Interopérabilité Sémantique Multi-lingue des Ressources Lexicales en Données Liées Ouvertes

Andon Tchechmedjiev

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade, décerné par l'Université Grenoble Alpes, de

DOCTEUR ÈS SCIENCES

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Andon Tchechmedjiev

Thèse dirigée par **Gilles Sérasset**
et codirigée par **Jérôme Goulian**

préparée au sein **du Laboratoire d'Informatique de Grenoble, équipe GETALP**
et de **l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (ED MSTII)**

Interopérabilité sémantique multilingue des ressources lexicales en données lexicales liées ouvertes

Thèse soutenue publiquement le 14 Octobre 2016, ,
devant le jury composé de :

M. Eric Gaussier

PR, Université Grenoble Alpes, président, examinateur

M. Roberto Navigli

Pr., Université Sapienza di Roma, rapporteur

M. Mathieu Lafourcade

MCF, HDR, LIRMM, Université de Montpellier, rapporteur

M. Denis Maurel

Pr., Université François Rabelais, Tours, examinateur

M. Nabil Hathout

DR CNRS, HDR, CLLE/ERSS, Toulouse, examinateur

M. Gilles Sérasset

MCF, Université Grenoble Alpes, directeur de thèse

M. Jérôme Goulian

MCF, Université Grenoble Alpes, codirecteur de thèse

M. Didier Schwab

MCF, Université Grenoble Alpes, invité

In Memoriam Velitchka Pesheva (1965–2015) À ma mère.

Abstract

When it comes to the construction of multilingual lexico-semantic resources, the first thing that comes to mind is that the resources we want to align should share the same data model and format (representational interoperability). However, with the emergence of standards such as LMF and their implementation and widespread use for the production of resources in the form of lexical linked data (Ontolex), representational interoperability has ceased to be a major challenge for the production of large-scale multilingual resources. However, as far as the interoperability of sense-level multilingual alignments is concerned, a major challenge is the choice of a suitable interlingual pivot. Many resources make the choice of using English senses as the pivot (e.g. BabelNet, EuroWordNet), although this choice leads to a loss of contrast between English senses that are lexicalized with different words in other languages. The use of acception-based interlingual representations, a solution proposed over 20 years ago, could be viable. However, the manual construction of such language-independent pivot representations is very difficult due to the lack of expert speaking enough languages fluently and algorithms for their automatic constructions have never materialized, mainly because of the lack of a formal axiomatic characterization that ensures the preservation of their correctness properties. In this thesis, we address this issue by first formalizing acception-based interlingual pivot architectures through a set of axiomatic constraints and rules that guarantee their correctness. Then, we propose algorithms for the initial construction and the update of interlingual acception-based multilingual resources by exploiting the combinatorial properties of pairwise bilingual translation graphs. Secondly, we study the practical considerations of applying our construction algorithms on a tangible resource, DBNary (a lexical linked data resource extracted from Wiktionary).

Résumé

Lorsqu'il s'agit de la construction de ressources lexico-sémantiques multilingues, la première chose qui vient à l'esprit, est la nécessité que les ressources à aligner partagent le même format de données et la même représentation (interopérabilité représentationnelle). Avec l'apparition de standards tels que LMF et leur adaptation au web sémantique pour la production de ressources lexico-sémantiques multilingues en tant que données lexicales liées ouvertes (Ontolex), l'interopérabilité représentationnelle n'est plus un verrou majeur. Cependant, en ce qui concerne l'interopérabilité des alignements multilingues, le choix et la construction d'un pivot interlingue est l'un des obstacles principaux. Pour nombre de ressources (par exemple BabelNet, EuroWordNet), le choix est fait d'utiliser l'anglais, ou une autre langue, comme pivot interlingue. Ce choix mène à une perte de contraste dans les cas où des sens du pivot ont des lexicalisations différentes dans la même acception dans plusieurs autres langues. L'utilisation d'un pivot à acceptions interlingues, solution proposée il y a déjà plus de 20 ans, pourrait être viable. Néanmoins, leur construction manuelle est trop ardue du fait du manque d'experts parlant assez de langues et leur construction automatique pose problème du fait de l'absence d'une formalisation et d'une caractérisation axiomatique permettant de garantir leurs propriétés. Nous proposons dans cette thèse de d'abord formaliser l'architecture à pivot interlingue par acceptions, en développant une axiomatisation garantissant leurs propriétés. Nous proposons ensuite des algorithmes de construction initiale automatique en utilisant les propriétés combinatoires du graphe des alignements bilingues ainsi que des algorithmes de mise à jour. Dans un deuxième temps, nous étudions les implications de l'application de ces algorithmes sur DBNary (une ressource en données lexicales liées ouvertes extraite à partir de Wiktionary).

