 and c_2. If a correspondance is detected, the name of the lexical relation is returned, otherwise $\emptyset$ is returned. Results of the procedure depend on the set $W|c$. We consider here that this set is defined, and that some occurences of relations do exists (on syn, $hypo$, $hyper$, $mero$ and $holo$). We have:

$$c_1 = [(w_1^1, pos_1^1), (w_2^1, pos_2^1), \ldots, (w_{n_1}^1, pos_{n_1}^1)] \quad (6)$$

$$c_2 = [(w_1^2, pos_1^2), (w_2^2, pos_2^2), \ldots, (w_{n_2}^2, pos_{n_2}^2)] \quad (7)$$

Let be len the function asosciating to a string its length (for example, $len(c_1) = n_1$ and $len(c_2) = n_2$):

$$len : (\bigcup_{i=1}^{\infty}(W|c \times POS)^i) \times (\bigcup_{i=1}^{\infty}(W|c \times POS)^i) \rightarrow N \quad (8)$$

We should remind here that strings are composed of the various segments composing a given identifier, segments being ordered by importance. The discovery of a relation is done in two steps: first, is the lookup of the longest prefix pc_1c_2 between c_1 and c_2.

$$pc_1c_2 = [(w_1^{pcc}, pos_1^{pcc}), (w_2^{scc}, pos_2^{pcc}), \ldots, (w_s^{pcc}, pos_s^{pcc})] \quad (9)$$

such that

$$\forall i \in [1, s], syn(w_i^1, w_i^2) \quad (10)$$

We have then $len(pc_1c_2) = pcc$. Now, for the second step, 4 tests are done to identify the proper lexical relation. The relation type returned corresponds to the first test that passes.

1) if $len(c_1) = len(c_2) = len(sc_1c_2)$, then SYN is returned.
2) if $len(c_1) = len(pc_1c_2)$ and $len(c_1) > 0$, then $HYPER$ is returned.
3) if $len(c_2) = len(pc_1c_2)$ and $len(c_2) > 0$, then $HYPO$ is returned.
4) if $len(c_1) \neq len(pc_1c_2)$ and $len(c_2) \neq len(pc_1c_2)$ and $len(pc_1c_2) > 0$, then $COHYP$ is returned.
5) $\emptyset$ is returned.

Here follow some examples of lexical relation identification:

- Let $c1 = [(car, Noun)]$ and $c2 = [(auto, Noun)]$. Morevover, $syn(car, auto)$ is defined in $W|c$. in that case, $pc_1c_2 = [(car, Noun)]$, because of $syn(car, auto)$ (otherwise $pc_1c_2 = \emptyset$). This fullfills condition 1 and SYN is returned.
- Now, suppose we have $c1 = [(car, Noun)(big, Adj)]$ and $c2 = [(auto, Noun)]$ with $W|c$ as previously. We still have $pc_1c_2 = [(car, Noun)]$. But this time, condition 3 fullfills, thus $HYPO$ is returned.
- Finaly, let be $c1 = [(car, Noun)(big, Adj)]$ and $c2 = [(auto, Noun)(little, Adj)]$ with $W|c$ as previously. We still have $pc_1c_2 = [(car, Noun)]$. But this time, condition 4 fullfills, thus $COHYP$ is returned.

IV. Experiment and Results

We ran our system on a set with thousands of identifiers from various models and software packages (see table I). Those are real models and code (as open software) freely available on the web. We evaluated over 400 identifiers taken randomly by manually executing the chain of processes (segmentation, labelling, dependency analysis, and relation identification).

A. Results for segmentation and labelling

394 identifiers out of 400 were well segmented (0.985 ratio) – some examples have been given previously. 356 identifiers out of 400 were well labelled (0.89 ratio). 48 identifiers well segmented got at least one wrong label. For example, the identifier *ParseResult* got a verb label for *Parse* which is linguistically correct, but the clearly intended meaning was the noun and should have been *ParsingResult*. Such, linguistically inconsistent formation of identifier is in fact quite common.

B. Results for dependency analysis

356 identifiers out of 400 got a proper dependency analysis (0.89 ratio). All well labelled identifiers were correctly analysed.

C. Results for relation identification

We run the relation identification on an *a priori general* context, until we extracted around 200 relations. As picking up randomly two identifers is too time consuming and ineficient, the process was to select one identifier randomly and check for relations all other identifiers having at least one element (from segmentation) in common. in order to get around 200 relation it took a bit more than 4000 tries, which means that less than 5 percent of identifiers having one element in commun may have an insghtfull relation between them. We got an 0.84 ratio for proper relations (168 out of 200). Failure cases are typical of wrong semantic relations, valid in the general cases, but invalid for *computing* context. For example, *getThreadId* and *getStringId* were found synonyms because in the general context *string* and *thread* can be synonyms. In a more specific context, both identifier wouldn't have been identified as synonyms. Interesting and typical results follow (other actual results have been given as examples previsouly):

- LevelImpl syn LevelImplementation because Impl syn Implementation
- (OneArgumentOptionHandler, ShortOptionHandler, MapOptionHandler) $hypo$ OptionHandler $hypo$ Handler
- OneArgumentOptionHandler $cophyp$ ShortOptionHandler $cophyp$ MapOptionHandler
- (ColorStringParser, ShortStringParser) $hypo$ StringParser
- ColorStringParser $cophyp$ ShortStringParser
- getMeaning syn getSense
- getValueList $hyper$ getNumberList as value $hyper$ number
- getBigInteger syn getLargeInt
- ExpandCharArr syn ExpandCharArray syn ExpandCharacterArray
- isErrorLogged syn isErrorConnected

V. Conclusion

Automating the discovery of mappings between schemas, ontologies, documents or models has been thoroughly investigated [Rahm and Bernstein 2001], [Shvaiko 2005]. In the context of Model-Driven Engineering, several approaches for semi-automatic generation of transformations based on mapping have recently been proposed. Mixing NLP techniques and model specification has also been a track followed by some works([Liu et al. 2004] [Ilieva and Ormandjeva 2005]). As for model transformation generation, in [Roser and Bauer 2006], model transformations are generated based on ontological information, but less frequently on lexical ones. The two models are supposed to have their semantics provided by a mapping onto a known ontology. Reasoning on the ontology then allows to generate a model transformation, adapting a bootstrap transformation that is whether automatically generated or existing. When dealing with a lexical network as we do in this paper, the same features apply to both models, and we take advantage to do it jointly.

Working on names, or more generally on identifiers, is an issue for MDE [Caprile et Tonella 1999] [Lawrie et al., 2006] In this paper, we have first modelled possible and useful relations between identifiers in models, inspired from lexical semantics in NLP, with a contextual orientation, and an opportunity to compose relations for model transformation generation. We have thus presented some approaches that may be combine to extract relations form identifiers, by using POS tagging (in English, using Tree Tagger) to retrieve words functions in compound identifiers, and then obvious dependency rules to assign a government role to a given

item. The role of each item is crucial in order to assert its position in the hierarchy of identifiers, and to detect relations between words. Naturally, dependency rules are shaped for English since most programming names are english based denominations for attributes, classes or variables.

Experiments conducted so far are very promising and clearly show the benefit that can be leveraged from introducing NLP techniques in the domaine of UML modelling. Possible ongoing tracks in NLP for this research could be the use of mutual information approach (like LSA) applied on models and programs in order to access terms sharing the same context and possibly revealing some not so trival relations between them; this approach has been successfully applied to texts.

ACKNOWLEDGMENT

The authors would like to thank France Telecom R&D (Orange Labs) for their support. (CPRE 5326).

REFERENCES

[Aspell 2008] Aspell (2008). Aspell. http ://aspell.net.

[Budanitsky and Hirst 2006] A. Budanitsky and G. Hirst, Evaluating WordNet-based Measures of Lexical Semantic Relatedness, *Computational Linguistics*, vol. 32, no. 1, pp. 1347, 2006.

[Caprile et Tonella 1999] Caprile, B. et Tonella, P. (1999). Nomen est omen : Analyzing the language of function identifiers. In *WCRE*, pages 112-122.

[Duchateau et al., 2007] Duchateau F., Bellahsene Z., Roantree M., Roche M. An Indexing Structure for Automatic Schema Matching *SMDB-ICDE'07: International Workshop on Self-Managing Database Systems*, (2007)

[Falleri et al. 2007] Falleri, J.-R., Arevalo, G., Huchard, M. et Nebut, C. (2007a). *Rapport de tâche 1 du projet ftidm*. http ://www.lirmm.fr/falleri/ftidm/-data/rapports/rapportTache1.pdf.

[Falleri et al. 2007] Falleri, J.-R., Arevalo, G., Huchard, M. et Nebut, C. (2007b). *Rapport de tâche 2 du projet ftidm.* http ://www.lirmm.fr/ falleri/ftidm/- data/rapports/rapportTache2.pdf.

[Falleri et al. 2008] Falleri, J.-R., Lafourcade, M., Prince, V.,Huchard, M. et Nebut, C. (2008a). *Rapport de tâche 3 du projet ftidm.* http ://www.lirmm.fr/ falleri/ ftidm/data/rapports/rapportTache3.pdf.

[Falleri et al., 2008b] 008]Falleri08b Falleri, J.-R., Lafourcade, M., Prince, V., Huchard, M. et Nebut, C. (2008b). *Rapport de tâche 4 du projet ftidm.* http ://www.lirmm.fr/ falleri/ftidm/data/rapports/rapportTache4.pdf.

[Fan et al. 2007] Jung-Wei Fan, Hua Xu and Carol Friedman Using contextual and lexical features to restructure and validate the classification of biomedical concepts *BMC Bioinformatics* 2007, 8:264

[Ilieva and Ormandjeva 2005] M. Ilieva and O. Ormandjieva, Automatic Transition of Natural Language Software Requirements Specication into Formal Presentation, in Natural Language Processing and Information Systems, ser. Lecture Notes in Computer Science, vol. 3513. Springer Berlin / Heidelberg, 2005, pp. 392397.

[Lawrie et al., 2006] Lawrie, D., Morrell, C., Feild, H. et Binkley, D. (2006). Whats in a name ? a study of identifiers. In *ICPC*, pages 3-12. IEEE Computer Society.

[Liu et al. 2004] D. Liu, K. Subramaniam, A. Eberlein, and B. H. Far, Natural Language Requirements Analysis and Class Model Generation Using UCDA, in *Innovations in Applied Articial Intelligence*, ser. Lecture Notes in Computer Science, vol. 3029. Springer Berlin / Heidelberg, 2004, pp. 295304. [Online]. Available: http://www.springerlink.com/ content/peeghrjy5kmkfrpc

[Miller 1994] Miller, G. A. (1994). Wordnet : A lexical database for english. In HLT. Morgan Kaufmann.

[Montes y Gomez et al., 2007] M. Montes y Gomez, A. Gelbukh, A. Lopez Lopez, R. Baeza-Yates. Flexible Comparison of Conceptual Graphs. Lecture Notes in Computer Science, N 2113, Springer, 2001, pp. 102-111;

[Rahm and Bernstein 2001] Rahm, E., and P. A. Bernstein: A Survey of Approaches to Automatic Schema Matching. VLDB Journal 10(4), 2001.

[Roser and Bauer 2006] B. Roser and S.,Bauer. "An Approach to Automatically Generated Model Transformation Using Ontology Engineering Space *Proceedings of SWESE(2nd International Workshop on Semantic Web Enabled Software Engineering*) 2006.

Prog	Classes	Itf	Att	Methods
JSON	172	69	1092	3748
SableCC	358	78	1502	5602
JavaCC	280	74	1943	5364
OpenCloud	187	94	1182	4102
Salome TMF	483	252	4715	11058
JUnit	251	89	1138	4124
NgramJ	244	113	1453	4518
JWNL	269	130	1744	5761
SimMetrics	261	78	1171	4076
Commons CLI	184	69	1153	3877
Args4J	187	72	1143	3812
JSAP	452	183	4236	8949
Choco	546	170	2301	8324
Colt	784	280	5524	13826
JGA	721	266	5638	13554
JScience	342	170	2419	6780
JSci	301	95	1471	5620
Commons Math	340	122	1830	5570
Lucene	473	108	2785	6996
JCommon	679	267	5567	12685
XOM	421	189	1784	6312
Julia	491	211	2364	6513

TABLE I
THE CORPUS OF MODELS AND PACKAGES USED IN OUR EXPERIMENT

[Schmid 1994] Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing,* volume 12. Manchester, UK.

[Shvaiko 2005] Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: J. Data Semantics IV, Volume 3730 of LNCS. (2005) 146-171

[Vossen 1998] Vossen, P. (1998). Eurowordnet a multilingual database with lexical semantic networks. *Computational Linguistics,* 25(4).

Évaluation et consolidation d'un réseau lexical via un outil pour retrouver le mot sur le bout de la langue

Alain Joubert (1), Mathieu Lafourcade (1), Didier Schwab (2), Michael Zock (3)

(1) LIRMM, Université Montpellier II (2) LIG, Université Grenoble II (3) LIF-CNRS, Marseille
{alain.joubert, mathieu.lafourcade}@lirmm.fr, didier.schwab@imag.fr, michael.zock@lif.univ-mrs.fr

Résumé Depuis septembre 2007, un réseau lexical de grande taille pour le Français est en cours de construction à l'aide de méthodes fondées sur des formes de consensus populaire obtenu via des jeux (projet JeuxDeMots). L'intervention d'experts humains est marginale en ce qu'elle représente moins de 0,5% des relations du réseau et se limite à des corrections, à des ajustements ainsi qu'à la validation des sens de termes. Pour évaluer la qualité de cette ressource construite par des participants de jeu (utilisateurs non experts) nous adoptons une démarche similaire à celle de sa construction, à savoir, la ressource doit être validée sur un vocabulaire de classe ouverte, par des non-experts, de façon stable (persistante dans le temps). Pour ce faire, nous proposons de vérifier si notre ressource est capable de servir de support à la résolution du problème nommé 'Mot sur le Bout de la Langue' (MBL). A l'instar de JeuxdeMots, l'outil développé peut être vu comme un jeu en ligne. Tout comme ce dernier, il permet d'acquérir de nouvelles relations, constituant ainsi un enrichissement de notre réseau lexical.

Abstract Since September 2007, a large scale lexical network for French is under construction through methods based on some kind of popular consensus by means of games (JeuxDeMots project). Human intervention can be considered as marginal. It is limited to corrections, adjustments and validation of the senses of terms, which amounts to less than 0,5 % of the relations in the network. To appreciate the quality of this resource built by non-expert users (players of the game), we use a similar approach to its construction. The resource must be validated by laymen, persistent in time, on open class vocabulary. We suggest to check whether our tool is able to solve the *Tip of the Tongue* (TOT) problem. Just like JeuxDeMots, our tool can be considered as an on-line game. Like the former, it allows the acquisition of new relations, enriching thus the (existing) network.

Mots-clés Réseau lexical, JeuxDeMots, évaluation, outil de MBL, mot sur le bout de la langue
Keywords Lexical network, JeuxDeMots, evaluation, TOT software, tip of the tongue

Introduction

Grâce à un nombre important de participants à des jeux en ligne (notamment JeuxDeMots et PtiClic), nous avons obtenu un réseau lexical de grande taille pour la langue française (actuellement plus de 220000 termes[1], reliés par plus d'un million de relations sémantiques) représentant une connaissance générale commune. La communauté dispose donc d'une ressource lexicale dont nous souhaitons évaluer la qualité. Une évaluation manuelle pose au moins deux problèmes : d'une part, elle peut être biaisée par les compétences de l'évaluateur, et d'autre part, elle nécessite un temps prohibitif dès que l'on souhaite effectuer une évaluation quelque peu conséquente. Nous aurions pu envisager une évaluation automatique par comparaison avec une référence, mais à notre connaissance une telle référence n'existe pas, du moins

[1] Un terme peut être constitué de plusieurs mots (par exemple : *étoile de mer*)

pour la langue française, ayant une couverture et un nombre de types de relation suffisant. L'évaluation manuelle par échantillonnage ne nous semble pas satisfaisante car elle est nécessairement trop réduite, trop ponctuelle et d'une qualité difficile à apprécier. Nous avons donc décidé d'évaluer notre ressource, via un logiciel de détermination du « mot sur le bout de la langue » (MBL), évaluation qui pourrait être faite de façon permanente et avec un grand nombre d'évaluateurs en simple aveugle (ces derniers ne sachant pas qu'ils évaluent). Notre réseau lexical représentant des connaissances générales, un tel logiciel doit s'appliquer à un domaine ouvert, à savoir du vocabulaire tout venant, y compris des termes peu courants. Compte tenu du caractère sémantique du réseau lexical, l'outil de MBL opèrera exclusivement de manière quasi-sémantique, utilisant essentiellement des associations d'idées, des relations ontologiques ou celles de typicalité. Un mode d'accès par la phonétique, voire la notion de rébus, est donc exclu.

Nous commencerons cet article en présentant d'abord la problématique du MBL, pour rappeller ensuite brièvement le processus de constitution de notre réseau lexical, avant de présenter notre outil de MBL, dénommé AKI[2]. Enfin, nous commenterons les résultats obtenus grâce à AKI pour évaluer notre réseau lexical.

1 Problématique

Difficulté de l'évaluation - Nous sommes confrontés au problème d'évaluation de donnée lexicale, où aucun standard de référence n'est disponible et où une l'évaluation manuelle n'est pas envisageable. Dans un premier temps, on serait tenté de répondre aux questions de complétude et de précision (exactitude) :

- notre réseau lexical est-il « complet », à savoir comporte-t-il « tous » les termes et « toutes » les relations ?
- n'y a-t-il pas dans notre réseau des termes ou des relations erronés ?

Bien évidemment, la réponse stricte à ces deux questions est négative, ne serait-ce qu'en raison du caractère évolutif de la langue ; par exemple, le terme *révolution de jasmin* n'est apparu qu'en janvier 2011. Cependant, nous pouvons dégager une question plus réaliste :

- pour chaque terme de notre réseau lexical, l'ensemble des relations qu'il entretient avec d'autres termes suffit-il à le caractériser de façon unique ?

Dans l'affirmative, tout terme est susceptible d'être retrouvé via un ensemble réduit de termes indices. Ceci étant, nous avons créé un outil de "mot sur le bout de la langue" (MBL) pour réaliser cette évaluation.

2 Le problème du 'mot sur le bout de la langue' (MBL)

Le terme "manque de mot" désigne à la fois l'absence de terme dans le dictionnaire mental (Aitchison, 2003) d'un locuteur, ainsi que l'incapacité de pouvoir y accéder à temps. Nous nous intéressons ici uniquement à ce dernier cas. Le manque de mot, est une situation connue par tout producteur de langue notamment à l'oral (discours spontané). Cette défaillance sera qualifiée d'anomie, d'Alzheimer, ou de *mot sur le bout de la langue* (MBL), selon la durée et la fréquence du blocage et selon la nature d'information accessible (sémantique, phonologique) au moment crucial, la production écrite ou orale.

L'expression *« avoir le mot sur le bout de la langue »* (MBL) » ou, son analogue anglais, *« it's on the tip of my tongue »* (TOT), décrivent une forme de blocage très particulier. Un locuteur cherchant à exprimer une idée est conscient de connaître le terme, il sent sa production imminente (le plus gros du travail étant accompli), pourtant, il échoue tout près de la fin. La dernière partie est inaccessible. Tel un éternuement non consommé, la forme sonore reste bloquée, et la traduction du sens en forme linguistique n'aboutit pas. Ce qui caractérise le MBL et ce qui le distingue d'autres formes de manque de mot[3] c'est que le locuteur connaît

[2] http://www.lirmm.fr/jeuxdemots/AKI.php

[3] Comme indiqué, il y a d'autres cas de figure d'échec lexical. L'un, où le locuteur ignore tout simplement le mot recherché (cas fréquent en langue étrangère), et l'autre, où il connaît le terme, mais il n'arrive pas à l'évoquer à temps : c'est le blanc ou le vide total, situation typique pour des mots rares ou très techniques. D'ailleurs, beaucoup de gens utilisent le terme de MBL de manière générique, voulant lui faire endosser tout type de manque de mot. Ceci est impropre, car il peut y avoir différentes raisons pour

le terme, qu'il en est conscient et que le terme recherché est imminent, d'où l'expression, 'sur le bout de la langue' (Brown & McNeill, 1966). Que le locuteur connaisse le terme est démontrable. Soit il le produit spontanément peu de temps après (en général dans la journée), reconnaissant par ailleurs, et souvent avec soulagement, que c'est bien le terme recherché, soit il l'identifie dans une liste, tâche qu'il effectue avec une vitesse et certitude étonnante (taux d'erreurs extrêmement faible).

D'autres particularités du MBL sont le fait que le locuteur, sait énormément de choses concernant le mot-cible bien qu'il soit incapable de le produire : fragments de *sens* ou fonctions pratiques ('cela sert à s'orienter lors d'une navigation en mer'), *informations syntaxiques* (catégorie lexicale : nom/verbe ; genre grammatical : masculin/féminin), *informations morphologiques* (type et origine de l'affixe) ; *informations phonologiques* (contour intonatif, nombre de syllabes, première et dernière syllabe).

Les informations données par les personnes se trouvant dans cet état ont souvent été utilisées par des psychologues comme argument pour construire et justifier un modèle de production lexicale. La plupart des chercheurs s'accordent pour dire qu'il y a deux étapes se succédant avec, ou sans, chevauchement (Levelt et al. 1999; Ferrand, 1998 ; mais voir également Caramazza, 1997). L'une consiste à déterminer le *lemme* (pour un sens donné on choisit une forme lexicale, qui elle est abstraite), l'autre a pour fonction de déterminer la forme concrète (forme morphologique, graphémique ou phonologique), le *lexème*. Si l'enchaînement de ces deux étapes se fait généralement en continu, donc sans interruption, des problèmes peuvent survenir. Ainsi, il se peut que la première étape se déroule correctement mais pas la seconde, auquel cas on aboutit à l'état nommé MBL : l'information sémantique et grammaticale étant disponibles intégralement, mais pas l'information graphique ou phonologique. Bien entendu, on peut aussi imaginer que l'accès sémantique soit déficient, auquel cas il est logiquement impossible d'accéder à la forme phonologique, car, à moins de ne répéter un mot, il est impossible d'avoir accès à sa forme phonologique sans en avoir déterminé le sens.

Etant donné la faiblesse de la trace phonologique on pourrait être tenté à vouloir renforcer celle-ci, et c'est bien ce que certains psychologues ont suggéré (Abrams et al., 2007). Pourtant, ce n'est pas la voie que nous allons emprunter ici et il y a pour cela plusieurs raisons.

L'analyse d'erreurs (Rossi, 2001)[4] et l'étude du phénomène du MBL suggèrent que l'accès lexical se fait par deux voies : par le sens et par la forme (notamment les sons, phonèmes). Ceci dit, l'accès par le sens (boisson-vin) n'exclut nullement l'accès par des termes associés, par exemple, le terme 'vin' activant le terme 'fromage'. Pourtant, le terme 'vin' n'est pas un élément de sens du terme 'fromage'. C'est une co-occurrence, ou si l'on préfère, c'est une association sur l'axe syntagmatique. Outre cette co-occurrence, les deux termes entretiennent une relation sémantique (ou encyclopédique : 'en mangeant du fromage on boit du vin'). Deux autres points méritant être rappelés sont l'aspect relationnel des termes (ils sont du type associatifs : un terme *x* pouvant évoquer un terme *y* avec une probabilité *z*) et leur organisation sous forme de graphe. Ces deux caractéristiques sont capitales, et elles offrent plusieurs avantages :

- le fait que des termes soient liés élargit le champ de recherche : chaque terme source (mot donné en entrée) active un ensemble de termes associés (termes cibles potentiels), ensemble susceptible de contenir le terme recherché;
- le fait que les termes soient organisés sous forme de graphe permet leur accès par différents chemins. Si cette forme de représentation introduit une certaine redondance dans la représentation des données, elle a l'immense mérite de permettre de retrouver le bon chemin, au cas où l'on se serait trompé à un certain embranchement, situation guère possible, ou du moins beaucoup plus compliquée, dans le cas d'arborescences.

---

causer cette forme de blocage. Produire un mot suppose avoir effectué des traitements à différents niveaux (conceptuels, sémantiques, phonologiques). Or, l'échec (erreur, incomplétude) à n'importe lequel de ces niveaux peut bloquer la machine et produire ce qu'on appelle *manque de mot*. Le terme MBL ne décrit qu'une situation très particulière : le blocage se situe uniquement au niveau phonologique (informations erronées, informations manquantes), les informations venant des autres niveaux étant généralement disponibles dans leur intégralité.

[4] Des erreurs comme 'à ma gauche' au lieu de 'à ma droite' et 'élision' à la place de 'illusion' illustrent ces deux voies d'accès.

En somme, lorsqu'on est en état de MBL on peut essayer de retrouver un terme via d'autres termes phonologiquement proches (accès par la forme, Abrams, 2007; Zock; 2002), mais aussi via des termes ayant un lien sémantique. Nous nous sommes intéressés ici uniquement à cette dernière solution.

2.1 Hypothèses de travail

Il semble difficile de distinguer les deux usages suivants d'une application de MBL : 1) l'utilisation comme un utilitaire afin de retrouver un terme, 2) l'usage ludique de type devinette. Les motivations pour le second usage sont variables, mais en général visent à «mettre en difficulté le système ».

Il semble *a priori* difficile de savoir si les utilisateurs abordent notre outil de façon utilitaire ou ludique. Plutôt qu'effectuer une étude approfondie sur cette question (étude sans doute longue, difficile et coûteuse), nous allons donc considérer ces deux activités comme identiques. Plus précisément, nous partons des deux hypothèses suivantes :

- Hypothèse 1 : les termes recherchés par les utilisateurs de notre jeu MBL sont des termes de fréquence moyenne ou faible (termes de difficulté[5] moyenne ou importante). Les utilisateurs sont vraiment intéressés à trouver le terme recherché.
- Hypothèse 2 : le vocabulaire ciblé est de basse et de moyenne fréquence (termes de difficulté moyenne ou importante). Les joueurs cherchent à vérifier l'efficacité de l'outil.

Etant donné que le vocabulaire qui déclenche le MBL et celui avec lequel les joueurs de MBL jouent sont identiques, cela nous amène à postuler que « *l'évaluation d'un outil de MBL peut se faire grâce à des joueurs* ».

Par ailleurs, une seconde hypothèse de travail consiste à dire que l'éventail de comportements des joueurs est comparable à celui des personnes ayant réellement besoin de retrouver un mot. A savoir, leur motivation consiste à essayer de piéger l'outil soit avec un terme simple et des indices à la marge, soit avec un terme rare, improbable, ou récent, et des indices plus directs. On peut donc raisonnablement conclure qu'une telle évaluation est plus défavorable que celle portant sur des cas réels de MBL et qu'elle caractérise une ligne basse : *l'évaluation d'un outil de MBL via un jeu fournit une valeur plancher de ses performances*.

Ce sont ces deux hypothèses que nous tenterons de vérifier dans la suite de cet article.

3 Constitution du réseau lexical

3.1 JeuxDeMots[6] : construction du réseau

Le principe de base conduisant grâce à un jeu en ligne à la construction progressive du réseau lexical, à partir d'une base de termes préexistante, a déjà été décrit par (Lafourcade et Joubert, 2009). Une partie se déroule entre deux joueurs, en double aveugle et en asynchrone. Pour un même terme cible T et une même consigne C (synonymes, domaines, association libre…), les deux joueurs proposent des termes correspondant, selon eux, à cette consigne C appliquée à ce terme T. Ces propositions sont limitées en nombre, ce qui a pour effet d'augmenter leur pertinence, mais également dans le temps pour favoriser leur caractère spontané. Nous mémorisons alors les réponses communes à ces deux joueurs[7]. Les validations sont

[5] Nous faisons également l'hypothèse que ce que nous appelons la difficulté d'un terme est contra-variante à sa fréquence, la difficulté d'un terme exprimant à la fois la difficulté à trouver des indices s'y rapportant mais également la difficulté à faire émerger ce terme chez un interlocuteur.

[6] http://jeuxdemots.org

[7] La limitation dans le temps de la saisie des propositions des joueurs peut favoriser les fautes d'orthographe, mais, comme nous ne mémorisons que les réponses communes aux deux joueurs d'une même partie, l'expérience montre que ce risque est très limité et que seules subsistent les fautes d'orthographes qui de toutes façons auraient été faites par les joueurs (par exemple : *beau* pour *bot*, en parlant de *pied*).

donc faites par concordance des propositions entre paires de joueurs pour un même couple (C,T). Ce processus de validation rappelle celui utilisé par (von Ahn et Dabbish, 2004) pour l'indexation d'images ou plus récemment par (Lieberman et al., 2007) pour la collecte de « connaissances de bon sens ». À notre connaissance, il n'avait jamais été mis en œuvre dans le domaine de la construction des réseaux lexicaux.

La structure du réseau lexical ainsi obtenu s'appuie sur les notions de nœuds et de relations entre nœuds, selon un modèle initialement présenté par (Collins et Quillian, 1969) et davantage explicité par (Polguère, 2006). Chaque nœud du réseau est constitué d'une unité lexicale (terme, raffinement d'un terme ou segment textuel) liée aux autres termes via des relations des fonctions lexicales, telles que présentées par (Mel'čuk et al., 1995). Les relations obtenues grâce à l'activité des joueurs sont typées et pondérées[8] : elles sont typées par la consigne imposée aux joueurs, elles sont pondérées en fonction du nombre de paires de joueurs qui les ont proposées, comme indiqué dans (Lafourcade et Joubert, 2009). Plusieurs exemples de relations acquises ont été donnés dans (Lafourcade et Joubert, 2009). Au moment du lancement de JeuxDeMots en juillet 2007, le réseau comportait 152 000 termes (non reliés entre eux, c'est-à-dire aucune relation n'existait). Courant mars 2011, à l'issue d'environ 900 000 parties jouées par plus de 2500 joueurs, notre réseau compte 229 000 termes et plus de 1 100 000 relations.

3.2 PtiClic : consolidation du réseau

De manière analogue à JeuxDeMots (JDM), une partie de PtiClic (http://pticlic.org) se déroule, en double aveugle et asynchrone, entre deux joueurs. Un premier joueur se voit proposer un terme cible T, origine de relations, ainsi qu'un nuage de mots provenant de l'ensemble des termes reliés à T dans le réseau lexical produit par JDM. Plusieurs consignes correspondant à des types de relations sont également affichées. Le joueur associe, par cliquer-glisser, des mots du nuage aux consignes auxquelles il pense qu'ils correspondent. Ce même terme T, ainsi que le même nuage de mots et les mêmes consignes, sont également proposés à un deuxième joueur. Selon un principe analogue à celui mis en place pour JDM, seules les propositions communes aux deux joueurs sont prises en compte, renforçant ainsi les relations du réseau lexical.

Contrairement à JDM, PtiClic est un jeu fermé où les utilisateurs ne peuvent pas proposer de nouveaux termes, mais sont contraints de choisir parmi ceux affichés. Ce choix de conception a pour but de réduire le bruit dû aux termes mal orthographiés ou aux confusions de sens. PtiClic réalise donc une consolidation des relations produites par JDM et permet de densifier le réseau lexical. Notons également que PtiClic permet de créer de nouvelles relations entre termes précédemment reliés par au moins une relation d'un autre type (même si ce n'est pas l'objectif premier de ce logiciel). Afin de réduire le silence correspondant aux termes non proposés par les utilisateurs de JDM, (Zampa et Lafourcade, 2009) ont suggéré de générer le nuage de mots à l'aide de la LSA, en utilisant un corpus externe de grand volume (l'expérimentation réalisée utilise un corpus comportant une année du journal « Le Monde ».). Cette solution permet d'augmenter le réseau lexical par ajout de nouvelles relations, en proposant aux joueurs de nouveaux termes cibles sans liens à T dans le réseau.

3.3 Raffinement des termes : enrichissement du réseau

Le processus permettant d'aboutir aux raffinements de termes est décrit dans (Lafourcade et Joubert, 2010). Nous avons fait l'hypothèse que les sens d'usage, plus communément appelés usages, d'un terme correspondent dans le réseau aux différentes cliques auxquelles ce terme appartient. Cette approche est analogue à celle développée par (Ploux et Victorri, 1998) à partir de dictionnaires de synonymes. En calculant la similarité entre les différentes cliques d'un même terme, nous pouvons construire son arbre des usages nommés. La racine de l'arbre regroupe tous les sens de ce terme. Plus on s'éloigne de la racine, c'est-à-dire plus la profondeur des nœuds dans l'arbre est importante, plus on rencontre des distinctions fines d'usages. Les nœuds de profondeur 1 dans cet arbre correspondent généralement aux différents sens de ce terme répertoriés dans les dictionnaires traditionnels. Après un processus de validation par un expert

[8] Une relation peut donc être considérée comme un quadruplet : terme source, terme cible, type et poids de la relation. Entre deux mêmes termes, plusieurs relations de types (et de poids) différents peuvent exister.

lexicographe de ces différents sens, nous les intégrons dans le réseau en tant que nœuds de raffinement du terme considéré ; le réseau est ainsi enrichi de nouveaux nœuds à partir desquels ou vers lesquels les joueurs de JDM peuvent créer des relations. Actuellement, sur les 229 000 termes connus par le réseau, près de 5 000 ont été raffinés.

4 Un algorithme et un outil de MBL

AKI est un outil d'accès lexical accessible sur le Web à partir du portail JeuxDeMots ou directement à http://www.lirmm.fr/jeuxdemots/AKI.php. AKI peut être envisagé comme un jeu : l'utilisateur fait ″deviner″ un terme cible à l'ordinateur (espérant, éventuellement, de manière secrète, de mettre en défaut sa capacité à trouver un terme à partir d'indices). AKI peut également être considéré comme une assistance, pour retrouver un terme qu'on a sur le bout de la langue. L'utilisateur est invité à fournir, un par un, une succession de termes indices qui lui paraissent pertinents pour trouver le terme cible recherché. Ce mécanisme est comparable à celui de certains jeux télévisés. Après chacun de ces termes indices AKI fait une proposition. Si elle correspond au terme recherché, l'utilisateur valide la proposition, sinon il introduit un nouvel indice. Ce dialogue se poursuit jusqu'à ce que l'une des deux situations se produise : AKI trouve le terme cible ou il abandonne et demande à l'utilisateur de fournir la solution.

4.1 AKI : principe et réalisation

L'utilisateur saisit un premier terme indice i_1. En utilisant le réseau lexical, l'algorithme calcule la signature lexicale de i_1 : $S(i_1) = S_1 = t_1, t_2, \ldots$ où les t_i sont triés par activation décroissante. Autrement dit, t_1 est le terme pour lequel la somme des poids des relations le liant à i_1 est la plus élevée. La première proposition d'AKI, p_1, correspond à ce terme : $p_1 = t_1$. Si c'est le terme cible, l'utilisateur le valide et la partie est terminée, sinon, il est retiré de la signature S_1, ainsi que i_1 (qui ne peut pas être le terme cible). Donc, à ce stade, la signature courante est : $S'_1 = S_1 - \{p_1, i_1\}$. L'utilisateur est alors invité à saisir un deuxième terme indice i_2. L'algorithme calcule une deuxième signature lexicale par intersection entre la signature courante et celle de i_2 : $S_2 = (S'_1 \cap S(i_2)) - i_2$.

Le terme proposé p_2 est celui de S_2 dont l'activation est la plus forte. Autrement dit, parmi les termes reliés à la fois à i_1 et à i_2, AKI affiche celui pour lequel la somme des poids des relations le reliant à i_1 et à i_2 est la plus élevée, exception faite des termes déjà proposés par AKI pour cette partie (ainsi que des termes indices !). La signature courante est alors $S'_2 = S_2 - p_2$. D'une façon générale, à l'étape *n*, nous avons :

$$S_n = (S'_{n-1} \cap S(i_n)) - i_n \quad \text{et} \quad S'_n = S_n - p_n$$

où i_n est le n-ième indice fourni par l'utilisateur et p_n la n-ième proposition de AKI. Le nombre de termes constituant la signature diminue donc au fur et à mesure de l'insertion d'indices. Il est fréquent que la signature devienne vide, avant même que le terme cible n'ait été trouvé ; dans ce cas là, AKI ne peut plus proposer de terme. Le processus pourrait s'arrêter là, mais afin d'améliorer le taux de rappel, une ″procédure de rattrapage″ est mise en œuvre : au lieu d'effectuer des intersections de signatures, on utilise leur somme :

$$S_n = (S'_{n-1} + S(i_n)) - i_n \quad \text{et} \quad S'_n = S_n - p_n$$

Cette procédure favorise l'apprentissage en créant des relations entre des termes isolés. Aussi utile soit elle, cette astuce doit néanmoins être utilisée avec précaution. En effet, au-delà de quelques itérations, le nombre de termes constituant la nouvelle signature devient vite prohibitif ; notre expérience tend à montrer qu'il ne faut pas dépasser deux itérations. Au delà, l'algorithme donne des propositions trop éloignées des mots proposés. Le processus se termine lorsque AKI a trouvé le terme cible recherché, ou lorsque la signature lexicale courante devient vide (ce qui est relativement rare, compte tenu de la procédure de rattrapage). À partir de 5 indices (cette limite de 5 étant un paramètre modifiable) l'utilisateur a la possibilité d'abandonner en indiquant à AKI qu'il fait fausse route. En effet, nous avons estimé que si, au bout de 5 indices, AKI n'a pas trouvé le terme recherché, cela signifie probablement que ces indices ne sont pas pertinents. La figure 1 présente quelques exemples de parties.

Les utilisateurs d'AKI ont la possibilité de faire précéder leur indice de mot-clé faisant référence à des fonctions lexico-sémantiques. Actuellement il est possible d'utiliser dix fonctions : hyperonymie, hyponymie, synonymie, antonymie, domaine, matière, lieu (lieu typique où l'on peut trouver ce que l'on cherche), caractéristique, holonymie, méronymie. Elles correspondent toutes à un type de relation existant dans le réseau JeuxDeMots.

Figure 1 : Quelques exemples de parties. Dans les trois premiers cas, AKI a trouvé le terme cible. Dans le quatrième cas, ne pouvant plus faire de proposition, AKI a abandonné et l'utilisateur a saisi la bonne réponse (il s'agissait donc d'une utilisation ludique de AKI).

4.2 Consolidation du réseau

Dans l'hypothèse jeu, quand le terme cible n'a pas été trouvé par AKI, l'utilisateur est invité à le saisir. Il y a alors création dans le réseau lexical de relations typées « AKI » avec un poids très faible (+1 à chaque occurrence). Ces relations sont régulièrement vérifiées et validées (ou non) par un expert lexicographe. En effet, dans la mesure où c'est l'utilisateur lui-même qui choisit le terme cible, ainsi que les termes indices, la sécurité de la pertinence de telles relations peut difficilement être garantie (un joueur peut toujours commettre des erreurs, sans parler d'éventuels joueurs malveillants). Ceci est différent de JeuxDeMots où les relations sont créées par intersections de propositions de joueurs. Ici, il arrive fréquemment qu'un utilisateur joue plusieurs fois le même terme, avec des indices différents mais également avec des indices communs. Nous sommes en train de réfléchir comment sécuriser ces relations typées « AKI ».

5 Evaluation du réseau via AKI

Évaluation informelle - Nous avons mené une évaluation informelle des performances d'AKI à partir du jeu de société *Tabou* inversé. Le principe du jeu de société *Tabou* est de faire deviner un terme à des personnes à l'aide d'indices, en excluant certains termes dits tabous. La version commerciale de ce jeu fournit une collection de 500 fiches comprenant chacune un terme cible et 5 termes tabous. La version inversée de ce jeu consiste à faire deviner le mot cible en énumérant ces termes tabous, l'hypothèse étant que ces termes sont particulièrement évocateurs du terme cible lorsqu'ils apparaissent ensemble.