Changements apportés à la version de soutenance du manuscrit

Cette version de soutenance du manuscrit incorpore principalement des corrections de forme, d'orthographe et de grammaire, ainsi que quelques corrections d'erreurs ponctuelles :

— Suite aux remarques de R. Navigli sur l'utilisation du symbole $\sqsubset$ pour représenter la relation de raffinement, et du fait que la relation est un ordre partiel strict, nous avons remplacé le symbole $\sqsubset$ par le symbole $>$ qui est bien plus cohérent.

— (**Théorème** 3.1, **Théorème** 3.3, page 80) Correction des titres du **Théorème** 3.1 et du **Théorème** 3.3, qui démontrent l'injectivité et non la surjectivité de la relation d'appartenance à une acception et à une hiérarchie.

— (Algorithme 3.3, page 88) Dans l'algorithme de construction initiale récursif, l'appel au calcul de cliques était fait à chaque appel récursif, ce qui n'est pas nécessaire. L'appel au calcul des cliques se fait maintenant une fois au début de l'algorithme.

— (Section 3.3.1.2, pages 91-92) Correction d'approximations incorrectes pour le calcul de la complexité de calcul dans le pire des cas pour l'algorithme récursif de construction initiale des acceptions interlingues. La complexité de calcul correcte est $O(\frac{\left(\frac{n!}{(n-k-1)!}\right)^2}{k^k})$, ce qui ne change en rien les conclusions vis à vis de la nécessité de l'algorithme itératif qui reste bien plus rapide.

— Suppression des sections de références par chapitre.

— Format plus compact pour la liste des figures, tableaux, etc.

Remerciements

Je tiens tout d'abord à remercier Gille Sérasset et Jérôme Goulian, mes directeurs de thèse. Gilles pour m'avoir proposé un sujet ambitieux et pour m'avoir guidé avec enthousiasme tout au long de cette aventure de 4 ans. Jérôme pour ses innombrables conseils et recommandations et pour son soutien au travers des nombreuses démarches administratives que ce soit relative aux missions ou à l'enseignement.

Je remercie chaleureusement Didier Schwab, dont les projets de recherche à la fois innovants et passionnants m'ont initialement attiré vers le monde du traitement automatique des langues il y a de cela presque 6 ans, lors de mon stage de M1 puis de M2R. Je remercie également Didier pour les nombreuses collaborations toujours intéressantes, qu'elles soient passées ou à venir et pour les séances de brainstorming scientifique, mais aussi pour sa relecture assidue de mon manuscrit.

Je remercie Christian Boitet pour ses nombreux conseils et ses remarques, les nombreux échanges d'idées, mais aussi pour ses critiques toujours vraies, qui m'ont toujours poussé de l'avant dans mon exploration scientifique. Je le remercie également d'avoir accepté de relire mon manuscrit.

Je remercie également tous les autres collègues de l'équipe pour une ambiance toujours conviviale et de nombreuses discutions intéressantes.

Je remercie toute ma famille, en particulier ma sœur Zornitza (et Adrien), mon frère Louis, ma tante Ilka ainsi que mes grands-parents pour tout leur soutien durant cette thèse, et aussi pour leur tolérance envers les heures et jours sans fin passées à travailler.

Je remercie particulièrement Jean Alexandre, ses parents Myriam et Jean Christian ainsi que le reste de la famille pour leur accueil toujours chaleureux, leur amitié, de nombreux moments conviviaux et des discussions toujours passionnantes. *Shoutout* à Pierre en sa lointaine Californie !

Je remercie Belén, Raquel, Paola, Ruslan, Ritesh, Mohammad et tous mes amis du labo, que ce soit pour les échanges culturels et intellectuels au détour d'une pause café, les activités, pour des débats parfois enflammés ainsi que tous les moments partagés.

Je remercie mon frennemy préféré, la procrastination et naturellement par extension PhD comics et la production parfois excessive de code en résultant.

Table des Matières

Liste des figures

Chapitre 3

Chapitre 4

Liste des tableaux

Liste des définitions

Liste des axiomes

Liste des théorèmes

Acronymes

RDF ***R**esource **D**escription **F**ramework*
OWL ***W**eb **O**ntology **L**anguage*
LMF ***L**exical **M**arkup **F**ramework*
LAF ***L**inguistic **A**nnotation **F**ramework*
KAF ***K**YOTO **A**nnotation **F**ramework*
ISO ***I**nternational **S**tandards **O**rganization*
GrAF ***Gr**aph **A**nnotation **F**ramework*
ODE ***O**xford **D**ictionary of **E**nglish*
UNL ***U**niversal **N**etworking **L**anguage*
UNLKB ***U**niversal **N**etworking **L**anguage **K**nowledge **B**ase*
UW ***U**niversal **W**ords*
DEC **D**ictionnaire **E**xplicatif et **C**ombinatoire
MATE ***M**ultilevel **A**nnotation, **T**ools **E**ngineering*
GATE ***G**eneral **A**rchitecture for **T**ext **E**ngineering*
UIMA ***U**nstructured **I**nformation **M**anagement **A**rchitecture*
SPARQL ***S**PARQL **P**rotocol and **R**DF **Q**uery **L**anguage*
NIF ***N**LP **I**nterchange **F**ormat*
TAL *Traitement Automatique des Langues*
GWAP ***G**ames **W**ith **A** **P**urpose*
XML *e**X**tensible **M**arkup **L**anguage*
PWN ***P**rinceton **W**ord**N**et*
ILI ***I**nter**L**ingual **I**ndex*
UML ***U**nified **M**odeling **L**anguage*
JWKTL ***J**ava-based **W**iktionary **L**ibrary*
JWPL ***J**ava-based **W**iktpedia **L**ibrary*
WSI ***W**ord **S**ense **I**nduction*
AAS **A**lignement **A**utomatique de **S**ens
RLS **R**essource **L**exico-**S**émantiques
WSD ***W**ord **S**ense **D**isambiguation*
UL Unités Lexicales
PWMI ***P**oint-**W**ise **M**utual **I**nformation*
DVP **D**écomposition en **V**aleurs **P**ropres
FMNN **F**actorisation de **M**atrices **N**on **N**égatives
ASL **A**nalyse **S**émantique **L**atente
SVM **S**éparateur à **V**aste **M**arge

Introduction

En *Traitement Automatique des Langues* (TAL), selon la définition de Witt et al. (2009), le terme ressource langagière est un terme vaste regroupant aussi bien des données langagières (ressources statiques) que des outils permettant de les traiter (ressources dynamiques). Les formes les plus élémentaires de ressources langagières statiques sont les données langagières brutes (par exemple un corpus textuel, un signal de parole), qui n'ont subi aucune sélection particulière.

Lorsque la sélection d'un sous-ensemble est faite à partir de données brutes, pour un but, un domaine, une contrainte ou une application particulière, les données du sous-ensemble sélectionné seront appelées «ressource langagière primaire». Cette sélection peut être obtenue soit de manière experte, soit par un traitement effectué au moyen d'une ressource langagière dynamique. À ce niveau, les ressources langagières ne sont dotées d'aucune interprétation explicite.