Ceci étant, nous avons soumis cette collection de 500 fiches à AKI ainsi qu'à trois personnes (à des fins de comparaisons). AKI a retrouvé le terme cible au plus tard au bout des 5 indices dans 494 cas (soit 98,8% de réussite). Les personnes prises comme références, ont globalement trouvé (dans les mêmes conditions) 402 fois (soit 80,4% de réussite). Ce dernier chiffre n'est qu'une indication, vue la faible taille de l'échantillon considéré, de trois participants.

5.1 Protocole

L'évaluation, tout comme l'apprentissage, ne se fait qu'en fonction de ce que les joueurs ont renseigné. Elle se fait donc sur du vocabulaire appartenant à la classe ouverte. Comme déjà mentionné, AKI peut être envisagé comme un jeu ou un outil de MBL. *A priori*, notre logiciel ne sait pas faire la distinction entre les deux usages. En effet, sur une seule partie et si AKI trouve la solution, nous ne pouvons pas savoir a priori si l'utilisateur connaissait ou s'il recherchait le terme cible. Par contre, si dans un laps de temps relativement court (de l'ordre de quelques minutes) un même terme est joué plusieurs fois, nous pouvons faire l'hypothèse qu'il s'agit d'une utilisation de type jeu (au moins à partir de la deuxième partie) où l'utilisateur essaie de faire trouver le terme cible par AKI, en proposant généralement des indices différents. Dans chacun des deux cas, jeu ou outil de MBL, les termes cibles sont majoritairement des termes de fréquence moyenne, voire faible. En effet, jouer pour trouver un mot fréquent ne présente pas un grand intérêt, et généralement on ne recherche pas grâce à un outil de MBL un terme courant. Les graphiques ci-après montrent l'évolution dans le temps du rapport entre le nombre de parties gagnées par AKI, parties où l'utilisateur a indiqué que le logiciel a trouvé le terme cible, et le nombre de parties jouées. Les graphiques de cette section reflètent 6522 parties réalisées entre le 30/12/2010 et le 24/01/2011.

5.2 Analyse quantitative et évolution dans le temps

Le premier graphique (figure 2) présente l'évolution du rapport entre le nombre de parties gagnées et le nombre de parties jouées par fenêtre glissante de 500. Par exemple, à l'abscisse 100 (correspondant à 2000 parties), la courbe correspond à la moyenne des valeurs entre les parties 1501 et 2000. Lorsqu'il y a moins de 500 valeurs, la courbe présente la moyenne des *n* premières valeurs. Ce graphique montre globalement une légère amélioration des résultats au cours du temps, avec un passage de 60% de réussite à 80%.

Nous avons analysé le type de mots joués par les utilisateurs. Nous avons considéré comme **courants** les mots issus de **l'échelle orthographique Dubois Buyse**[9], c'est-à-dire, ceux connus par un enfant de 12 ans. Nous considérions les autres comme normaux. Par exemple, *fourchette*, *pie*, *écureuil*, *restaurant* sont courants, tandis que *séquoia géant*, *Rabat* ou même *Akinator* sont considérés comme des termes normaux. On peut considérer que globalement, les termes normaux ont une fréquence d'utilisation allant de moyen à rare (les mots courants ayant une fréquence d'utilisation élevée). L'analyse des parties jouées, ainsi que le nombre de mots différents nous révèle que dans les deux cas, environ 25% concerne des mots courants : sur 1701 mots différents joués, 435 sont courants (soit 25,6%) et sur 6488 parties, 1565 concernent des mots courants (24,1%). Il est important de noter que les mots sont ceux qui étaient déjà bien complets dans le réseau lexical de JDM. On remarquera que l'addition pondérée par le nombre de parties des deux courbes (figures 3 et 4) donne la courbe de la figure 2.

Sur les parties jouées sur des mots courants, on observe globalement une stagnation des résultats, preuve que le réseau était déjà bien complet. Ce qui n'exclut pas que le réseau ait été enrichi de nouveaux résultats, bien que ceux-ci soient très peu visibles. En revanche, en ce qui concerne les mots normaux (ceux qui ne sont pas considérés comme courants), la progression est bien plus claire. Alors qu'au départ, la moyenne à 1000 était inférieure à 60%, elle atteint 80% à la fin. À quoi est due cette progression ? Une première explication possible serait que les joueurs découvrent AKI ; et ce n'est que petit à petit qu'ils réussissent à proposer des indices pertinents. Ceci est quelque peu contredit par l'expérience : en effet, il semble que les joueurs essaient de plus en plus de proposer des indices indirects afin de «mettre le système en défaut ». Il

[9] L'échelle orthographique Dubois Buyse permet d'indiquer les mots normalement acquis par 75% des enfants d'une classe d'âge. Nous considérons donc comme vocabulaire courant les mots bien orthographiés par 75% des enfants de 12 ans. On peut trouver cette liste, entres autres, à http://o.bacquet.free.fr/db2.htm

nous parait plus plausible que cette progression serait due à la capacité d'apprentissage du système. Cette hypothèse serait bien entendu à vérifier sur un plus long terme mais le nombre de relations acquises lors de cette expérience semble la corroborer.

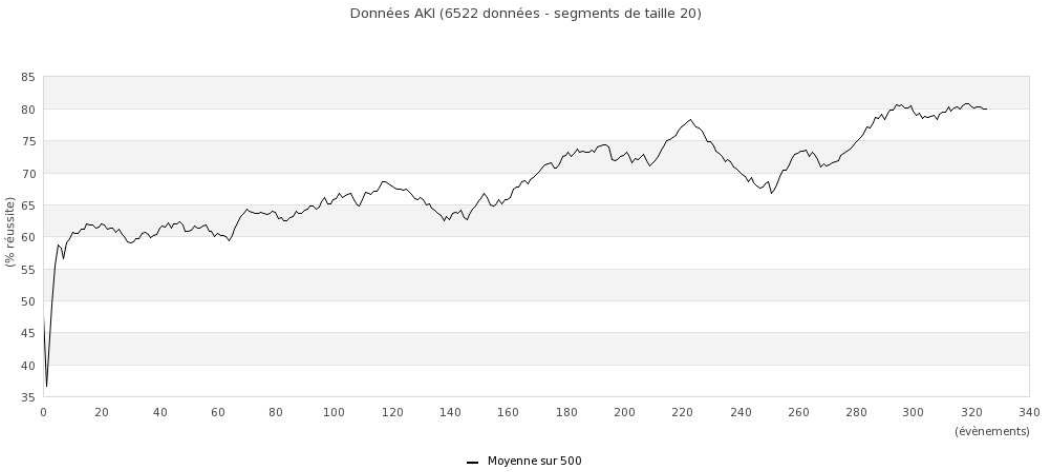


Figure 2 : Graphique montrant l'évolution dans le temps du rapport entre le nombre de parties AKI gagnées et le nombre de parties jouées (moyenne glissante sur les 500 dernières parties jouées)

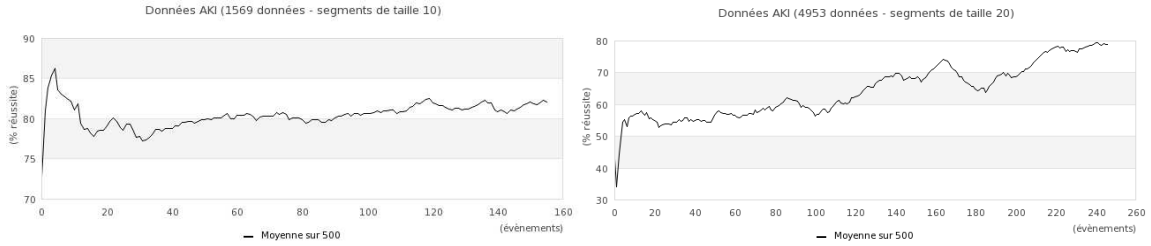


Figure 3 et 4 : A gauche, l'évolution dans le temps du rapport entre le nombre de parties gagnées et le nombre de parties jouées sur des **mots courants** (moyenne glissante sur les 500 dernières parties jouées) - A droite, l'évolution dans le temps du rapport entre le nombre de parties gagnées et le nombre de parties jouées sur des **mots de fréquence moyenne à faible** (moyenne glissante sur les 500 dernières parties jouées)

Acquisition de termes. Depuis le 1er janvier 2011, 208 nouveaux termes ont été insérés dans le réseau lexical via AKI. Ils résultent de 724 parties jouées. La quasi-totalité (90%) de ces termes sont des entités nommées (*DCRI, Révolution de jasmin, Noob, ...*), ou (10%) des termes composés et des néologismes divers (*sexe par surprise, gaz lacrymogène,…*), souvent liés à l'actualité.

Acquisition de relations. Depuis le 1er janvier 2011, 11434 relations on été acquises à l'aide de AKI (6546 parties). Si on ne compte que celles absentes du réseau, ce nombre tombe à 2105. Donc, en moyenne le réseau acquiert 1 nouvelle relation toutes les trois parties.

5.3 Analyse qualitative des parties

Sur le type de vocabulaire. Le vocabulaire (après une première analyse) se découpe en nombre de parties jouées en 24% de mots courants, le reste se divisant en 50% de mots de fréquence moyenne ou faible, et 26% de termes souvent nouveaux et liés à l'actualité. On peut considérer que ce dernier groupe est à

rapprocher des 50% si on partitionne les termes entre mots courants et les autres. Les termes liés à l'actualité conduisent souvent (69%) à un échec ce qui semble normal, étant donné qu'il s'agit de termes nouveaux (par exemple : *Révolution de jasmin, Jean-Luc Mélenchon, Médiator* - essentiellement des entités nommées) ou de termes déjà connus, mais recherchés via des indices nouveaux (*président, Tunisie, fuite => Ben Ali*). Certains mots déjà connus de AKI, c'est-à-dire présents dans le réseau lexical, sont réactualisés par l'actualité : *trafic d'organes, Lance Armstrong, aspartame*. Le compte sur l'ensemble des termes joués (indépendamment du nombre de parties) donne environ la même répartition de 1/4 de vocabulaire courant et de 3/4 de termes rares ou récents.

Sur les indices proposés. Le nombre moyen d'indices pour trouver un mot est de 2,8. Dans 40% des cas, un terme courant est trouvé dès le premier indice. Un peu moins de 3% des parties sont poursuivies au-delà de 5 indices, les 97% se divisant entre les trois cas suivants : a) le mot est trouvé avant, b) AKI échoue avant, ou c) l'utilisateur abandonne. Quand la recherche va au delà de 5 indices, la partie aboutit à une réussite dans 60% des cas. Il s'agit de termes de domaine fortement lexicalisés, et AKI est sur la bonne voie.

Une analyse des indices donnés lors des parties indique que moins de 0,3% des parties comporte au moins un indice apparemment non cohérent. Soit l'utilisateur a voulu volontairement mettre le système en défaut en donnant un indice sans rapport, soit il s'agit d'une erreur ou d'une confusion. On remarquera (avec satisfaction) que la quasi totalité des parties sont jouées "honnêtement" (ce qui peut s'expliquer par le manque d'intérêt à mettre en défaut le système avec des indices absurdes). On peut grouper les indices proposés en deux catégories :

- les indices **frontaux** (noté F) sont ceux qui amènent rapidement à la solution (trois indices au maximum). Dans le réseau, ils sont fortement connectés à la solution, en général de façon bidirectionnelle. Par exemple : *félin* pour *chat*.
- les indices **latéraux** (notés L) sont ceux qui sont très faiblement connectés à la solution et par ailleurs beaucoup plus fortement connectés à d'autres termes. Par exemple : *lisse, blanc, froid* pour *lavabo*.

Les parties concernant les mots courants correspondent à des parties dont la séquence type est : L+ (une succession d'indices latéraux). Plus le terme cible devient rare ou récent, plus la séquence type se rapproche de F+ (une succession d'indices frontaux). Il existe quelques autres schémas de parties, mais qui restent fort minoritaires. Les schémas les plus notables sont :

- L+ : uniquement des indices latéraux : *garçon, caillou, oiseau, miette* pour *Le Petit Poucet.*
- F+ : uniquement des indices frontaux : *sang* ou *couleur* pour *rouge* ou encore *mammifère, marin, défense* pour *morse*.
- L+, F : une série d'indices latéraux puis un dernier indice de type frontal permettant de trouver la solution *: blanc, lisse, dur, éléphant* pour *ivoire.*

On peut raisonnablement supposer que le schéma L+, F correspond à une activité ludique, plutôt qu'à une activité utilitaire. C'est sans doute également le cas pour le premier schéma, lorsqu'il s'agit de termes fréquents.

Sur le type d'activité. Pouvons nous déduire de l'activité enregistrée qu'il s'agit d'une activité ludique ou d'un usage MBL ? Sans doute seulement partiellement. En revanche, de nombreux indices nous portent à croire que la plupart des parties enregistrées lors de notre expérience relèvent du jeu. Nous savons que les liens partagés par nous ou des joueurs des réseaux sociaux Facebook et Twitter[10], ont généré 60% de l'activité de AKI. Ces liens proposaient de jouer tel ou tel mot. L'envoi du lien vers AKI à des listes de diffusion professionnelles (laboratoires, enseignements, sociétés savantes) générait une forte augmentation du trafic (pratiquement les autres 40%). À moins de ne penser que toutes ces personnes cherchaient le même mot à ce moment précis, on peut supposer que l'immense majorité de l'activité de AKI relève du jeu. Toujours pour aller dans ce sens, nous n'avons eu directement que deux témoignages attestant une utilisation non ludique ; dans ces deux cas, AKI s'est révélé fort utile puisqu'il a donné satisfaction aux utilisateurs.

[10] Pour voir en temps réel le compte des parties d'AKI, http://twitter.com/#!/Tot_aki

5.4 Conclusion de l'évaluation

À l'aune des résultats ci-dessus, il ne nous est pas permis de conclure avec certitude que les hypothèses 1 et 2 présentées au début de cet article sont globalement valides, mais elles ne sont pas invalidées pour autant. De nombreux indices nous laissent penser que quasiment toutes les parties analysées proviennent du jeu et non d'une utilisation réelle. Nous avons également montré que le vocabulaire utilisé durant ces jeux était le même vocabulaire que celui faisant l'objet du MBL. Ceci permet de valider notre première hypothèse de travail, à savoir, que *l'évaluation d'un outil de MBL peut se faire grâce à des joueurs.* En revanche, notre seconde hypothèse de travail —(*l'éventail des comportements des utilisateurs jouant avec un outil de MBL inclut le comportement de ceux ayant réellement besoin de retrouver un mot*)— demanderait une analyse plus fine.

Le réseau et sa consolidation via l'activité générée avec AKI permettent, dans le cas de vocabulaire complètement ouvert, de trouver le terme dans 78% des cas. Dans le cas de vocabulaire considéré comme courant, on se situe aux alentours de 82%. Enfin, dans le cas de vocabulaire filtré (issu du jeu Tabou inversé), on atteint 98,8%. On notera que, dans ce dernier cas, la performance des êtres humains se situe aux alentours de 80%.

Nous avons évalué les performances de cinq personnes sur du vocabulaire tout venant. A cette fin, nous avons choisi au hasard pour chacun d'eux 100 termes parmi ceux joués dans AKI et pour lesquels ce dernier avait soit majoritairement trouvé (50 termes) soit échoué (50 termes). Les indices donnés étaient les 5 termes les plus fortement associés dans le réseau. La performance globale a été de 46%, chiffre à comparer avec les 75-80% d'AKI.

Conclusion

Nous avons construit un réseau lexical évolutif de grande taille grâce à l'activité d'utilisateurs jouant en ligne (projet JeuxDeMots). Ces joueurs n'étant *a priori* pas des spécialistes, ce réseau représente un ensemble de connaissances générales communes. Avec des joueurs experts, il serait envisageable d'étendre ces connaissances, et donc le réseau, à des domaines spécialisés (n'est-ce pas déjà en partie le cas ?). L'intervention d'experts lexicographes, limitée à certaines corrections ainsi qu'à la validation des raffinements de termes, est ″négligeable″ compte tenu de la taille du réseau. Les questions concernant l'évaluation de la qualité d'une telle ressource, celles concernant son utilité, et la forme que peut prendre cette évaluation restent cependant ouvertes.

Le but poursuivi ici était d'évaluer la ressource lexicale ainsi produite à l'aide d'un logiciel (dénommé AKI). Celui-ci peut être considéré soit comme un jeu, soit comme un outil de MBL avec une approche exclusivement sémantique et lexicale. A l'heure actuelle, il n'y a aucune prise en compte de facteurs morphologiques ou phonologiques. AKI permet donc une évaluation à grande échelle du réseau par les utilisateurs eux-mêmes, qu'ils soient ou non des joueurs ayant contribué via JeuxDeMots. Quel que soit l'ensemble des termes considérés (termes courants ou termes de fréquence plus réduite) les performances d'AKI sont d'environ 80% ± 5%. Les résultats montrent par ailleurs que AKI est réellement utile, permettant de trouver des termes dans une ressource existante, tout en étant susceptible de l'enrichir grâce à sa capacité d'apprentissage.

On peut déduire des performance de AKI que 75% des termes pour lesquels il a été sollicité sont bien indexés, en tout cas suffisamment bien pour permettre le bon choix en cas de désambiguïsation lexicale (avocat: profession vs. fruit). L'évaluation se poursuit au long cours et les participants cherchant constamment à mettre en défaut AKI renforcent l'indexation, mais également l'évaluation avec une sévérité croissante – les deux se compensant. Il y a un auto-ajustement des joueurs en faveur d'indices faisant partie de la longue traîne. Une question restant cependant ouverte est de savoir à quel taux de réussite AKI va asymptotiquement plafonner. Cette valeur pourrait être un indice concernant une performance maximale en désambiguïsation lexicale en utilisant notre ressource.

Références

ABRAMS L., TRUNK D. L., & MARGOLIN S. J. (2007). Resolving tip-of-the-tongue states in young and older adults: The role of phonology. IN L. O. RANDAL (ED.), A*ging and the Elderly: Psychology, Sociology, and Health* (pp. 1-41). HAUPPAUGE, NY: NOVA SCIENCE PUBLISHERS, INC.

VON AHN L., DABBISH L. (2004). Labelling Images with a Computer Game. *ACM Conference on Human Factors in Computing Systems (CHI)*. pp. 319-326

AITCHISON J. (2003). W*ords in the Mind: an Introduction to the Mental Lexicon*. OXFORD, BLACKWELL.

BROWN R. & MCNEILL D. (1966). The "tip-of-the-tongue" phenomenon. *Journal of Verbal Learning and Verbal Behavior, 5*, pp. 325-337.

CARAMAZZA A. (1997). « How many levels of processing are there in lexical access ? » *Cognitive Neuropsychology*, 14, pp. 177-208.

COLLINS A, QUILLIAN M.R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behaviour*, 8(2), pp. 240-248.

FERRAND L. (1998). Encodage phonologique et production de la parole. *L'année psychologique*. vol. 98, n°3. pp. 475-509.

JI H., PLOUX S. AND WEHRLI E. (2003) Lexical knowledge representation with contexonyms. In *Proceedings of the 9th MT summit*, pp. 194-201

LAFOURCADE M., JOUBERT A. (2009). Similitude entre les sens d'usage d'un terme dans un réseau lexical. *Traitement Automatique des Langues*, vol.50/1, pp. 177-200

LAFOURCADE M., JOUBERT A. (2010). Détermination et pondération des raffinements d'un terme à partir de son arbre des usages nommés. *Traitement Automatique des Langues Naturelles (TALN'10)*. Montréal, 6 p.

LEVELT W., ROELOFS A. & A. MEYER. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences*, 22, 1-75.

LIEBERMAN H., SMITH D.A., TEETERS A. (2007). Common Consensus: a web-based game for collecting commonsense goals. *International Conference on Intelligent User Interfaces (IUI'07)*. Hawaï, USA.