Witt et al. (2009) présentent une autre distinction, orthogonale, des ressources langagières. D'après eux, il faut faire une distinction entre les ressources langagières à base de texte et celles à base d'entrées. Les premières incluent non-seulement, les corpus mais aussi les outils qui permettent de traiter un texte dans son ensemble (analyseur syntaxique, outils de compréhension du langage etc.). Les deuxièmes comprennent les lexiques et tous les outils qui permettent de traiter les données par entrées (par exemple, un système de désambiguïsation lexicale, un analyseur morphologique).

Lorsque l'on souhaite ajouter ou expliciter des informations relatives aux données primaires, il faut les doter d'annotations à différents niveaux en se conformant à la spécification d'une hiérarchie ou d'un empilement de types d'annotation (annotation des méta-données, annotations linguistiques). Cette annotation peut être soit experte, soit le résultat d'un traitement automatique.

Witt et al. (2009) proposent de caractériser la nature des étiquettes d'annotation (types) dans le cadre d'une méta-spécification visant à l'interopérabilité de la représentation des données (Figure 1).

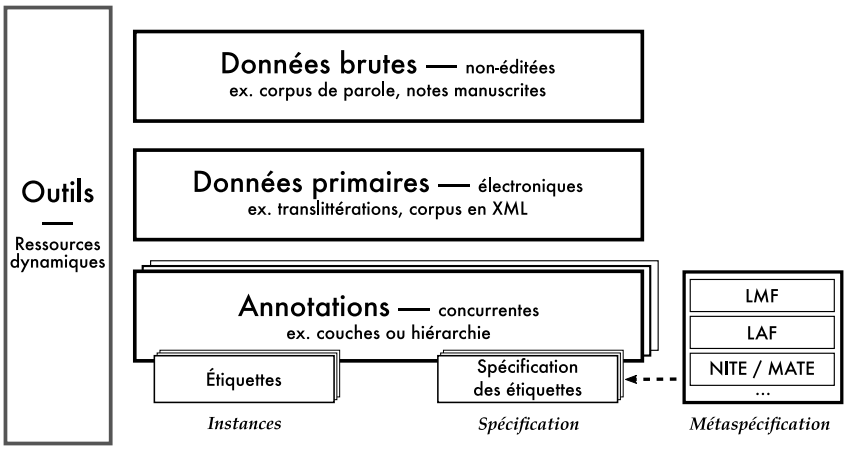


Figure 1 – Classification des ressources langagières d'après Witt et al. (2009).

Cette thèse porte particulièrement sur les ressources lexico-sémantiques ; par conséquent, nous restreindrons notre propos à ce type de ressources, même si certaines peuvent avoir une portée plus générale.

Lorsque l'on souhaite utiliser plusieurs de ces ressources ensemble, il est nécessaire de s'assurer qu'elles sont interopérables. Il existe trois niveaux d'interopérabilité (Figure 2), qui se construisent successivement, chacun sur la base des précédents [1] (Witt et al., 2009) :

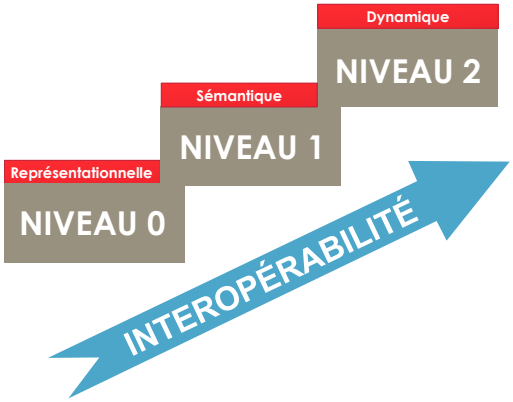


Figure 2 – Les trois niveaux d'interopérabilité pour les ressources langagières.

— *Interopérabilité représentationnelle (niveau 0)* Elle vise à s'assurer que les données des ressources sont représentées d'une manière compatible. Cela veut dire qu'il faut que les annotations associées aux données correspondent à des types d'annotation compatibles et issues d'une hiérarchie commune. Par exemple, dans le cas de deux dictionnaires, il faut s'assurer que les entrées et les sens correspondent aux mêmes éléments et que les données sont représentées et définies de la même manière : nous ne pouvons pas directement comparer des sens de mots (point de vue lexicographique [2]) à des concepts/synsets (point de vue cognitif, voir **Définition 1**) sans une étape intermédiaire définissant le passage de l'un à l'autre [3].