MEL'ČUK I.A., CLAS A., POLGUÈRE A. (1995). *Introduction à la lexicologie explicative et combinatoire*. Editions Duculot AUPELF-UREF

MILLER G.A., BECKWITH R., FELLBAUM C., GROSS D. AND MILLER K.J. (1990). Introduction to WordNet: an on-line lexical database , *International Journal of Lexicography*, 3 (4), pp. 235-244.

PLOUX S., VICTORRI B. (1998). Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes. *Traitement Automatique des Langues*, vol.39/1, 161-182

POLGUÈRE A. (2006). Structural properties of Lexical Systems : Monolingual and Multilingual Perspectives. *Proceedings of the Workshop on Multilingual Language Resources and Interoperability (Coling/ACL)*, Sydney, pp. 50-59.

ROSSI M. (2001) : Les lapsus et la production de la parole. *Psychologie Française*, n° 46, pp. 27-41.

SITBON L. (2007). *Combinaisons de ressources linguistiques pour l'aide à l'accès lexical : études de faisabilité*, actes de RECITAL 2007, 5-8 juin 2007, Toulouse, France

SOWA J. (1992). *Semantic networks*, Encyclopedia of Artificial Intelligence, edited by S.C. Shapiro, Wiley, New York

SPENCE D.P. & OWENS K.C. (1990). Lexical co-occurrence and association strength, *Journal of Psycholinguistic Research*, 19 (5)

ZAMPA V., LAFOURCADE M. (2009). Evaluations comparées de deux méthodes d'acquisition lexicale et ontologique : JeuxDeMots vs Latent Semantic Analysis. *XVIèmes rencontre de Rochebrune : ontologie et dynamique des systèmes complexes, perspectives interdisciplinaires*

ZOCK M., FERRET O., SCHWAB D. (2010) Deliberate word access : an intuition, a roadmap and some preliminary empirical results, In *A. Neustein (éd.) 'International Journal of Speech Technology'*, 13(4):107-117, 2010. Springer Verlag.

ZOCK M. (2002). Sorry, what was your name again, or how to overcome the tip-of-the tongue problem with the help of a computer? *SemaNet workshop* (Building and Using Semantic Networks), Coling, Taipei, pp. 107-112.

Multiscale Visual Analysis of Lexical Networks

Guillaume Artignan, Mountaz Hascoët, Mathieu Lafourcade
Univ. Montpellier II
LIRMM, UMR 5506 du CNRS,
161, rue Ada 34392 MONTPELLIER Cedex, France
{artignan,mountaz,lafourcade}@lirmm.fr

Abstract

A lexical network is a very useful resource for natural language processing systems. However, building high quality lexical networks is a complex task. "Jeux de mots" is a web game which aims at building a lexical network for the French language. At the time of this paper's writing, "jeux de mots" contains 164 480 lexical terms and 397 362 associations. Both lexical terms and associations are weighted with a metric that determines the importance of a given term or association. Associations between lexical terms are typed. The network grows as new games are played. The analysis of such a lexical network is challenging.

The aim of our work is to propose a multi-scale interactive visualization of the network to facilitate its analysis. Our work builds on previous work in multi-scale visualization of graphs. Our main contribution in this domain includes (1) the automatic computation of compound graphs, (2) the proximity measure used to compute compound nodes, and (3) the computation of the containment relation used to exhibit the dense relation between one important node and a set of related nodes.

Keywords—**Multiscale, Hierarchical Graph, Visualization, Hierarchical Clustering**

1 Introduction

Many natural language processing tasks like information retrieval or anaphora resolution require lexical information usually found in resources such as thesauri, ontologies, or lexical networks. Creating such resources can be done either manually in the case of Wordnet [10] for example or automatically from text corpora as in [21]. In both cases, the generation of accurate and comprehensive data over time is a complex task.

"Jeux de Mots" [17, 16] is a game where players contribute to the creation of a complex lexical network by playing. The game is a two player blind game based on agreement: at the beginning of a game session player A is given an instruction related to a target term. For example: *give any term that is related to "cat"*. User A has a limited amount of time to propose as many terms as possible. At the end of the session, player A's proposed terms are compared to those of a previous player say player B. Points are earned on the basis of agreement, e.g. the intersection of the two sets of terms proposed by A and B. The lexical network of "Jeux de Mots" is built by adding the terms in the agreement. A relation to the target term is also added. The relation between the target term and the terms agreed depends on the initial instruction. In the previous example the relation is a relation of type association. There are 35 other types of relations in "Jeux de mots" including synonymy, antonymy, hyperonymy, etc. Weights are further computed for terms and for relations between terms in order to reflect their importance in the network [15]. At the time of this paper's writing, "jeux de mots" contains 164 480 lexical terms and 397 362 associations. Therefore, the visualization of the network is challenging. JeuxDeMots lexical network can be considered as a large graph with terms as nodes and semantic relations between terms as edges.

Multiscale interactive visualization of graphs is an interesting solution to the visual analysis of large graphs. Hierarchical graphs, introduced in [9] for the first time, have largely influenced the literature in this domain. Approaches vary at different levels. Our approach is based on compound graph construction and full zoom exploration. The construction of the compound graph is further based on a proximity measure used to compute compound nodes and the computation of the containment relation used to exhibit dense relation between one important node and a set of related nodes.

This paper is organized as follows: we start by a review of related work, we further present the data and a careful analysis of some properties that matter for visualization. We further present our main contributions e.g. compound graph construction (section 4) and full-zoom exploration of JeuxDeMots lexical network (section 5).

2 Related Work

Our approach to the visual analysis of JeuxDeMots lexical network is based on previous work and mainly related to multilevel graph exploration.

Multilevel graphs are largely used in graph visualization. Indeed multilevel graph drawing methods have been studied in order to accelerate run time and also to improve the visual quality of graph drawing algorithms. In [24], Chris Walshaw presents a multilevel optimization of the Fruchterman's and Reingold's spring embedder algorithm. The GRIP algorithm [11] coarsens a graph by applying a filtration to the nodes. This filtration is based on shortest path distance. Fast Multipole Multilevel Method (FM^3) [13] is also a force-directed layout algorithm. FM^3 is proved subquadratic (more precisely in $O(NlogN+E)$) in time, contrary to previous algorithms. Work in [3] is based on the detection of topological structures in graphs. This algorithm encodes each topological structure by a metanode to construct a hierarchical graph.

Graph hierarchies are also used in Focus-based multilevel clustering. In [6, 7, 8] several hierarchical clustering techniques are proposed, to visualize large graphs. These contributions are mainly concerned with accounting for a user focus in the construction of a multi-level structure. Sometimes this results in new multi-level structures such as for example MuSi-Tree (Multilevel Silhouette Tree) in [8]. Other approaches are based on zooming strategies that include level-of-details dependant of one or more foci [12].

Multilevel graph exploration is challenging. Multilevel graph exploration systems can be divided into two categories : systems needing precomputation to create a hierarchical structure and systems which create hierarchy during the exploration. Our approach fall into the first category.

Approaches that fall in the first category take more time during the construction step but they facilitate multi-level exploration. In [9] the authors propose an algorithm for creating a graph hierarchy in three dimensions, each level is drawn on a plane. In [20] authors propose a comparison between two methods of multi-level exploration:"Fisheyes View" and "Full-zoom" methods. Work in [12] is based on a zooming technique associated with a precomputed hierarchical graph. The level of detail is computed on-the-fly and depends on the distance to one or more foci. Abello et al. [1] define a compound fisheye view based on a hierarchy graph. In addition the authors link a treemap with a graph hierarchy. In [23], the authors create a force directed layout, and use it on graphs in order to highlight clusters. This technique is similar to the approaches that merge clusters in small world vizualization. In [5] the contribution is to propose the visualisation of complex software in 3D or in 2D. Edge bundles are created in order to simplify edges. This method uses visual simplification of graphs using a level-of-detail approach.

Approaches that fall into the second category compute a hierarchical graph during the exploration step. The layout of the graph is computed on the fly. The authors of [22] present a tool, ASK-GraphView, based on clustering and interactive navigation. Hierarchical clustering is obtained by detecting biconnex components, and by a recursive call to a clustering algorithm on biconnex components. In [4] the authors propose Grouse. Grouse is based on previous work [3] and decomposes the graph based on topological features. Grouse further uses adapted layout algorithms to layout subgraphs.

3 Data

In the rest of this paper we focus on a subgraph of JeuxDeMots. The subgraph is obtained by studying only the edges that correspond to the relation of type "Associated Idea". Furthermore, we filtered nodes that were not connected to the biggest connected component. The resulting subgraph is composed of 20 238 words and 64 564 edges.

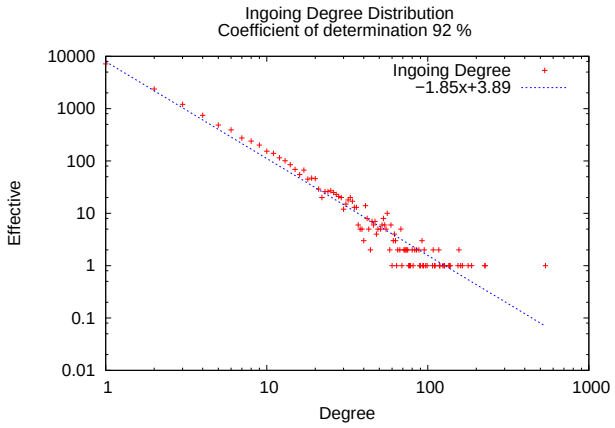


Figure 1: Degree repartition, for ingoing degree.

3.1 Data analysis

The aim of this section is to better characterise the type of graph we are working on. A study of degree repartitions (distribution of ingoing edges Fig. 1 and distribution of outgoing edges Fig. 2) is useful to show that our degree distributions have power-law tails. $\gamma_{in} = -1.85$ the indegree exponent and $\gamma_{out} = -2.27$ the outdegree exponent, are high determination coefficients [2].

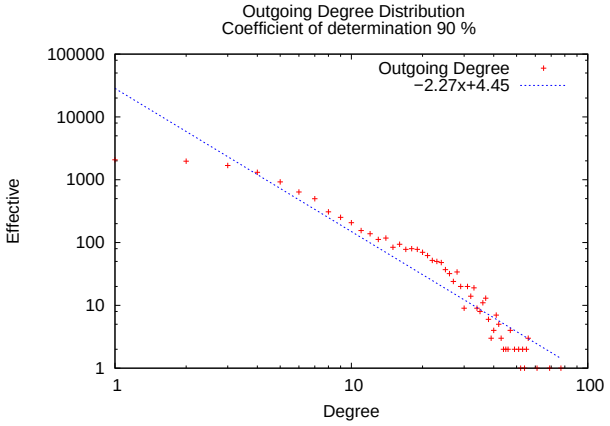


Figure 2: Degree repartition, for outgoing degree.

The graph studied is clearly a scale-free graph [2]. A second study can be made to compute the clustering coefficient [2]. The average of our graph $\overline{\mathcal{C}} = 0.2617$, and the degree average is $\overline{D} = 6.3805$. Moreover the clustering coefficient of a random graph of the same size and average degree is $C_{rand} = 0,00032$. Our graph has an average clustering coefficient order of magnitude higher than the coefficient of clustering of a random graph with the same size and the same average degree. Furthermore, the diameter of our graph is 12. For all these reasons, our graph can be considered as a small world network.

4 Compound graph construction

In order to provide full-zoom exploration of the lexical network it is necessary to automatically compute a hierarchical graph that is coherent for an end-user of lexical networks like, for example, a searcher in natural language engineering or a lexicographer.

The originality of our approach is (1) that it is based on metrics derived from natural language engineering metrics that compute at low cost, and (2) we create a compound graph instead of a clustered graph. It is important to stress that the clustered graph approach is the most frequent one found in the litterature and that it constitutes a serious drawback when it comes to lexical network exploration as will be discussed in the section 4.2.

In order to create a compound graph, we first adapted one proximity measure used in information retrieval and natural language processing tools and we further extend it to provide a multilevel proximity measure used in the construction of the compound graph.

4.1 Proximity Measure

The “Direct Proximity Measure” is computed for an edge in a graph. This measure is useful in computing another measure the “Hierarchical Proximity Measure” which will be described in the next section. The hierarchical measure applies to two nodes n and m of a hierarchy, and accounts for the direct proximity measure of the edges linking n to m.

4.1.1 Direct Proximity Measure

The proximity measure is adapted from the measure of tf-idf (term frequency - inverse document frequency) [19]. It is computed on each edge and accounts for the weight of the edge. The weight of each edge represents a degree of confidence. This measure is defined as follows:

Let G be a graph such that $G = (V, E)$, we take an edge $e \in E$ and a node $n \in N$ and we define :

- $source(e)$ the node source of edge e, and $target(e)$ the node target of edge e.
- $\omega(e)$ the weight of edge e.
- $\delta^+(n)$ is the weighted outgoing degree of n, and $\delta^-(n)$ is the weighted ingoing degree of n.

The proximity value [18] is computed using the following formula :

$$prox(e) = \frac{\omega(e)}{\delta^-(target(e))} \times \frac{\omega(e)}{\delta^+(source(e))}$$

The first (resp. second) factor of our formula corresponds to the proportion of the weight of e in ingoing edges (resp. outgoing edges) of e. The proximity measure, can be computed for a weighted graph, oriented or not. It reflects the importance of e for its extremities. In the example Fig. 3 the importance of e is weak in comparison to the total weight of all incident edges. Consequently, the two nodes have a weak proximity as shown here $prox(e) = \frac{10}{460} \times \frac{10}{285} = 0,00072$

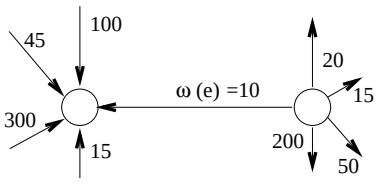


Figure 3: A sample for the proximity metric

4.1.2 Hierarchical Proximity Measure

A hierarchical proximity measurement is computed between two nodes x and y on a hierarchy of l levels with $l >= 0$ (see figure 5). This measure accounts for the edges e_i between x and nodes that are in the shortest path between x and y.

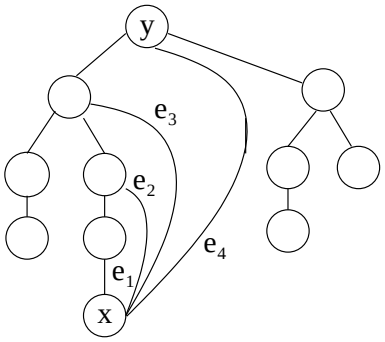


Figure 4: Hierarchical proximity measure

The measure $prox_\rho(x)$ is computed with the following formula :

$$prox_\rho(x,y) = \sum_{i=1}^{l} prox(e_i) * (l+1-i)$$

When an edge doesn't exist we will consider its weight to be equal to zero. The hierarchical measure takes parents in the multilevel graph into account and favours close parents over more distant parents.

The Fig. 5 gives an example of the computation of the hierarchical proximity measure. In this example, the computation of the hierarchical proximity measure is the following:

$$prox_\rho(x,y) = \\ prox(e_1) * 4 + prox(e_2) * 3 + prox(e_3) * 2 + prox(e_4)$$

4.2 Compound graph versus clustered graph

As mentionned previously, we chose to construct a compound graph instead of a clustered graph. The main difference between compound graphs and clustered graphs is that in the latter case meta-nodes that represent clusters are created [14]. A difficulty is then to find labels to attach to the meta-nodes created. By building a compound graph we avoid this problem since no new node has to be created. The final structure contains only the nodes of the original graph. Important nodes are used as clusters or compound nodes and the containment hierarchy can be used to express important relations between compound nodes and related nodes. This strategy offers several advantages. Firstly it underlines important nodes. Secondly, it encodes edges with high proximity measure by the containment relation of our compound graph. This graphical coding is not only stronger than simple links, it also simplifies the graphical representation by eliminating a lot of links. Thirdly, as mentionned above, there is no additionnal work to find representatives for meta-nodes, since meta-nodes are nodes, their name is directly found and meaningful.

4.3 Algorithm

In this section we present and explain our algorithm. Our algorithm Alg. 1 can be decomposed into three parts : (1) The initialisation, from line 1 to line 3, (2) the grouping of neighbours, from line 5 to line 10, and (3) the reassignement of neighbours, from line 11 to line 20.

Algorithm: Graph2GraphHierarchy(Graph G;X,Y,Z integers)

```
1  max ← getMaxDegreeNode(G,X);
2  color all nodes in max in BLACK;
3  leaves ← max ;
4  while leaves ≠ ∅ do
5      leaves2 ← get the neighbours not BLACK of
       leaves;
6      for each node n ∈ leaves2 do
7          near ← neighbours of n in leaves;
8          p ← give a node n' in near maximizing
           prox_ρ(n,root(n'));
9          child(p) ← child(p) ∪ node;
10         color node in BLACK ;
11     for each leaf ∈ leaves do
12         children ← Child(leaf);
13         selected ←
           MaxProx&DegNode(Y,Z,children);
14         for each n ∈ children \ selected do
15             near ← neighbours of n in selected;
16             node ← give a node n' in near
               maximizing prox_ρ(n,root(n'));
17             if node ≠ parent(n) then
18                 remove n from child(parent(n));
19                 child(node) ← child(node) ∪ n;
20         leave3 ← leave3 ∪ child(leaf);
21     leave ← leave3;
```

The initialisation consists in choosing X vertices with a maximum weighted degree (line 1). These vertices constitute seeds for our sub-hierarchies at the level 1. The Fig. 5 (A) describes this initialisation, here we take the nodes $\{a, b, c\}$.

The parts 2 and 3 are enclosed in a Breadth-First search algorithm. The nodes are colored in black in order to know which nodes have already been processed.

The second part of our algorithm consists in taking the neighbourhood of our seeds (line 5). Each seed will be the compound nodes of all nodes in the neighbourhood. Each seed has a different neighbourhood, nevertheless a node can be in several different neighbourhoods, see (B) in Fig. 5, the node d can be affected in two different groups.

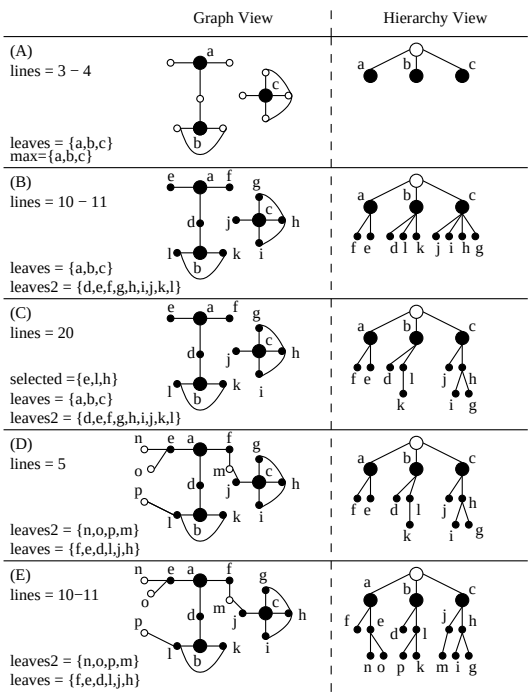


Figure 5: Execution of Graph2GraphHierarchy Algorithm

The third part of our algorithm is the reassignment. Each seed now has a list of child nodes noted $children$. We select the Y nodes in $children$ which have the highest weighted degree, and on these nodes we select the Z nodes which maximize the proximity with their parents. We obtain a list of $selected$ nodes considered as important (line 13) in the algorithm. We must further reassign all previously added nodes, to nodes in the $selected$ set of nodes if they maximize the proximity value. For example, see the Fig. 5, line (C), node i, g, k are reassigned to new parent nodes in $selected$.

We iterate with nodes contained in the next level of our hierarchy see Fig. 5 areas (D) and (E). The algorithm stops when all nodes are colored black.

The algorithm is particulary adaptable to scale-free graphs. In particular, it is possible to adapt parameter Y (number of nodes of maximal degree) and Z (number of nodes of maximal proximity) in order to favour either closer or higher degree nodes in the selected set of nodes. If the value of parameter Y is chosen in order to favour the nodes with higher degree it helps to reduce the number of links displayed (replaced by the containment relation) which in turn makes the diagram clearer.

5 Full-zoom exploration

Zoom is used to support multi-level exploration of the lexical network. It is based on the compound graph generated according to the procedure described in the previous section.

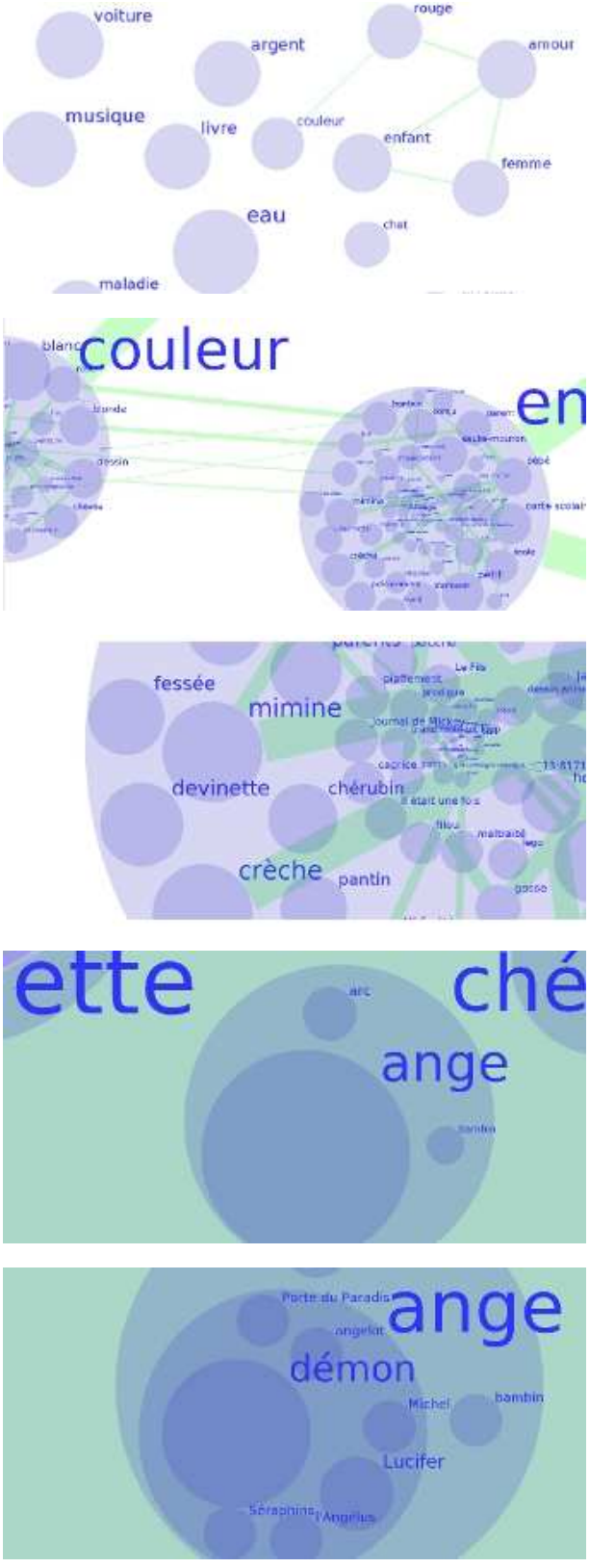


Figure 6: Full-zoom exploration

In the graphical representation, (see figure 5) nodes are represented by circles. The compound graph structure imposes that a node can contain a graph which can be empty or contains other nodes.

At each zoom level, the computation of the surface of a node is defined by :

$$surface(x) = \begin{cases} 4 * \pi^2 & \text{if } child(x) = \emptyset \\ \phi \times \sum_{c \in child(x)} surface(c) & otherwise \end{cases}$$

where ϕ is a percentage of freedom. For instance if $\phi = 120\%$, 20% of the total surface of children is left empty for the legibility of the diagram.

Furthermore, a node is expanded if its surface is higher than a percentage ζ. For instance, if $\zeta = 25\%$ the node will be expanded when its surface takes more than 25% of the screen surface. This choice allows us to adapt to various screen resolutions.

6 Conclusion and perspective

In this paper we have proposed an original method for the multi-level exploration of a lexical network. The graph underlying the lexical network has scale free and small-world properties. Even though we applied our approach to a given lexical network, we believe that our approach is general enough to apply to other networks with similar scale-free and small-world properties. For example, an interesting application would be the multiscale visualization of tags in social bookmarking systems.

In future work, we plan to extend our multi-level exploration tool with editing capacities so that it is possible for a user to modify the generated compound graph when necessary. We also want to integrate a search tool so that it is possible to automatically animate the graph toward a specific term. Finally, we plan to conduct controlled experiments to validate the approach on various datasets.

References

[1] J. Abello, S. G. Kobourov, and R. Yusufov. Visualizing large graphs with compound-fisheye views and treemaps. pages 431–441. Springer, 2004.

[2] R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47, 2002.

[3] D. Archambault and D. Auber. Topolayout: Multilevel graph layout by topological features. *IEEE TVCG*, 13(2):305–317, 2007.

[4] D. Archambault, T. Munzner, and D. Auber. Grouse: Feature-based, steerable graph hierarchy exploration. pages 67–74, 2007.

[5] M. Balzer and O. Deussen. Level-of-detail visualization of clustered graph layouts. In Seok-Hee Hong and Kwan-Liu Ma, editors, *APVIS*, pages 133–140. IEEE, 2007.

[6] F. Boutin and M. Hascoët. Focus dependent multi-level graph clustering. In *AVI'04*, pages 167–170. ACM, 2004.

[7] F. Boutin and M. Hascoët. Multi-level exploration of citation graphs. In Rachel Heery and Liz Lyon, editors, *ECDL*, volume 3232, pages 366–377. Springer, 2004.

[8] F. Boutin, J. Thièvre, and M. Hascoët. Multilevel compound tree - construction visualization and interaction. In *INTERACT*, volume 3585, pages 847–860. Springer, 2005.

[9] P. Eades and Q. Feng. Multilevel visualization of clustered graphs. pages 101–112. Springer-Verlag, 1996.

[10] C. Fellbaum. *Wordnet an electronic lexical database*. MIT Press, 1998.

[11] P. Gajer and S. G. Kobourov. Grip: Graph drawing with intelligent placement. *JGAA*, 6:2002, 2000.

[12] Emden R. Gansner. Topological fisheye views for visualizing large graphs. *IEEE TVCG*, 11(4):457–468, 2005.

[13] Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multilevel algorithm (extended abstract), 2004.

[14] Michael Junger and Mutzel Petra. *Graph Drawing Software*. Springer, 2004.

[15] M. Lafourcade. Conceptual vectors, lexical networks, morphosyntactic trees and ants : a bestiary for semantic analysis. In *SNLP 2007*, 2007.

[16] M. Lafourcade. Making people play for lexical acquisition. In *SNLP 2007*, 2007.

[17] Mathieu Lafourcade. Jeuxdemots website, November 2007-2008. http://jeuxdemots.org.

[18] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[19] G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGrawHill, 1983.

[20] D. Schaffer, S. Greenberg, L. Bartram, J. Dill, and M. Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *TOCHI*, 3:162–188, 1996.

[21] K. Sparck Jones. *Synonymy and Semantic Classification*. Edinburgh University Press, 1986.

[22] F. van Ham and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE TVCG*, 12(5):669–676, 2006. Member-Abello,, J.

[23] F. van Ham and Jarke J. van Wijk. Interactive visualization of small world graphs. In *INFOVIS'04*, pages 199–206. IEEE Computer Society, 2004.

[24] C. Walshaw. A multilevel algorithm for force-directed graph drawing. In *GD'00*, pages 171–182. Springer-Verlag, 2001.

Conclusion

L'analyse sémantique peut être abordée de façon généraliste, c'est-à-dire comme tâche ne se positionnant pas *a priori* comme préliminaire à une application particulière (comme par exemple, la traduction automatique, l'indexation de documents, le résumé, etc.). Il est alors nécessaire d'en définir les contours, à savoir les *objectifs*, les *ressources*, les *moyens* et les *usages*.

Comment concevoir l'analyse sémantique de textes ?

Les phénomènes linguistiques qui nous semblent, de façon classique, pertinents à tenter de résoudre sont l'ambiguïté lexicale, les rattachements prépositionnels, les problèmes de référence (anaphores, etc.), la reconnaissance en contexte des termes composés, l'identification des idées saillantes. L'analyse sémantique telle que nous l'avons définie ici sera la mise en relation des objets du texte, qu'ils soient les termes eux-mêmes ou des segments (syntagmes, etc.). D'autres objets implicitement présents doivent être explicités, et reliés. D'une chaîne de termes, nous obtenons donc un réseau d'objets situés (des segments du texte) et d'objets *flottants* (des idées associées au texte). Les relations sont typées et pondérées, pour en distinguer la nature et l'intensité, en premier lieu de façon à guider les processus , mais également en vue d'un filtrage du résultat. L'analyse sémantique se distingue de l'analyse thématique dans ses objectifs, mais pas nécessairement dans les moyens mis en œuvre pour y parvenir. Il s'agit alors pour l'une et l'autre de construire un objet thématique qui pourra prendre la forme d'un ou de plusieurs vecteurs d'idées ou de signatures lexicales.

Les ressources utilisées pour l'analyse peuvent être vectorielles (vecteurs d'idées) ou associatives (réseaux lexicaux). Le point critique est leur acquisition. Constituer une base de vecteurs d'idées ou créer un réseau lexical est une tâche particulièrement délicate. La construction automatique à partir de ressources se confronte aux limites de l'analyse, ou au bruit inhérent à des approches statistiques. La réduction du bruit par un filtrage sévère a comme inconvénient majeur de supprimer les activations faibles, celles constituant la *longue traîne*. L'information thématique (celle du champ sémantique) est nécessaire et doit être aussi fine que possible. Toutefois, des relations plus précises sont utiles voire indispensables. Nous pouvons nous inspirer des fonctions lexicales de I. Mel'čuk ([Mel'čuk, 1988, Mel'čuk *et al.*, 1995, Mel'čuk, 1996]) ou des qualias de Pustejovsky ([Pustejovsky, 1993, Pustejovsky *et al.*, 1993] avec en particulier les rôles télique et agentif). Ces relations ont récemment été mises en œuvre au sein du réseau lexical JeuxDeMots.

Quelles constructions de ressources lexico-sémantiques ?

Nous avons présenté, illustré et analysé comment les vecteurs d'idées (conceptuels, anonymes et lexicaux) constituent des structures intéressantes pour capturer des éléments de sémantique lexicale. Nous en avons fait de même avec la notion de réseau lexical, qui semble plus fine mais moins aisément manipulable. Les vecteurs d'idées constituent en définitive une approche particulièrement

intéressante comme *compilation locale* d'un réseau lexical. Les expériences que nous avons menées nous confirment que les vecteurs d'idées constituent des structures facilement manipulables permettant de modéliser l'information commune entre éléments lexicaux. Ce type d'information semble clairement liée à l'information mutuelle qui serait alors une mesure de la dépendance statistique entre les objets lexicaux, et va au delà de la corrélation (qui reste linéaire). Les déclinaisons entre vecteurs conceptuels, vecteurs anonymes et vecteurs lexicaux (les signatures lexicales) offrent des représentations de plus en plus lexicalisées qui semblent **complémentaires**.

La construction de ces types d'objets est un problème en soi. L'approche consistant à construire un réseau lexical à travers des jeux proposés à des non-spécialistes semble avoir démontré non seulement sa faisabilité mais son efficacité. La notion de **consensus populaire** permet non seulement d'acquérir des informations pertinentes pour le locuteur moyen, mais, et c'est une bonne surprise, aussi des termes et associations relevant de domaines de spécialité. La construction conjointe dans le temps de vecteurs d'idées à partir du réseau lexical est une alternative opératoire à celle fondée sur des corpus de définitions. De plus, les vecteurs d'idées peuvent ainsi être typés et relever de fonctions lexicales.

Nous avons montré que la modélisation de certaines fonctions lexicales, en particulier pour l'analyse sémantique, peut être réalisée via les vecteurs d'idées. Les signatures lexicales constituent alors l'approche la plus fidèle au modèle proposé par I. Mel'čuk, mais sont peut-être insuffisantes en ce qui concerne alors l'information mutuelle. La transition des vecteurs conceptuels vers les réseaux lexicaux (via les signatures lexicales) permet des approches de plus en plus lexicalisées. Le passage en sens inverse (du réseau lexical vers des vecteurs) est une forme de compilation des connaissances de plus en plus conceptuelles. Ces approches sont complémentaires et forment un ensemble diversifié de représentations lexicales et sémantiques.

Le *raffinement* progressif des structures est une idée forte concernant l'acquisition de ressources lexicales. Il ne s'agit pas simplement d'énumérer des termes et des associations, mais surtout d'en extraire des **usages de termes** qui constituent des acceptions supposées pertinentes pour le locuteur. L'ensemble du réseau s'affine au cours du temps par le couplage fort entre l'activité des utilisateurs et l'émergence des structures au sein de ce réseau.

Comment approcher le calcul ?

Nous avons mené quelques expériences en ce qui concerne l'analyse thématique et sémantique de textes. Certaines de ces expériences étaient faites conjointement au calcul de vecteurs d'idées (notamment pour le calcul de vecteurs conceptuels à partir de définitions de dictionnaires). Nous avons présenté des résultats pour les vecteurs d'idées qui ont été évalués à l'aide du jeu JeuxDeMots. Dans certains cas, l'accord entre les robots développés à cette occasion et les joueurs est plus élevé qu'entre les joueurs eux-mêmes.

Pour ce qui est de l'analyse sémantique de textes, nous avons proposé un modèle *bioinspiré* et un modèle *holistique*. Le calcul de l'analyse telle que nous l'avons présentée ici se base sur la notion de **propagation**. Des structures (vecteurs d'idées, signatures) sont déplacées, copiées et combinées à travers un environnement. Cet environnement est modifié soit dans sa topologie soit dans son contenu (les objets qui y sont présents). L'environnement peut être un arbre d'analyse morpho-syntaxique ou un graphe (dans l'esprit des graphes conceptuels, par exemple dans le cas du projet UNL). La propagation peut être accompagnée par de la *diffusion* (dans un espace vectoriel et par voisinage) et être modélisée de façons diverses via des agents réalisant des heuristiques pertinentes pour des tâches de granularité fine.

La notion de **bouclage** est une idée récurrente dans nos travaux. Il peut s'agir d'une boucle d'apprentissage pour le calcul de vecteurs d'idées, de boucles de rétroaction avec la construction de ponts dans le modèle à fourmis, ou de boucle d'inhibition dans le modèle holistique. Le bouclage est étroitement lié à la conception de système dont l'apprentissage et l'évaluation sont permanents.

La notion d'**activation** repose sur celle d'information mutuelle que ce soit de façon explicite par les vecteurs ou implicite dans un réseau. Ceci étant, considérer pleinement l'**inhibition** comme une part importante du processus de calcul nous semble primordial. Nos expériences tendent à montrer que l'inhibition pourrait jouer un rôle aussi important si ce n'est plus important que l'activation.

Quelles applications ?

L'*ingénierie des modèles* est un client particulier pour les méthodes proposées en sémantiques lexicales. En effet, il est possible de concevoir un modèle de calcul sur un réseau d'objets qui serait commun à certaines tâches d'ingénierie des modèles et à l'analyse sémantique de textes.

Le domaine de la *représentation* graphique, exploration, et manipulation de grands graphes est un domaine en pleine expansion. Le réseau lexical construit avec JeuxDeMots constitue une *donnée réelle de grande taille* pour les méthodes exploratoires. Nous pouvons raisonnablement espérer que l'étude globale de tels réseaux permettra non seulement d'améliorer les approches mais aussi d'en trouver des propriétés intéressantes pouvant être exploitées avec profit en TAL.

La définition d'un outil d'aide à la recherche de termes (un *oracle lexical*) est une application directe de la construction du réseau lexico-sémantique. Ce type d'outil, outre les services qu'il peut rendre aux utilisateurs, peut être également vu comme un moyen d'évaluation à grande échelle du réseau. Le côté ludique d'essayer de *coincer l'oracle* en fait également un jeu intéressant suscitant un certain engouement. Enfin, la capacité à retenir la réponse fournie par le joueur en cas d'échec a un impact positif sur le réseau lexical et son développement. La question de la valeur plafond du taux de réussite d'un tel outil reste ouverte. Nous obtenons un succès d'environ 75%. Quelle est la valeur maximum qu'il est raisonnable d'espérer atteindre ? Le plafond est-il un *indépassable* ou bien simplement une limite du modèle et des données ?

... et après ?

Quelles suites pouvons-nous donner à ces travaux ?

Vers un réseau lexical multilingue. Mais quelle masse critique de joueurs pour le constituer ? Quelles approches pour assurer une bonne couverture lexicale et une qualité raisonnable ?

Vers le passage à l'échelle pour l'analyse holistique. Quels autres principes de calcul pouvons-nous imaginer ? Nous pourrons penser à, entre autres, (1) l'analogie (quadrangulation), (2) l'inclusion de processus relevant de la morphologie, et (3) la qualification explicite automatique du type de contenu (humoristique, raciste, etc.). Les règles à mettre en œuvre peuvent-elles être apprises automatiquement à partir de corpus ? Peut-on utilement les pondérer ?

Vers l'agrégation d'éléments en structures complexes dans les réseaux lexico-sémantiques. Par exemple, créer des objets de la forme : *(*[embarcation] $\xrightarrow{carac}$ [surchargée]*)* $\xrightarrow{conséquence}$ [*chavirer*] . Comment amener des joueurs ou des contributeurs à les construire simplement ?

Vers le bouclage de la sémantique des relations. Comment associer une signification dans le système aux types des relations ? Comment quantifier certaines associations (*toujours vraie*, *probable*, *possible*, mais également *sexiste*, *raciste*, *humoristique*, etc.) ? Nous penserons ici à la réification dans le réseau des types de relations et de leurs occurrences sous la forme de nœuds.

Pour finir, la figure qui suit 5.19 est une tentative d'illustration de certains des liens qu'entretiennent les différents thèmes abordés dans ce mémoire.