Définition 1 **Synset**

L'idée du *synset* est basée sur un modèle cognitif de la sémantique qui suppose que les «concepts» mentaux sont représentés par une association de mots qui peuvent apparaître ensemble dans certains contextes d'usage où leur sens est le même. Le *synset* serait alors un *synonym set* (ensemble de synonymes). Nombre de lexicographes considèrent qu'il n'existe un synonyme que si tous les contextes d'usage sont identiques, et de ce fait, les synsets ne sont pas des ensembles de synonymes, mais plutôt des ensembles de quasi-synonymes (*near-synonyms* en anglais) (Edmonds et Hirst, 2002).

Ainsi, si nous prenons l'exemple de deux dictionnaires que nous voulons aligner, il faudra d'abord que nous assurer que la représentation soit la même, c'est-à-dire que les données soient représentées dans le même format et que les entrées soient définies de la même manière.

1. C.à.d. que le niveau 1 est nécessaire pour arriver au niveau 2 et ainsi de suite.
2. Il s'agit de la vision classique en dictionnairique.
3. Ici, un sens est la lexicalisation d'un concept dans un contexte d'usage particulier.

Dans l'état de l'art se développent des standards pour les données lexicales liées ouvertes (modèle `lemon/ontolex`) sur la base des technologies du web sémantique qui gagnent en popularité. En effet la plupart des ressources majeures ont maintenant une version en `lemon`. Ces technologies rendent aisés l'alignement et l'accès aux ressources au niveau de la représentation, et sont une avancée significative dans la direction d'une représentation universelle pour les ressources lexicales (voir la **Définition 2** et le Chapitre 1).

Définition 2 **Ressource lexico-sémantique**

Une ressource lexico-sémantique est une ressource langagière à base d'entrées. Chaque entrée correspond à un mot du lexique et est associée à un ensemble de sens. Les sens dénotent les différents usages du mot ayant des significations différentes. L'ensemble des mots et des sens forment un graphe où différents mots et sens peuvent être reliés par des relations lexicales ou sémantiques (synonymie, antonymie, hypernymie, traduction, etc.).

— *Interopérabilité sémantique (niveau 1)* Lorsque les représentations des données sont compatibles, on peut vouloir aligner les éléments des différentes ressources et créer des liens entre eux (annotation d'éléments de la ressource par des éléments de la même ou d'autres ressources).
Chaque élément (phrase, mot, sens, entrée) porte des informations qui décrivent l'information sémantique portée par l'élément en question. Pour que deux éléments ou plus soient jugés équivalents, il faut s'assurer que l'information sémantique portée par les éléments soit la même.
Si des éléments portent une information plus spécifique ou plus générale, une hiérarchie d'équivalences partielles est nécessaire pour assurer une interopérabilité au niveau sémantique, qui garantit qu'il n'y aura pas de perte d'information lorsque l'on utilise les ressources ensemble.
Si l'on reprend l'exemple de deux dictionnaires, une fois que nous avons vérifié que leurs représentations sont compatibles, faut choisir comment sera représenté l'alignement entre les sens, de manière à ce que l'on puisse passer des sens d'un dictionnaire vers les sens de l'autre, idéalement sans perte des distinctions sémantiques. Le problème encore plus important lorsque l'on passe à une situation où les ressources sont toutes dans des langues différentes.

Dans l'état de l'art, nous pouvons distinguer les approches par transfert qui alignent plusieurs ressources entre elles deux à deux, et les approches par pivot qui alignent tous les sens des ressources lexicales à une structure commune.
Cependant, les approches par pivot utilisent souvent les sens d'une langue ou d'une ressource en particulier comme pivot (pivot naturel). Cela crée des pertes de contraste entre des langues présentant des distinctions sémantiques plus fines qui peuvent mener à des lexicalisations divergentes (par exemple *river* en anglais qui peut à la fois se traduire par *fleuve* et par *rivière*).
La raison principale de ce choix est qu'il est difficile de construire un pivot indépendant de la langue : il y a peu d'experts parlant suffisamment de

langues pour le construire manuellement et il n'y a pas de formalisation permettant de construire un tel pivot automatiquement (voir Chapitre 2).

— *Interopérabilité dynamique (niveau 3)* Les ressources langagières ne sont pas figées dans le temps. De nouvelles versions sortent et, dans le cas des ressources collaboratives, les changements se font de manière continue dans le temps. Lorsqu'une application utilise une ressource et souhaite, par exemple, bénéficier des ajouts de la nouvelle version, il faut s'assurer qu'une continuité est maintenue avec la nouvelle version. De plus, si l'on apporte des modifications à une ressource par un procédé quelconque, il faut pouvoir réconcilier les nouvelles informations avec de futures modifications qui seront apportées à la ressource.

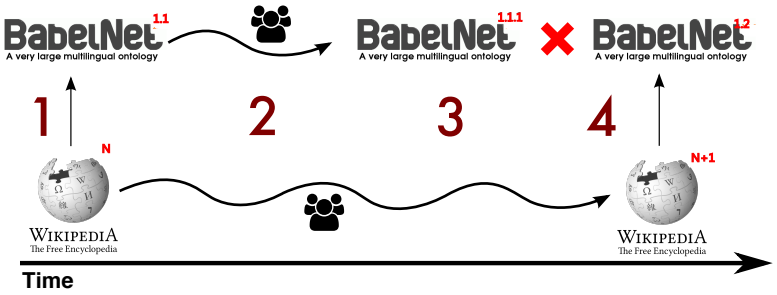


FIGURE 3 – Une illustration du besoin d'interopérabilité dynamique sur l'exemple de *BabelNet*.

La Figure 3 illustre ce problème au travers d'une situation hypothétique qui prend l'exemple de BabelNet, qui est construit en partie sur Wikipédia, une ressource collaborative.

1. Supposons que BabelNet 1.1 a été généré à une date N (étape 1),
2. et que, par la suite, il a été nécessaire d'apporter des corrections manuelles directement dans BabelNet 1.1 pour obtenir BabelNet 1.1.1 (étape 2) (NAVIGLI, JURGENS et VANNELLA, 2013),
3. puis qu'à une date $N+1$ Wikipédia a changé (ajout/suppression de pages, modifications) du fait de nouvelles contributions (étape 3),
4. et qu'enfin la version 1.2 de BabelNet a été générée automatiquement en partie à partir de cette version modifiée de Wikipédia (étape 4). Alors, les modifications manuelles faites dans BabelNet 1.1.1 doivent également être manuellement transférées dans BabelNet 1.2.

À l'heure actuelle, dans l'état de l'art, il n'existe pas de travaux portant sur une interopérabilité dynamique systématique et garantie d'un alignement de ressources lexicales, que ce soit avec une architecture pivot ou une architecture par transfert. Pour certaines **R**essource **L**exico-**S**émantiques (RLS), des mesures *ad hoc* peuvent être mises en place à chaque changement de version (voir *WordNet* (FELLBAUM, 1998) ou *BabelNet* (NAVIGLI, JURGENS et VANNELLA, 2013)).

SYNTHÈSE DES VERROUS ET CONTRAINTES. **[Contrainte 1]** Ce qui ressort de cette analyse est qu'à l'heure actuelle *il existe des standards et des formats de données permettant d'assurer une bonne interopérabilité représentationnelle des ressources lexico-sémantiques et qui sont utilisées de manière de plus en plus répandue.*

[**Verrou 1**] Cependant, pour certains types de ressources langagières, et plus particulièrement les RLS, des problèmes d'interopérabilité se manifestent au niveau sémantique lorsque l'on veut aligner plusieurs ressources entre elles, en particulier dans un contexte multilingue.

Ainsi, une architecture par transfert n'est pas viables pour un nombre important de paires de langues et pose des problèmes d'accès. Ce problème n'est pas présent avec les architectures par pivot, cependant les représentations pivot utilisées dans l'état de l'art sont biaisée car le pivot est une langue naturelle.