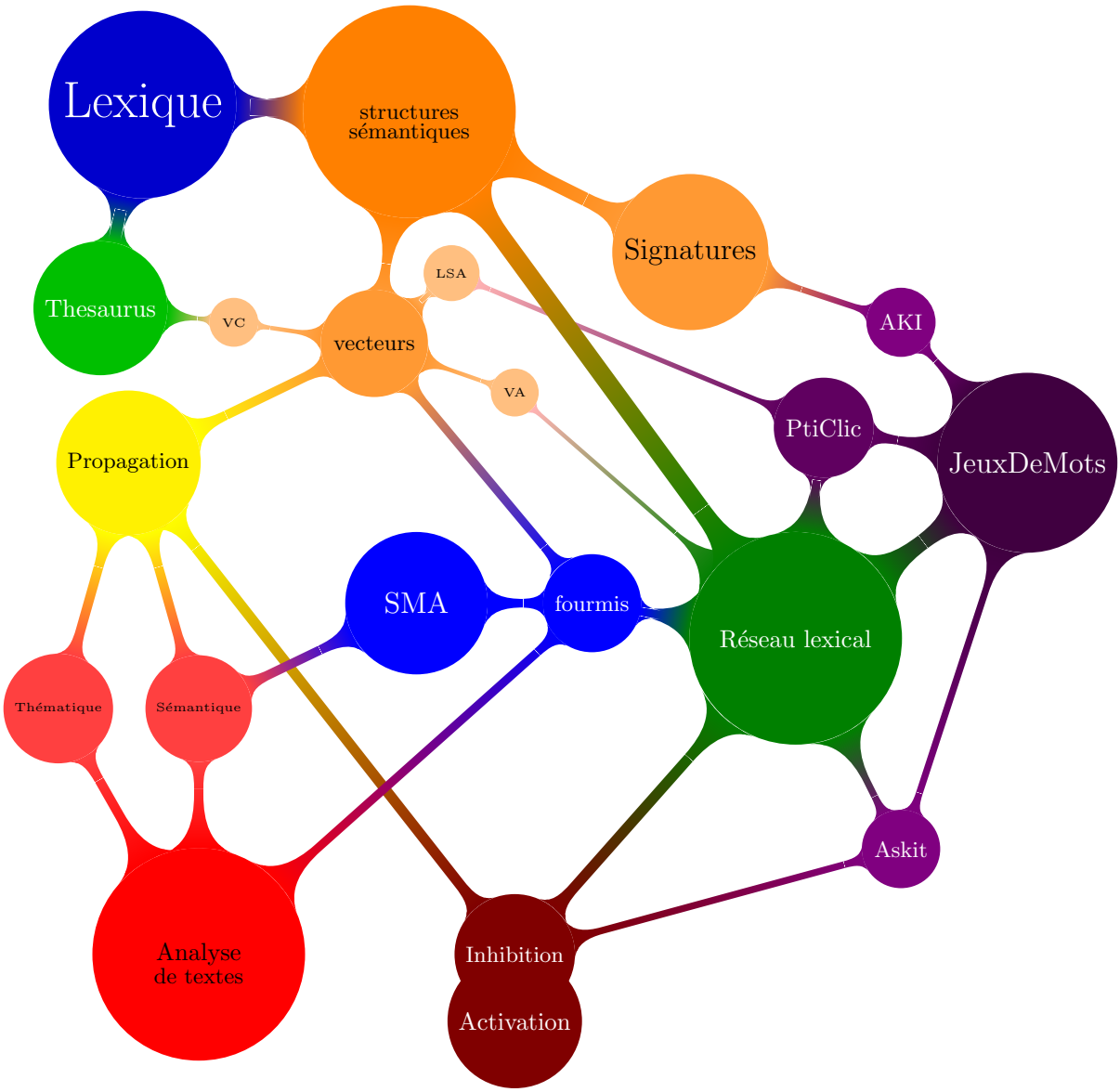


FIGURE 5.19 – Quelques liens entre quelques thèmes de ce mémoire.

Bibliographie personnelle

1. J. P. Prost et M. lafourcade (2011) *Pairing Model-Theoretic Syntax and Semantic Network for Writing Assistance* CSLP@Context'11 - 6th International Workshop on Constraints and Language Processing, Karlsruhe, Germany, 27 September 2011, 10 p.
2. M. Lafourcade et V. Prince, Eds. (2011) Actes des conférences TALN 2011 et Recital 2011. Volume 1 ISBN 978-2-84210-151-0, 554 p. et Volume 2 ISBN 978-2-84210-152-7, juin 2011, 338 p.
3. Joubert A., Lafourcade M., Schwab D., Zock M. (2011) *Évaluation et consolidation d'un réseau lexical via un outil pour retrouver le mot sur le bout de la langue* TALN'2011, 27 juin - 1er juillet 2001, Montpellier (2011), 12 p.
4. M. Lafourcade and A. Joubert (2010) *Computing trees of named word usages from a crowdsourced lexical network.* Investigationes Linguisticae, volume XXI, 2010, pp. 39-56.
5. F. Guinand and M. Lafourcade (2010) *Artificial ants for Natural Language Processing. in Artificial Ants.* From Collective Intelligence to Real-life Optimization and Beyond - N. Monmarché, F. Guinand and P. Siarry (Ed.) ISBN 978-1-84821-194-0 (2010) - Chapter 20, pp. 455-492.
6. M. Lafourcade and A. Joubert (2010) *Computing trees of named word usages from a crowdsourced lexical network.* In proc CLA 2010 - Computational Linguistics Applications - International Multi-Conference on Computer Science and Information Technology, Wisla, Pologne, 18-20 October 2010, 18 p.
7. V. Zampa et M. Lafourcade (2010) *PtiClic et PtiClic-kids : jeux avec les mots permettant une double acquisition.* In proc TICE2010, 7e colloque TICE, Nancy : 6-8 décembre 2010, 7 p.
8. A. Joubert et M. Lafourcade (2010) *Détermination et pondération des raffinements d'un terme à partir de son arbre des usages nommés.* In proc of TALN'10, Montreal, Canada, 19-23 Juillet 2010, 7 p.
9. M. Lafourcade et A. Joubert (2010) *Construction de l'arbre des usages nommés d'un terme dans un réseau lexical évolutif.* In proc of JADT'2010, Sapienza, University of Rome, Italie, 9-11 juin 2010, pp. 1249-1258.
10. J.-R. Falleri, M. Huchard, M. Lafourcade, C. Nebut, V. Prince, and M. Dao (2010) *Automatic Extraction of a WordNet-like Identifier Network from Software.* In proc ICPC 2010 : 18th IEEE International Conference on Program Comprehension, Braga, Portugal, 30th June - 2nd July 2010, ISBN : 978-0-7695-4113-6, pp. 4-13.
11. M. Lafourcade et A. Joubert (2009) *Similitude entre les sens d'usage d'un terme dans un réseau lexical.* Dans Traitement Automatique des Langues (TAL), Volume 50, Numéro 1. Varia, 2009, pp. 179-200.
12. J.-R. Falleri, V. Prince, M. Lafourcade, M. Dao, M. Huchard, and C. Nebut (2009) *Using Natural Language to Improve the Generation of Model Transformation in Software Design.* In proc Computational Linguistics Applications - International Multi-Conference on Computer Science and Information Technology, Mragowo, Pologne (2009), 8 p.
13. Artignan G., Hascoët M., and Lafourcade M. (2009) *Multiscale Visual Analysis of Lexical Networks.* In proc IV'09 : 13th International Conference Information Visualisation, Barcelona, Spain, 15-17 July 2009, ISBN : 978-0-7695-3733-7, 6 p.

14. F. Guinand et M. Lafourcade (2009) *Fourmis Artificielles et Traitement de la Langue Naturelle.* Dans Fourmis Artificielles 2. Nouvelles Directions pour une Intelligence Collective - IC2 Traité Informatique et Systèmes d'Information, Lavoisier (Ed.) (2009) - chapitre 8, pp. 225-267.
15. M. Lafourcade and V. Zampa (2009) *JeuxDeMots and PtiClic : games for vocabulary assessment and lexical acquisition.* In proc of Computer Games, Multimedia Allied technology 09 (CGAT'09). Singapore : 11th-13th may 2010, 8 p.
16. V. Zampa et M. Lafourcade (2009) *Comparaison et combinaison de deux méthodes d'acquisition lexicales pour la création d'ontologies multilingues.* in Multilinguisme et traitement des langues naturelles. PUQ. Montréal. Canada, Ismaïl Biskri et Adel Jebali Eds., ISBN 978-2-7605-2569-6 (2009) - chapitre 8, pp. 133-150.
17. M. Lafourcade, A. Joubert et S. Riou (2009) *Sens et usages d'un terme dans un réseau lexical évolutif.* Actes de TALN'09, Senlis, France, 24-26 juin 2009, 10 p.
18. V. Zampa et M. Lafourcade (2009) *Evaluations comparées de deux méthodes d'acquisitions lexicale et ontologique : Jeux De Mots vs Latent Semantic Analysis.* Actes de XVIemes rencontres de Rochebrune : ontologie et dynamique des systèmes complexes, perspectives interdisciplinaires, février 2009, 12 p.
19. M. Lafourcade and V. Zampa (2009) *PtiClic : a game for vocabulary assessment combining JeuxDeMots and LSA.* In proc of CICLing (Conference on Intelligent text processing and Comptational Linguistics). Mexico : 1-7 March 2009, 8 p.
20. M. Lafourcade et A. Joubert (2008) *Détermination des sens d'usage dans un réseau lexical construit grâce à un jeu en ligne.* In proc of TALN'08 : Traitement Automatique des Langues Naturelles, (2008), Avignon, France, 11 p.
21. M. Lafourcade and A. Joubert (2008) *Evolutionary Basic Notions for a Thematic Representation of General Knowledge.* In proc of LREC'08 : Language Resources and Evaluation Conference, (2008), Marrakech, Maroc, 28-30 May 2008, 5 p.
22. A. Joubert, M. Lafourcade (2008) *JeuxDeMots : un prototype ludique pour lémergence de relations entre termes.* In proc of JADT'2008, Ecole normale supérieure Lettres et sciences humaines, Lyon, France, 12-14 mars 2008, pp. 657-666.
23. J.-R. Falleri, M. Huchard, M. Lafourcade, C. Nebut (2008) *Meta-model Matching for Automatic Model Transformation Generation.* In proc of MODELS'08 : 11th International Conference on Model Driven Engineering Languages and Systems, 28 September - 3 October 2008, Toulouse, France, 15 p.
24. M. Lafourcade (2007) *Conceptual Vectors, Lexical Networks, Morphosyntactic Trees and Ants : a bestiary for Semantic Analysis.* Keynote speech as **invited speaker** at SNLP 2007, 7th Symposium on Natural Language Processing. Pattaya, Thailande, 13-15 December 2007.
25. M. Lafourcade, (2007) *Making people play for Lexical Acquisition.* In Proc. SNLP 2007, 7th Symposium on Natural Language Processing. Pattaya, Thailande, 13-15 December 2007, 8 p.
26. D. Schwab, Lim Lian Tze, M. Lafourcade (2007) *Conceptual vectors, a complementary tool to lexical networks.* NLPCS 2007 : The 4th International Workshop on Natural Language Processing and Cognitive Science, Funchal, Madeira - Portugal, 12-13 June, 2007, 10 p.
27. D. Schwab, M. Lafourcade (2007) *Lexical Functions for Ants Based Semantic Analysis.* ICAI'07 - The 2007 International Conference on Artificial Intelligence, Monte Carlo Resort, Las Vegas, Nevada, USA, 25-28 June, 2007, 10 p.
28. D. Schwab, Lim Lian Tze, et M. Lafourcade (2007) *Les vecteurs conceptuels, un outil complémentaire aux réseaux lexicaux.* Actes de TALN'2007 : Traitement Automatique des Langues Naturelles, Toulouse, 5-8 juin 2007, pages 293-302.
29. N. Gala et M. Lafourcade (2007) *PP Attachment Ambiguity Resolution with Corpus-Based Pattern Distributions and Lexical Signatures.* ECTI Journal, Vol.2, No2, ISSN 1905-050X, pp. 116-120.
30. D. Schwab et M. Lafourcade (2007) *Modelling, Detection and Exploitation of Lexical Functions for Analysis.* ECTI Journal, 2007, Vol.2, No2, ISSN 1905-050X, pp. 97-108.
31. M. Lafourcade (2006) *Conceptual Vector Learning - Comparing Bootstrapping from a Thesaurus or Induction by Emergence.* In proc of LREC'2006, Magazzini del Cotone Conference Center, Genoa, Italia, 24-26 Mai 2006, 4 p.

32. A. Joubert et M. Lafourcade (2006) *D'une hiérarchie figée de concepts vers une hiérarchie évolutive de notions de base.* Actes de JADT'2006, Université de Franche-Conté, Besançon, France, 19-21 avril 2006, 8 p.
33. A. Joubert, M. Lafourcade et D. Schwab (2006) *Approche évolutive des notions de base pour une représentation thématique des connaissances générales.* Actes de TALN'2006 : Traitement Automatique des Langues Naturelles, Leuven, Belgique, 10-13 Avril 2006, 10 p.
34. L. Abrouk et M. Lafourcade (2006) *Enrichissement d'ontologies dans le secteur de l'eau douce en environnement Internet distribué et multilingue.* In proc of EGC'2006, ENIC Telecom, Lille, France, 25 au 27 janvier 2006, pp. 709-710.
35. M. Bouklit et M. Lafourcade (2006) *Propagation de signatures lexicales dans le graphe du Web.* In proc of RFIA'2006, Tours, France, 25 au 27 janvier 2006, 9 p.
36. V. Prince, and M. Lafourcade (2006) *Mixing Semantic Networks and Conceptual Vectors - Application to Hyperonymy.* IEEE Transactions on Systems, Man, and Cybernetics : Part C., 11 p.
37. L. Abrouk and M. Lafourcade, (2005) *Application Of The Papillon Project To Ontology Management.* In proc. PAPILLON-SNLP-05, 6th Symposium on Natural Language Processing. Chiang Rai, Thailande, 6 p.
38. M. Lafourcade and D. Schwab (2005) *Multi-castes ants algorithms for holistic semantic text analysis.* In proc. SNLP-05, 6th Symposium on Natural Language Processing. Chiang Rai, Thailande, 6 p.
39. N. Gala and M. Lafourcade, (2005) *Combining corpus-based pattern distributions with lexical signatures for PP attachment ambiguity resolution.* In Proc. SNLP-05, 6th Symposium on Natural Language Processing. Chiang Rai, Thailande, 6 p.
40. M. Lafourcade (2005) *Analyse sémantique de textes et algorithmes à fourmis.* Présentation orale au DELIC, Aix-en-Provence, 3 novembre 2005.
41. M. Lafourcade (2005) *Semantic Analysis through Ant Algorithms, Conceptual Vectors and Fuzzy UNL Graphs.* In Universal Networking Language Advances in Theory and Applications, Jesús Cerdeñosa, Alexander Gelbukh, Edmundo Tovar Eds., ISBN : 970-36-0226-6, pp. 125-137.
42. D.Schwab, M. Lafourcade, et V. Prince (2005) *Extraction semi-supervisée de couples d'antonymes grâce à leur morphologie.* In proc of TALN'2005 : Traitement Automatique des Langues Naturelles , Dourdan, France, Juin 2005, 10 p.
43. M. Lafourcade et D. Schwab (2005) *Estimation automatique de la distribution des sens de termes.* In proc of INFORSID'2005 : informatique des organisations et systèmes d'information et de décision , Grenoble, France, Mai 2005, 14 p.
44. F. Guinand et M. Lafourcade, (2005) *Algorithme de fourmis pour le traitement de la langue naturelle.* Actes du 6e congrès de la société française de recherche opérationnelle et d'aide à la décision (ROADEF'05). École polytechnique de l'Université de Tours 14-16 février 2005, pp. 239-240.
45. M. Lafourcade, F. Rodrigo, et D. Schwab (2004) *Low Cost Automated Conceptual Vector Generation from Mono and Bilingual Resources.* In proc of PAPILLON-2004, Grenoble, France, 30 Août - 1 Septembre 2004, 10 p.
46. F. Jalabert et M. Lafourcade (2004) *Classification automatique de définitions en sens.* Actes de JEP-TALN 2004, Fez, Maroc, 19-22 avril 2004, 6 p.
47. D. Schwab, M. Lafourcade et V. Prince (2004) *Hypothèse pour la construction et l'exploitation conjointe d'une base lexicale sémantique basée sur les vecteurs conceptuels.* In proc. of JADT 2004 : 7es Journées internationales d'Analyse statistique des Données Textuelles, Louvain-le-Neuve, Belgique, mars 2004, 12 p.
48. M. Lafourcade et V. Prince (2004) *Modélisation de lHyperonymie via la combinaison de réseaux sémantiques et de vecteurs conceptuels.* In proc. of JADT 2004 : 7es Journées internationales d'Analyse statistique des Données Textuelles, Louvain-le-Neuve, Belgique, mars 2004, 8 p.
49. F. Jalabert et M. Lafourcade (2004) *Nommage sens à l'aide de vecteurs conceptuels.* In proc. of RFIA 2004, Toulouse, 28-30 janvier 2004, pp. 539-547.

50. M. Lafourcade (2003) *Ant Algorithms, Conceptual Vectors, and Fuzzy UNL Graphs.* In proc. of Convergences'03, International Conference on the Convergence of Knowledge, Language and Information Technologies (UNL), Alexandia, Egypt, December 2-6, 2003, ISBN 84-607-9579-9, 7 p.
51. M. Lafourcade (2003) *Conceptual Vectors and Fuzzy Templates for Discriminating Hyperonymy is-a and Meronymy part-of relations.* In proc. of OOIS 2003 Workshop MASPEGHI, P.Valtchev, M. Huchard, H. Astudillo (eds.), Montréal, Canada October 6th 2003, ISBN 2-89522-035-2, pp. 19-29.
52. M. Lafourcade (2003) *Sémantique lexicale et TALN - Vecteurs conceptuels et apprentissage.* Présentation orale à ONTOBIO-2003, LIRMM, Montpellier, 11-12 juin 2003, 2003.
53. M. Lafourcade, G. Sérasset, L.Metzger, A.Rahman, C.-K. Chuah (2003) *Dictionnaire Français-Anglais-Malais (FeM)* - version 2, avril 2003. CD-ROM, Dictionnaire en version XML et Application Java.
54. V. Prince et M. Lafourcade (2003) *Mixing Semantic Networks and Conceptual Vectors : the Case of Hyperonymy.* In proc. of ICCI-2003 (2nd IEEE International Conference on Cognitive Informatics), South Bank University, London, UK, August 18 - 20, 2003, pp. 121-128.
55. M. Mangeot-Lerebours, G. Sérasset, et M. Lafourcade (2003) *Construction collaborative dune base lexicale multilingue - Le projet Papillon.* In Traitement Automatiques des Langues (TAL), Vol 44, n°2/2003, pp. 151-176.
56. F. Jalabert et M. Lafourcade (2003) *From sense naming to vocabulary augmentation in Papillon.* In proc. of PAPILLON-2003, Sapporo, Japan, July 2003, 12 p.
57. D. Schwab, M. Lafourcade et V. Prince (2003) *Amélioration de liens entre acceptions par fonctions lexicales vectorielles symétriques.* In proc. of TALN'2003, Batz-sur-Mer, France, juin 2001, pp. 235-244.
58. M. Lafourcade (2002) *Guessing Hierarchies and Symbols for Word Meanings through Hyperonyms and Conceptual Vectors.* In proc. of OOIS 2002 Workshop MASPEGHI, Montpellier, France, September 2002, Springer, LNCS 2426, pp. 84-93.
59. M. Lafourcade (2002) *Lens Effects in Autonomous Terminology Learning with Conceptual Vectors.* In proc. of Seminar on linguistic meaning representation and their applications over the World Wide Web, Penang, Malaysia, 12 p.
60. M. Lafourcade, V. Prince, et D. Schwab (2002) *Vecteurs conceptuels et structuration émergente de terminologies.* In Traitement Automatiques des Langues (TAL), Vol 43, n°1/2002, pp. 43-72.
61. M. Lafourcade (2002) *Automatically Populating Acception Lexical Database through Bilingual Dictionaries and Conceptual Vectors.* In proc. of PAPILLON-2002, Tokyo, Japan, August 2002.
62. D. Schwab et M. Lafourcade (2002) *Hardening of Acception Links through Vectorized Lexical Functions.* In proc. of PAPILLON-2002, Tokyo, Japan, July 2002, 17 p.
63. D. Schwab, M. Lafourcade et V. Prince (2002) *Antonymy and Conceptual Vectors.* In proc. of COLING'2002, Taipei, Taiwan, August 2002, Vol2/2, pp. 904-910.
64. D. Schwab, M. Lafourcade et V. Prince (2002) *Vers l'apprentissage automatique pour et par les vecteurs conceptuels de fonctions lexicales l'exemple de l'antonymie.* In proc. of TALN'2002, Nancy, France, juin 2002, 10 p.
65. M. Lafourcade and Ch. Boitet (2002) *UNL lexical Selection with Conceptual Vectors.* In proc. of LREC'2002, Las Palmas, Canary Island, Spain, May 27, 2002, 7 p.
66. D. Schwab, M. Lafourcade et V. Prince (2002) *Amélioration de la représentation sémantique lexicale par les vecteurs conceptuels : le rôle de l'antonymie.* In proc. of JADT 2002 : 6es Journées internationales d'Analyse statistique des Données Textuelles, Saint-Malo, France, mars 2002, pp. 701-712.
67. M. Lafourcade et V. Prince (2001) *Synonymy and conceptual vectors.* In proc. of NLPRS'2001, Tokyo, Japan, November 2001, pp. 127-134.
68. Hasida K., Andres F., Boitet C., Calzolari N., Declerck T., Farshad Fotouhi, William Grosky, Shun Ishizaki, Asanee Kawtrakul, Mathieu Lafourcade, Katashi Nagao, Hammam Riza, Virach Sornlertlamvanich, Remi Zajac, Zampolli A. (2001) *Linguistic DS.* ISO/IEC JTC1/SC29/WG11, MPEG2001/ M7818.
69. M. Lafourcade et V. Prince (2001) *Synonymies et vecteurs conceptuels.* In proc. of Traitement Automatique du Langage Naturel (TALN'2001), Tours, France, Juillet 2001, pp. 233-242.