Cela pose des problèmes de perte de contraste dans les cas de lexicalisations divergentes. Par ailleurs, des représentations pivot artificielles ne dépendant d'aucune des langues existent, *cependant elles sont difficiles à construire manuellement*.

[**Verrou 2**] Il *n'existe pas non plus de vraie formalisation de ces représentations pour permettre la conception d'algorithmes de création automatique*.

[**Verrou 3**] Enfin, *la dimension de la dynamicité n'est à l'heure actuelle pas du tout considérée, alors que de plus en plus de RLS sont générées à partir de ressources collaboratives qui évoluent régulièrement* .

OBJECTIFS DE LA THÈSE. Au vu des verrous et de la contrainte, l'objectif de cette thèse est de proposer des solutions algorithmiques et techniques pour assurer l'interopérabilité au niveau sémantique de RLS multilingues par acceptions interlingues (pivot artificiel), et de proposer des applications et des moyens d'évaluation pratiques.

Les objectifs finals étant très ambitieux, en particulier vis-à-vis de l'application pratique et de l'évaluation, nous délimitons donc le sujet comme suit.

1. Même si les algorithmes et méthodes présentés dans cette thèse sont génériques, le côté applicatif se focalisera sur une ressource lexico-sémantique, DBNary (Sérasset, 2015). DBNary est représenté en données lexicales liées ouvertes par l'intermédiaire du modèle `lemon/ontolex` (voir la section 1.1.3) (McCrae et al., 2012).
2. Nous commencerons en nous plaçant dans un contexte formel afin de bien définir les propriétés attendues des acceptions interlingues. En effet, l'introduction initiale des acceptions interlingues les qualifie et les définit avec des termes formels, sans toutefois de spécification ou d'axiomatisation concrètes (Sérasset, 1994).
3. Même si nous ne nous intéressons pas directement à l'interopérabilité dynamique, nous veillerons à définir des opération de mise à jour compatibles avec une prise en compte ultérieure de cette dernière.

Plan du manuscrit

Afin de mener à bien les objectifs, nous étudions dans une première partie l'état de l'art relatif aux standards et aux formats pour l'interopérabilité, ainsi qu'aux algorithmes relatifs à l'alignement automatique de RLS multilingues.

Nous passons ensuite aux contributions de ce travail de thèse par rapport aux objectifs sus-cités. Nous présentons dans un premier temps les aspects formels et les algorithmes portant sur les acceptions interlingues. Dans un deuxième temps,

nous présenterons les contraintes liées à une application pratique et à une évaluation des ressources alignées avec un pivot par acceptions interlingues.

Nous résumons ci-dessous, pour chaque chapitre son contenu de manière plus détaillée.

Partie 1

CHAPITRE 1. Ce chapitre vise à présenter les standards existants pour l'interopérabilité des RLS, puis à présenter les différents types de ressources en les classant en fonction du degré d'interopérabilité sur les trois niveaux (représentationnel, sémantique, dynamique) possibles selon de l'état de l'art. Enfin, pour chaque type de ressource, le chapitre présente l'étude de cas des ressources majeures du type. La ressource utilisée dans la partie pratique de ce travail, DBNary y sera décrite de manière détaillée.

CHAPITRE 2. Ce chapitre présente d'abord l'état de l'art des algorithmes d'**A**lignement **A**utomatique de **S**ens (AAS), et en particulier les tâches similaires dans d'autre domaines, les algorithmes d'alignement en eux-mêmes, puis la question particulière de l'alignement de ressources multilingues. Il présente ensuite de manière détaillée les différentes architectures d'interopérabilité sémantique évoquées dans cette introduction, ainsi que les problèmes qui se posent avec l'approche habituelle de l'état de l'art. Il aborde enfin la question des acceptions interlingues et des différents travaux portant sur elles depuis leur introduction. Il passe notamment en revue des tentatives antérieures pour la construction automatique d'acceptions interlingues.