70. M. Lafourcade (2001) *Lexical sorting and lexical transfert by conceptual vectors.* In proc. of the First International Workshop on MultiMedia Annotation (MMA'2001) Tokyo, January 2001, 6 p.
71. M. Lafourcade (2001) *Lexiques et vecteurs conceptuels : vers une indexation, et une classification automatique.* Rapport LIRMM, 20 p.
72. M. Lafourcade et V. Prince (2001) *Fonction lexicales et vecteurs conceptuels synonymes.* Rapport LIRMM, 15 p.
73. D. Burnham, S. Luksaneeyanawin, Ch. Davis and M. Lafourcade, Eds. (2000) *Interdisciplinary Approaches to Language Processing.* Proc. of the International Conference on Human and Machine Processing of Language and Speech. Chulalongkorn University, Bangkok, Thailand, ISBN : 1-86341-859-8, 336 p.
74. M. Lafourcade et E. Sandford (1999) *Analyse et désambiguïsation lexicale par vecteurs sémantiques.* In proc. of Traitement Automatique du Langage Naturel (TALN'1999), Cargèse, France, Juillet 1999, pp. 351-356.
75. J. Chauché, F. Guinand et M. Lafourcade(1999) *Compression parallèle de donnée textuelles avec contraintes de positionnement.* In proc. ROADEF'99, 13-15 janvier 1999, 8 p.
76. M. Lafourcade et J. Chauché (1998) *Ficus - un agent dictionnaire coopératif et extensible.* NLP+IA'98, August 18-21, 1998, Moncton, New-Brunswick, Canada, 8 p.
77. M. Lafourcade, L. Fischer et B. M. Hamrouni (1998) *Alamet : Mémoires de traductions multilingues pour la localisation de logiciels.* COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Québec, Canada, 8 p.
78. M. Lafourcade (1998) *Problématique de la construction et de la distribution de dictionnaires n-lingues : exemples du projet de dictionnaire Français-anglais-malais-thaï et de l'outil ALEX.* In Stratégies informationnelles et valorisation de la recherche scientifique publique., Ed. F. Renzetti, ADBS.
79. Ch. Boitet J. C., G. Fafiotte, E. Keller, M. Lafourcade, E. Wehrli (1998) *Integrating French within C-STAR II : second report demos of the CLIPS++ group.* CLIPS++ (GETA-CLIPS-IMAG, LAIP, LATL, LIRMM), January 12-14 1998, 16 p.
80. Lafourcade M. Rivepiboon W. (1997) *Issues in the French-English-Thai Dictionary Project.* Proc. International Workshop on Human and Computer Processing of Language and Speech, December 8-12, Chulalongkorn University, Bangkok, Thailand, vol. 1/1, pp. 272-288.
81. Lafourcade M. et Hamrouni B. M. (1997) *Le projet ALAMET - vers l'utilisation des mémoires de traduction pour la localisation des logiciels.* In proc. Ves Journées Scientifiques du Réseau LTT (Lexicologie, Terminologie Traduction) - La mémoire des mots., 25-27 septembre 1997, Tunis, Tunisie, 8 p.
82. Lafourcade M. (1997) *Multilingual Computing and the Net.* In proc. International Workshop on Human and Computer Processing of Language and Speech, December 8-12, Chulalongkorn University, Bangkok, Thailand, vol. 1/1, pp. 289-306.
83. Lafourcade M. (1997) *Multilingual Dictionary Construction and Services - Case Study with the Fe* Projects.* In proc. PACLING'97, September 2-5 1997, Meisei University, Ohme, Tokyo, Japan, vol. 1/1, pp. 173-181.
84. Lafourcade M. (1997) *Construction et services dictionnaires n-lingues, exemple des projets Fe*.* In proc. Quatrième conférence annuelle sur Le traitement Automatique du Langage Natural (TALN), 12-13 juin 1997, Grenoble, France, vol. 1/1, pp. 162-168.
85. Lafourcade M. et Sérasset G. (1996) *CGI in Lisp.* MacTech magazine, 12/7, July 1996, pp. 25-32.
86. Lafourcade M. (1996) Serveurs de dictionnaires - Etude de cas avec l'outil Alex et leprojet de dictionnaire Français-Anglais-Malais. In proc. Séminaire Lexique - Représentation et Outils pour les Bases Lexicales - Morphologie Robuste, 13 et 14 novembre 1996, CLIPS-IMAG, Grenoble, France, vol. 1/1, pp. 185-192.
87. Lafourcade M. (1996) *TED 1.0* (a Tree EDitor). GETA, CLIPS, IMAG - available at `ftp://www.digitool.com` , June 1996. Freeware, version 1.0.
88. Lafourcade M. (1996) *DOP 3.5* (a Dictionary Object Protocol). GETA, CLIPS, IMAG - available at `ftp://www.digitool.com` , January 1996. Freeware, version 3.5.

89. Lafourcade M. (1996) *Software engineering in the LIDIA project : distribution of interactive disambiguation and other components between various processes and machines.* Proc. MIDDIM-96, Col de Porte - France, GETA (CLIPS, IMAG) ATR Interpreting Telecommunications (ATR), 6 p.

90. Lafourcade M. (1996) *Structured Lexical data : how to make them widely avalaible, useful and reasonable protected ? - a practical example with a trilingual dictionary.* Proc. COLING-96, Copenhagen, Denmark, vol. 2/2, pp. 1106-1110.

91. Lafourcade M. (1996) *ALEX 1.0* (a dictionary tool). GETA, CLIPS, IMAG - available at `ftp://www.digitool.com` , May 1996. Freeware, version 1.0.

92. Lafourcade M. (1996) *Geta-trees 1.0* (an MCL-based toolbox for tree manipulation). GETA, CLIPS, IMAG - available at `ftp://www.digitool.com` , February 1996. Freeware, version 1.0.

93. Gut Y., Yusoff Z., Samat S. A., Boitet C., Nedobejkine N., Lafourcade M. et al. (1996) *Kamus Perancis Melayu dewan - dictionnaire français-malais.* Dewan Bahasa dans Pustaka (DBP), Kuala Lumpur, 1 vol., 667 p. ISBN : 983-62-5363-7 (kkt 983-62-5364-5).

Bibliographie

[Abrouk & Lafourcade, 2006] L. ABROUK et M. LAFOURCADE. « Enrichissement d'ontologies dans le secteur de l'eau douce en environnement Internet distribué et multilingue. ». *In proc of EGC'2006, ENIC Telecom, Lille, France, 5 au 27 janvier 2006*, pp 709–710, 2006.

[Amar *et al.*, 2004] P. AMAR, J.-P. CORNET, F. KÉPÈS, et V. NORRIS, . *Proceedings of the Evry Spring School on "Modelling and Simulation of Biological Processes in the Context of Genomics"*, 2004.

[Artignan *et al.*, 2009] G. ARTIGNAN, M. HASCOËT, et M. LAFOURCADE. « Multiscale Visual Analysis of Lexical Networks. ». *IV'09 : 13th International Conference Information Visualisation, Barcelona, 14- 17 July 2009*, page 6, 2009.

[Attali *et al.*, 1992] A. ATTALI, G. BOURQUIN, M.-C. BOURQUIN-LAUNEY, A. EUVRARD, et C. VIGROUX. « Aide au transfert lexical dans une perspective de TAO : expérimentation sur un lexique non terminologique ». *Journal des traducteurs/Translators' Journal*, pp 770–790, 1992. http ://id.erudit.org/iderudit/001917ar.

[Bak, 1996] P. BAK. *How Nature works : the science of self-organized criticality*. Springer-Verlag, 1996.

[Barbier *et al.*, 2008] J. D. BARBIER, M. BERTRAND, O. CHABERT, et J. WANG. « Robot JeuxDeMots. ». *TER Master 1 informatique - FMIN200 - Rapport final, encadrement M. Lafourcade, Université Montpellier 2.*, page 48, 2008.

[BJ2004] E. BEN-JACOB et H. LEVINE. « Des fleurs de bactéries ». *Pour la Science - Les formes de la vie*, pp 78–83, 2004.

[Bonnin *et al.*, 2005] G. BONNIN, M. DOUIBI, L.-B. KOENIG, et A. LAKHOUA. « Propagation d'informations sur le Web. ». *TER Maîtrise informatique - Rapport final, encadrement M. Lafourcade et M. Bouklit, Université Montpellier 2.*, page 20, 2005.

[Bouklit & Lafourcade, 2006] M. BOUKLIT et M. LAFOURCADE. « Propagation de signatures lexicales dans le graphe du Web ». Dans les actes de *RFIA'2006, Tours, France, 25 au 27 janvier 2006*, Tours, France, 2006.

[Boulet & Balandjian, 2004] S. BOULET et J. BALANDJIAN. « Décomposition morphologique de mots inconnus. ». *TER Maîtrise informatique - Rapport final, encadrement M. Lafourcade et D. Schwab, Université Montpellier 2.*, page 40, 2004.

[Bur2000] *Interdisciplinary Approaches to Language Processing. Proc. of the International Conference on Human and Machine Processing of Language and Speech*. D. Burnham, S. Luksaneeyanawin, Ch. Davis and M. Lafourcade, Eds., 2000.

[Chaté & Grégoire, 2004] H. CHATÉ et G. GRÉGOIRE. « La forme des groupements animaux ». *Pour la Science - Les formes de la vie*, pp 57–61, 2004.

[Chauché, 1990] J. CHAUCHÉ. « Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance ». *TA Information*, pp 17–24, 1990.

[Colson *et al.*, 2005] E. COLSON, L.LACRESSONNIÈRE, et J. LOPEZ. « Extraction noms propres à partir de définitions encyclopédiques. ». *TER Maîtrise informatique - Rapport final, encadrement M. Lafourcade et D. Schwab, Université Montpellier 2.*, page 20, 2005.

[Daoud, 2010] M. DAOUD. « Utilisation de ressources non conventionnelles et de méthodes contributives pour combler le fossé terminologique entre les langues en développant des "préterminologies" multilingues. ». *Thèse de Doctorat en informatique, Dir. Ch. Boitet, LIG, Université Joseph-Fourier - Grenoble I*, page 192, 2010.

[Deerwester *et al.*, 1990a] S. DEERWESTER, S. DUMAIS, T. LANDAUER, G. R. FURNAS, et HARSHMAN. « Indexing by latent semantic analysis ». *Journal of the American Society of Information Science*, pp 391–407, 1990.

[Deerwester *et al.*, 1990b] Scott C. DEERWESTER, Susan T. DUMAIS, Thomas K. LANDAUER, George W. FURNAS, et Richard A. HARSHMAN. « Indexing by Latent Semantic Analysis ». *Journal of the American Society of Information Science*, pp 391–407, 1990.

[Delorme, 2003] Jean-Michel DELORME. « *Contribution à la réalisation d'un système de traduction automatique construit autour du moteur d'analyse et de génération transductionnel SYGMART* ». Diplôme d'ingénieur CNAM, Conservatoire National des Arts et Métiers, Montpellier Languedoc-Roussillon, 2003.

[Falleri *et al.*, 2009] J.-R. FALLERI, V. PRINCE, M. LAFOURCADE, M. DAO, M. HUCHARD, et C. Nebut C.. « Using Natural Language to Improve the Generation of Model Transformation in Software Design. ». *In proc Computational Linguistics Applications - International Multi-Conference on Computer Science and Information Technology, Mragowo, Pologne*, page 8, 2009.

[Falleri *et al.*, 2010] J.-R. FALLERI, M. HUCHARD, M. LAFOURCADE, C. NEBUT, V. PRINCE, et M. DAO. « Automatic Extraction of a WordNet-like Identifier Network from Software. ». *In proc ICPC 2010 : 18th IEEE International Conference on Program Comprehension, Braga*, page 8, 2010.

[Fellbaum, 1988] Christiane FELLBAUM, . *WordNet : An Electronic Lexical Database*. The MIT Press, 1988.

[Fomichov, 2010] V. A. FOMICHOV. « Semantics-Oriented Natural Language Processing : Mathematical Models and Algorithms. ». *IFSR International Series on Systems Science and Engineering, Springer : New York, Dordrecht, Heidelberg, London*, page 354, 2010.

[Gala & Lafourcade, 2007] N. GALA et M. LAFOURCADE. « PP Attachment Ambiguity Resolution with Corpus-Based Pattern Distributions and Lexical Signatures. ». *ECTI Journal, Vol.2, No2, ISSN 1905-050X*, pp 116–120, 2007.

[Gale *et al.*, 1992] W. GALE, K. W. CHURCH, et D. YAROWSKY. « A Method for Disambiguating Word Senses in a Large Corpus ». *Computers and Humanities*, pp 415–439, 1992.

[Gamallo & Bordag, 2011] Pablo GAMALLO et Stefan BORDAG. « Is singular value decomposition useful for word similarity extraction ? ». *Language Resources and Evaluation (LRE)*, pp 95–119, 2011, Springer Netherlands. 10.1007/s10579-010-9129-5.

[Grassé, 1959] P.-P. GRASSÉ. « La reconstruction du nid et les coordinations inter-individuelles chez Belicositermes natalensis et Cubitermes S.P. La théorie de la Stigmergie : essai d'interprétation du comportement des termites constructeurs ». *Insectes sociaux*, pp 41–80, 1959.

[Gui2010] *Artificial ants for Natural Language Processing. in Artificial Ants From Collective Intelligence to Real-life Optimization and Beyond*. N. Monmarché, F. Guinand and P. Siarry (Eds.), 2010.

[Gut *et al.*, 1996] Y. GUT, Z. YUSOFF, S. A. SAMAT, C. BOITET, N. NEDOBEJKINE, et M. LAFOURCADE, . *Kamus Perancis Melayu dewan - dictionnaire français-malais*. Dewan Bahasa dans Pustaka (DBP), Kuala Lumpur, 1996.

[Harabagiu *et al.*, 1999] S. M. HARABAGIU, G. A. MILLER, et D. I. MOLDOVAN. « WordNet 2 - A Morphologically and Semantically Enhanced Resource ». Dans les actes de *Workshop SIGLEX'99 : Standardizing Lexical Resources*, pp 1–8, 1999.

[Hascoët & Dragicevic, 2011] Mountaz HASCOËT et Pierre DRAGICEVIC. ≪ Visual Comparison of Document Collections Using Multi-Layered Graphs ≫. , 2011.

[Hearst, 1992] Marti HEARST. ≪ Automatic Acquisition of Hyponyms from Large Text Corpora ≫. Dans les actes de *COLING'1992 : 14th International Conference on Computational Linguistics*, pp 539–545, Nantes, France, 1992.

[Hofstadter, 1995] D. HOFSTADTER. *Fluid Concepts & Creative Analogies : Computer Models of the Fundamental Mechanisms of Thought*. HarperCollins Publishers, 1995.

[Hudson, 2007] R. HUDSON. ≪ Language Networks. The new Word Grammar. ≫. *Oxford University Press, ISBN-13 : 978-0199298389*, page 288, 2007.

[Jalabert & Lafourcade, 2002] F. JALABERT et M. LAFOURCADE. ≪ From sense naming to vocabulary augmentation in Papillon ≫. Dans les actes de *PAPILLON-2003, Sapporo, Japan, July 2002*, Sapporo, Japan, 2002.

[Jalabert & Lafourcade, 2003] F. JALABERT et M. LAFOURCADE. ≪ From sense naming to vocabulary augmentation in Papillon ≫. Dans les actes de *PAPILLON-2003*, page 12, Sapporo, Japon, Juillet 2003.

[Jalabert & Lafourcade, 2004a] F. JALABERT et M. LAFOURCADE. ≪ Classification automatique de définitions en sens ≫. Dans les actes de *Traitement Automatique du Langage Naturel (TALN'2005)*, Fèz, Maroc 2004.

[Jalabert & Lafourcade, 2004b] F. JALABERT et M. LAFOURCADE. ≪ Nommage sens à l'aide de vecteurs conceptuels. ≫. Dans les actes de *Reconnaissance des Formes et Intelligence Artificielle (RFIA'2004*, volume 2, pp 539–547, Toulouse, Janvier 2004.

[Jalabert, 2003] F. JALABERT. ≪ Catégorisation de définitions et nommage de sens ≫. Mémoire de dea, DEA informatique, Université Montpellier II, LIRMM, Juillet 2003.

[Joubert & Lafourcade, 2008a] A. JOUBERT et M. LAFOURCADE. ≪ Détermination des sens d'usage dans un réseau lexical construit grâce à un jeu en ligne ≫. *Actes de TALN'08, juin 2008, Avignon*, page 10 p., 2008.

[Joubert & Lafourcade, 2008b] A. JOUBERT et M. LAFOURCADE. ≪ JeuxDeMots : un prototype ludique pour lémergence de relations entre termes. ≫. *In proc of JADT'2008, Ecole normale supérieure Lettres et sciences humaines , Lyon, France, 12-14 mars 2008*, page 10, 2008.

[Joubert & Lafourcade, 2010] A. JOUBERT et M. LAFOURCADE. ≪ Détermination et pondération des raffinements d'un terme à partir de son arbre des usages nommés ≫. Dans les actes de *TALN'10, Montreal, Canada, 19-23 Juillet 2010*, Montreal, Canada, 2010.

[Joubert *et al.*, 2011] A. JOUBERT, M. LAFOURCADE, D. SCHWAB, et M. ZOCK. ≪ Évaluation et consolidation d'un réseau lexical via un outil pour retrouver le mot sur le bout de la langue. ≫. *TALN'2011, Montpellier, 27 juin - 1er juillet, 2011*, page 12, 2011.

[Kanerva *et al.*, 2000] P. KANERVA, J. KRISTOFERSSON, et A. HOLST. ≪ Random indexing of text samples for latent semantic analysis. ≫. *Proceedings of the 22nd Annual Conference of the Cognitive Science Society, Erlbaum*, 2000.

[Karlgren *et al.*, 2008] Jussi KARLGREN, Anders HOLST, et Magnus SAHLGREN. ≪ Filaments of Meaning in Word Space. ≫. *Lecture Notes in Computer Science, 2008, Volume 4956/2008, DOI : 10.1007/978-3-540-78646-7*$_5 2, pp 531 - -538, 2008$.

[Laf2003] *Dictionnaire Français-Anglais-Malais (FeM) - version 2, avril 2003*. CD-ROM, Dictionnaire en version XML et Application Java., 2003.

[Lafourcade & Boitet, 2002] M. LAFOURCADE et Ch. BOITET. ≪ UNL lexical Selection with Conceptual Vectors. ≫. *In proc. of LREC'2002, Las Palmas, Canary Island, Spain, May 27*, 2002.

[Lafourcade & Chauché, 1998] M. LAFOURCADE et J. CHAUCHÉ. ≪ Ficus - un agent dictionnaire coopératif et extensible ≫. Dans les actes de *NLP+IA'98, Moncton, New-Brunswick, Canada, 19-23 Août 1998*, Moncton, New-Brunswick, Canada, 1998.

[Lafourcade & Joubert, 2009] M. LAFOURCADE et A. JOUBERT. ≪ Similitude entre les sens dusage dun terme dans un réseau lexical. ≫. *Traitement Automatiques des Langues (TAL), Varia*, pp 179–200, 2009.

[Lafourcade & Joubert, 2010] M. LAFOURCADE et A. JOUBERT. ≪ Computing trees of named word usages from a crowdsourced lexical network ≫. Dans les actes de *International Multi-Conference on Computer Science and Information Technology, Wisla, Poland, 18-20 October 2010*, Wisla, Poland, 2010.

[Lafourcade & Prince, 2001a] M. LAFOURCADE et V. PRINCE. ≪ Synonymies et vecteurs conceptuels ≫. Dans les actes de *TALN'2001*, Tours, France, Juillet 2001.

[Lafourcade & Prince, 2001b] Mathieu LAFOURCADE et Violaine PRINCE. ≪ Synonymy and conceptual vectors ≫. Dans les actes de *NLPRS'2001*, pp 127–134, Tokyo, Japon, Novembre 2001.

[Lafourcade & Prince, 2004] M. LAFOURCADE et V. PRINCE. ≪ Modélisation de l'hyperonymie via la combinaison de réseaux sémantiques et de vecteurs conceptuels ≫. Dans les actes de *JADT 2004 : 7es Journées internationales d'Analyse statistique des Données Textuelles*, volume 2, pp 692–699, Louvain-la-Neuve, Belgique, Mars 2004.

[Lafourcade & Sandford, 1999] M. LAFOURCADE et E. SANDFORD. ≪ Analyse et désambiguïsation lexicale par vecteurs sémantiques ≫. Dans les actes de *TALN'99*, pp 351–356, Cargèse, France, July 1999.

[Lafourcade & Sérasset, 1998] M. LAFOURCADE et G. SÉRASSET. ≪ CGI in Lisp ≫. *MacTech magazine*, pp 25–32, 1998.

[Lafourcade & Zampa, 2009a] M. LAFOURCADE et V. ZAMPA. ≪ JeuxDeMots and PtiClic : games for vocabulary assessment and lexical acquisition. ≫. *Computer Games, Multimedia Allied technology 09. Singapore : 11th-13th may*, 2009.

[Lafourcade & Zampa, 2009b] M. LAFOURCADE et V. ZAMPA. ≪ PtiClic : a game for vocabulary assessment combining JeuxDeMots and LSA. ≫. *CICLing (Conference on Intelligent text processing and Comptational Linguistics). Mexico : 1-7 mars.*, 2009.

[Lafourcade, 1996] M. LAFOURCADE. ≪ Software engineering in the LIDIA project : distribution of interactive disambiguation and other components between various processes and machines ≫. Dans les actes de *MIDDIM-96, GETA (CLIPS, IMAG) ATR Interpreting Telecommunications (ATR), Col de Porte, France, July 1996*, Col de Porte, France, 1996.

[Lafourcade, 1998] M. LAFOURCADE. ≪ Problématique de la construction et de la distribution de dictionnaires n-lingues : exemples du projet de dictionnaire Français-anglais-malais-thaï et de l'outil ALEX. CD-ROM, Dictionnaire en version XML et Application Java ≫. *Ed. F. Renzetti, ADBS*, 1998.

[Lafourcade, 2001a] M. LAFOURCADE. ≪ Lexical sorting and lexical transfer by conceptual vectors ≫. Dans les actes de *Proceedings of the First International Workshop on Multimedia Annotation (MMA'2001), Tokyo, Japan*, page 6 pages, 2001.

[Lafourcade, 2001b] M. LAFOURCADE. ≪ Lexical sorting and lexical transfert by conceptual vectors ≫. Dans les actes de *First International Workshop on MultiMedia Annotation (MMA'2001), Tokyo, Japan, January 2001*, Tokyo, Japan, 2001.

[Lafourcade, 2002a] M. LAFOURCADE. ≪ Lens Effects in Autonomous Terminology Learning with Conceptual Vectors ≫. Dans les actes de *Seminar on linguistic meaning representation and their applications over the World Wide Web, Penang, Malaysia, July 2002*, Penang, Malaysia, 2002.

[Lafourcade, 2002b] M. LAFOURCADE. ≪ Structured Lexical data : how to make them widely avalaible, useful and reasonable protected ? - a practical example with a trilingual dictionary ≫. Dans les actes de *COLING-96, Copenhagen, Denmark, July 2002*, Copenhagen, Denmark, 2002.

[Lafourcade, 2003a] M. LAFOURCADE. ≪ Conceptual Vectors and Fuzzy Templates for Discriminating Hyperonymy is-a and Meronymy part-of relations ≫. Dans les actes de *OOIS 2003 Workshop MASPEGHI, Montréal, Canada, October 6th 2003*, Montréal, Canada, 2003.

[Lafourcade, 2003b] M. LAFOURCADE. ≪ Semantic Analysis through Ant Algorithms, Conceptual Vectors and Fuzzy UNL Graphs. ≫. *In Universal Networking Language Advances in Theory and Applications, Jesús Cerdeñosa, Alexander Gelbukh, Edmundo Tovar Eds., ISBN : 970-36-0226-6*, pp 125–137, 2003.

[Lafourcade, 2006] M. LAFOURCADE. ≪ Conceptual Vector Learning - Comparing Bootstrapping from a Thesaurus or Induction by Emergence ≫. Dans les actes de *LREC'2006, Magazzini del Cotone Conference Center, Genoa, Italia, 24-26 May 2006*, Magazzini del Cotone Conference Center, Genoa, Italia, 2006.

[Lafourcade, 2007] M. LAFOURCADE. ≪ Making people play for Lexical Acquisition. ≫. *In Proc. SNLP 2007, 7th Symposium on Natural Language Processing. Pattaya, Thailande, 13-15 December 2007*, page 8, 2007.

[Lafourcade *et al.*, 2002a] M. LAFOURCADE, V. PRINCE, et D. SCHWAB. ≪ Vecteurs conceptuels et structuration émergente de technologie ≫. *Traitement automatique des langues (TAL)*, pp 43–72, 2002.

[Lafourcade *et al.*, 2002b] Mathieu LAFOURCADE, Violaine PRINCE, et Didier SCHWAB. ≪ Vecteurs conceptuels et structuration émergente de terminologies ≫. *Traitement Automatiques des Langues (TAL)*, pp 43–72, 2002.

[Lafourcade *et al.*, 2004] M. LAFOURCADE, F. RODRIGO, et D. SCHWAB. ≪ Low Cost Automated Conceptual Vector Generation from Mono and Bilingual Resources ≫. Dans les actes de *PAPILLON-2004, Grenoble, France, 30 Août - 1 Septembre 2004*, Grenoble, France, 2004.

[Lahrizi *et al.*, 2008] Y. LAHRIZI, A. PELOV, R. PATERSON, et V. PANCALDI. ≪ Modélisation et implémentation d'un jeu de type devinette à JeuxDeMots. ≫. *TER Master 1 informatique - FMIN200 - Rapport final, encadrement M. Lafourcade, Université Montpellier 2.*, page 27, 2008.

[Landauer *et al.*, 1998] T. K. LANDAUER, P. W. FOLTZ, et D. LAHAM. ≪ An introduction to Latent Semantic Analysis. ≫. *Discourse Processes*, pp 259–284, 1998.

[Langton, 1996] C. G. LANGTON. *Artificial Life : an overview*. MIT Press, 1996.

[Larousse, 1992a] LAROUSSE, . *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Larousse, 1992.

[Larousse, 1992b] LAROUSSE, . *Thésaurus Larousse - des idées aux mots - des mots aux idées*. Larousse, Paris, 1992.

[Lesk, 1986] M. E. LESK. ≪ Automatic sense disambiguation using machine readable dictionaries : How to tell a pine cone from an ice cream cone. ≫. *In Proceedings of the SIGDOC Conference, Toronto*, pp 657–666, 1986.

[Lieberman *et al.*, 2007] H. LIEBERMAN, D.A. SMITH, et A. TEETERS. ≪ Web-based Models for Natural Language Processing ≫. *ACM Transactions on Speech and Language Processing, vol.2, n°1*, pp 1–30, 2007.

[Lopez & Zouani, 2008] C. LOPEZ et A. ZOUANI. ≪ Calculs de vecteurs conceptuels à partir de réseau lexical. ≫. *TER Master 1 informatique - FMIN200 - Rapport final, encadrement M. Lafourcade, Université Montpellier 2.*, page 21, 2008.

[Lopez, 2009] C. LOPEZ. ≪ Extraction de termes clés par vecteurs conceptuels. ≫. *Mémoire de Master 2 informatique - Rapport final, encadrement M. Lafourcade, Université Montpellier 2.*, page 69, 2009.

[Lund & Burgess, 1996] K. LUND et C. BURGESS. ≪ Producing high-dimensional semantic spaces from lexical co-occurrence. ≫. *Behavior Research Methods, Instruments Computers*, pp 203–208, 1996.

[Lund *et al.*, 1995] K. LUND, C. BURGESS, et R. A. ATCHLEY. « Semantic and associative priming in a high-dimensional semantic space. ». *Cognitive Science Proceedings (LEA)*, pp 660–665, 1995.

[Mangeot-Lerebours *et al.*, 2003] M. MANGEOT-LEREBOURS, G. SÉRASSET, et M. LAFOURCADE. « Construction collaborative dune base lexicale multilingue - Le projet Papillon ». *Traitement Automatiques des Langues (TAL)*, pp 151–176, 2003.

[Maranzana, juin 2007] Sébastien MARANZANA. « Analyse sémantique de textes par algorithmes à fourmis et combinaison de vecteurs conceptuels et de réseaux lexicaux ». *Mémoire de stage de M2R (IICW, CODA), LIRMM*, page 80 p., juin 2007.

[Mel'čuk, 1988] Igor MEL'ČUK. *Dictionnaire explicatif et combinatoire du français contemporain*, volume 2. Les presses de L'université de Montréal, Montréal, 1988.

[Mel'čuk, 1996] Igor MEL'ČUK. « *Lexical Functions in Lexicography and Natural Language Processing* », Chapitre Lexical Functions : A Tool for the Description of Lexical Relations in the Lexicon, pp 37–102. Benjamins, Amsterdam/Philadelphia, 1996.

[Mel'čuk *et al.*, 1995] Igor MEL'ČUK, André CLAS, et Alain POLGUÈRE. *Introduction à la lexicologie explicative et combinatoire*. Duculot, 1995.

[Miller *et al.*, 1990] G.A. MILLER, R. BECKWITH, C. FELLBAUM, D. GROSS, et K.J. MILLER. « Introduction to WordNet : an on-line lexical database ». *International Journal of Lexicography 3 (4)*, pp 235–244, 1990.

[Minsky, 1988] Marvin MINSKY. « *Semantic Informatic processing* », Chapitre The Society of Mind. Simon and Schuster, New York. ISBN 0-671-65713-5, 1988.

[Mitchell, 1993] M. MITCHELL. « *Analogy-Making as Perception : a Computer Model* ». PhD thesis, MIT Press Cambridge MA (USA), 1993.

[Navigli & Lapata, 2010] R. NAVIGLI et M. LAPATA. « An experimental study of graph connectivity for unsupervised word sense disambiguation. ». *IEEE Trans. Pattern Anal. Mach. Intell.*, page 678692, 2010.

[Navigli, 2009] R. NAVIGLI. « Word sense disambiguation : a survey. ». *ACM Computing Surveys*, pp 1–69, 2009.

[Navigli *et al.*, 2007] R. NAVIGLI, K. C. LITKOWSKI, et O. HARGRAVES. « Semeval-2007 task 07 : Coarse-grained english all-words task. ». *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic : Association for Computational Linguistics*, page 3035, 2007.

[Noyer & Ventalon, 2004] R. NOYER et A. VENTALON. « Extraction et classification d'informations thématiques à partir de sites Web. ». *TER Master 1 informatique - Rapport final, encadrement M. Lafourcade et D. Schwab, Université Montpellier 2.*, page 45, 2004.

[Page *et al.*, 1998] L. PAGE, S. BRIN, R. MOTWANI, et T. WINOGRAD.. « The PageRank Citation Ranking : Bringing Order to the Web. ». *Technical report,Computer Science Department, Stanford*, 1998.

[Pantel & Pennacchiotti, 2006] P. PANTEL et M. PENNACCHIOTTI. « Espresso : Leveraging Generic Patterns for Automatically Harvesting Semantic Relations ». Dans les actes de *ICCL 2006, 17th-21th July, 2006*, pp 113–130, 2006.

[Pechoin, 1991] PECHOIN, . *Thésaurus : Des idées aux mots, des mots aux idées*. Larousse, 1991.

[Puech *et al.*, 2005] M. PUECH, B. MARTINEZ, et A. CORBEEL. « Extraction de relations à partir de définitions encyclopédiques. ». *TER Maîtrise informatique - Rapport final, encadrement M. Lafourcade et D. Schwab, Université Montpellier 2.*, page 30, 2005.

[Pustejovsky, 1993] J. PUSTEJOVSKY. « The generative lexicon. ». *Computational Linguistics*, pp 409–441, 1993.

[Pustejovsky *et al.*, 1993] J. PUSTEJOVSKY, S. BERGLER, et P. ANICK. « Lexical Semantic Techniques for Corpus Analysis. ». *Journal Computational Linguistics - Special issue on using large corpora : II*, pp 331–358, 1993.

[Rehder *et al.*, 1998] B. REHDER, M. E. SCHREINER, M. B. WOLFE, D. LAHAM, T. K. LANDAUER, et W. KINTSCH. « Latent Semantic Analysis to assess knowledge : Some technical considerations. ». *Discourse Processes*, pp 337–354, 1998.

[Rodrigo, 2004] Frédéric RODRIGO. « Construction automatique d'une base d'acceptions à l'aide de dictionnaires bilingues et du modèle des vecteurs conceptuels ». Mémoire de dea, Université Montpellier II, Montpellier, 2004.

[Roget, 1852a] P. ROGET. *Thesaurus of English Words and Phrases*. Longman, London, 1852.

[Roget, 1852b] Peter Mark ROGET. *Roget's Thesaurus of English Words and Phrases*. Longman, London, 1852.

[Sagot & Fier, 2008] Benoît SAGOT et Darja FIER. « Construction d'un wordnet libre du français à partir de ressources multilingues ». Dans les actes de *TALN 2008, Avignon, France*, 2008.

[Sahlgren, 2005] M. SAHLGREN. « An introduction to random indexing. ». *In Witschel, H., ed. : Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering. Volume 87 of TermNet News : Newsletter of International Cooperation in Terminology.*, 2005.

[Salton & MacGill, 1983] G. SALTON et M. J. MACGILL. *Introduction to Modern Information Retrieval*. McGraw-Hill, New-York, 1983.

[Salton & McGill, 1983] Gerard SALTON et Michael MCGILL. *Introduction to Modern Information Retrieval*. McGrawHill, New York, 1983.

[Salton, 1991] Gerard SALTON. « The Smart Document Retrieval Project ». Dans les actes de *Proc. of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp 357–358, Chicago, IL, 1991.

[Salton *et al.*, 1975] G. SALTON, A. WONG, et C.S. YANG. « A vector space model for automatic indexing. ». *ACM*, pp 613–620, 1975.

[Schwab, 2001] Didier SCHWAB. « Vecteurs conceptuels et fonctions lexicales : application à l'antonymie ». Mémoire de dea, Mémoire de DEA, Université Montpellier II, LIRMM, Juillet 2001.

[Schwab, 2005] D. SCHWAB. « Approche hybride - lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte. ». *Thèse de Doctorat en informatique, Université Montpellier 2*, page 365 p., 2005.

[Schwab *et al.*, 2002] D. SCHWAB, M. LAFOURCADE, et V. PRINCE. « Vers l'apprentissage automatique pour et par les vecteurs conceptuels de fonctions lexiales l'exemple de l'antonymie ». Dans les actes de *TALN 2002*, Nancy, France, 2002.

[Schwab *et al.*, 2005] Didier SCHWAB, Mathieu LAFOURCADE, et Violaine PRINCE. « Extraction semi-supervisée de couples d'antonymes grâce à leur morphologie ». Dans les actes de *TALN'2005*, pp 73–82, Dourdan, France, Juin 2005.

[Schwab *et al.*, 2007] D. SCHWAB, Lim Lian TZE, et M. LAFOURCADE. « Conceptual vectors, a complementary tool to lexical networks ». Dans les actes de *NLPCS 2007 : The 4th International Workshop on Natural Language Processing and Cognitive Science, Funchal, Madeira, Portugal, 12-13 June, 2007*, Funchal, Madeira, Portugal, 2007.

[Smadja, 1993] F. SMADJA. « Retrieving collocations from text : Xtract. ». *Computational Linguistics*, pp 143–177, 1993.

[Veronis, 2001] Jean VERONIS. « Sense tagging : does it make sense ? ». *Corpus Linguistics'2001 Conference, Lancaster, U.K.*, page 10 p., 2001.

[Veyssier *et al.*, 2009] J. VEYSSIER, P. BISQUERT, B. Paiva Lima Da SILVA, et R. BELMONTE. « Développement d'un Oracle lexical. ». *TER Master 1 informatique - FMIN200 - Rapport final, encadrement M. Lafourcade, Université Montpellier 2.*, page 40, 2009.

[vonAhn & Dabbish., 2004] L. VONAHN et L. DABBISH.. « Labelling Images with a Computer Game ». *ACM Conference on Human Factors in Computing Systems (CHI)*, pp 319–326, 2004.

[Véronis & Ide, 1990] Jean VÉRONIS et Nancy IDE. « Word Sense Disambiguation with Very Large Neural Networks. Extracted from Machine Readable Dictionaries ». Dans les actes de *COLING'1990 : 13th International Conference on Computational Linguistics*, volume 2, pp 389–394, 1990.

[Zamora, 2005] Thibaud ZAMORA. « FOETAL : FOurmis et Émergence pour le Traitement Automatique des Langues ». Mémoire de dea, Université Montpellier II, LIRMM, Juin 2005.

[Zamora *et al.*, 2003] T. ZAMORA, S.BARBERIS, S. MARTIN, et C. BANCAREL. « Extraction d'informations lexicales et thématiques à partir de sites Web. ». *TER Licence IUP GMI informatique - Rapport final, encadrement M. Lafourcade et D. Schwab, Université Montpellier 2.*, page 45, 2003.

[Zampa & Lafourcade, 2009a] V. ZAMPA et M. LAFOURCADE. « Comparaison et combinaison de deux méthodes d'acquisition lexicales pour la création d'ontologies multilingues. ». *77e congrès de l'Acfas, Ottawa : 11-15 mai.*, 2009.

[Zampa & Lafourcade, 2009b] V. ZAMPA et M. LAFOURCADE. « Evaluations comparées de deux méthodes d'acquisitions lexicale et ontologique : Jeux De Mots vs Latent Semantic Analysis. ». *XVIemes rencontres de Rochebrune : ontologie et dynamique des systèmes complexes, perspectives interdisciplinaires.*, 2009.

[Zampa & Lafourcade, 2010] V. ZAMPA et M. LAFOURCADE. « PtiClic et PtiClic-kids : jeux avec les mots permettant une double acquisition ». Dans les actes de *TICE2010, 7e colloque TICE, Nancy, France, 6-8 décembre 2010*, Nancy, France, 2010.

[Zesch & Gurevych, 2009] T. ZESCH et I. GUREVYCH. « Wisdom of crowds versus wisdom of linguists measuring the semantic relatedness of words. ». *Natural Language Engineering, Cambridge University Press*, pp 25–59, 2009.

[Zhendong, 2009] Dong ZHENDONG. « Bigger Context and Better Understanding Expectation on Future MT Technology. ». *In Proceedings of International Conference on Machine Translation and Computer Language Information Processing, 26-28 June, 1999, Beijing, China.*, pp 17–25, 2009.

[Zock & Quint, 2004] M. ZOCK et J. QUINT. « Converting an electronic dictionary into a drill tutor. ». *In ICALL-2004, Venice, Italy, June 17-19*, 2004.

[Zock, 2002] M. ZOCK. « Sorry, what was your name again, or how to overcome the tip-of-the tongue problem with the help of a computer ? ». *In SemaNet workshop (Building and Using Semantic Networks), Coling, Taipei*, pp 107–112, 2002.

[Zock *et al.*, 2010] M. ZOCK, O. FERRET, et D. SCHWAB. « Deliberate word access : an intuition, a roadmap and some preliminary empirical results. ». *In In A. Neustein (éd.) International Journal of Speech Technology, 13(4), Springer Verlag*, pp 107–117, 2010.

Index

de mots, structures, acquisitions, calculs, et jeux