Partie 2

CHAPITRE 3. Ce chapitre regroupe l'ensemble des contributions théoriques concernant les acceptions interlingues. Il propose d'abord une formalisation des RLS puis, à partir des travaux antérieurs sur les acceptions interlingues, définit les axiomes et les propriétés les régissant et énonce des théorèmes portant sur les contrainte combinatoires de leur création. Le chapitre propose ensuite un algorithme de construction naïf reposant sur les propriétés combinatoires du graphe de traduction et sur la nature récursive des hiérarchies d'acceptions interlingues (tout en prenant en compte les axiomes et théorèmes). L'algorithme étant d'une complexité dans le pire des cas factorielle ($O(n^n \cdot n!)$), un algorithme plus performant est proposé, exploitant l'ordre de dégénérescence du graphe ; sa complexité dans le pire des cas est bornée par l'identification des cliques (exponentielle, $O(n^2 \cdot 3^n)$). Enfin sont proposées des techniques de mise à jour (ajout, suppression, correction) d'alignements par acceptions interlingues, dans le but de ne pas avoir besoin de trouver toutes les paires d'alignements.

CHAPITRE 4. Ce chapitre aborde les aspect plus pratiques préalables à la création d'acceptions interlingues dans le cas de DBNary. Il présente d'abord une expérience visant à ramener les relations de traduction au niveau de sens à la source en exploitant les données de Wiktionary. Ensuite, il présente les mesures de similarité translingue et les algorithmes d'alignement pertinents pour produire tous les alignements deux à deux des 21 éditions de langues de DBNary. Suit la présentation des techniques d'évaluation qui peuvent être utilisées afin de déterminer

la qualité des pivots produits, et la proposition d'un protocole expérimental pour une évaluation in vivo. Nous terminons en présentant LexSemA, la plateforme logicielle développée pour servir de base à la création d'acceptions interlingues, et qui regroupe tous les outils nécessaires.

Contributions & résultats principaux

Nous pouvons identifier les contributions et résultats principaux suivants.

1. [**Chapitre 3**] Formalisation axiomatique des RLS et des pivots par acceptions interlingues.
2. [**Chapitre 3**] Définition d'un algorithme de construction d'acceptions interlingues optimal d'un point de vue combinatoire et axiomatique.
3. [**Chapitre 3**] Définition d'algorithmes d'insertion, de suppression et de mise à jour pour les ressources alignées par acceptions interlingues.
4. [**Chapitre 4**] Rattachement des relations de traduction de DBNary aux sens source avec une qualité comparable à celle de l'état de l'art.
5. [**Chapitre 4**] Définition d'une méthode globale pour l'alignement bilingue au niveau des sens pour l'ensemble des éditions de Wiktionary.
6. [**Chapitre 4**] Définition d'un protocole expérimental visant à l'évaluation *in vivo* des ressources à base d'acceptions interlingues.

Première partie

ÉTAT DE L'ART

Chapitre 1

Standards pour l'interopérabilité des ressources langagières

Sommaire

Introduction

Avant d'aborder l'interopérabilité sémantique des RLS, il convient tout d'abord de présenter les ressources langagières et leur classification de manière générale. Nous caractérisons alors précisément les ressources lexico-sémantiques, ainsi que les différents niveaux d'interopérabilité (représentationnelle, sémantique, dynamique).

Ensuite, nous nous intéressons plus en détail à l'interopérabilité sémantique dans le cadre des ressources lexico-sémantiques en vue d'un alignement multilingue et plus particulièrement dans le cas d'alignements de trois ressources ou plus. Nous passons ensuite en revue quelques ressources lexico-sémantiques significatives soit parce qu'elles sont très utilisées soit parce qu'elles présentent des caractéristiques intéressantes par rapport à l'interopérabilité sémantique.

Pour chaque ressource, nous explicitons les problèmes d'interopérabilité sémantique et représentationnelle qui se posent.

Nous concluons enfin en synthétisant les contraintes et comment elles se placent dans le contexte scientifique et permettent d'apporter une réponse au problème de l'interopérabilité sémantique.

1.1 Interopérabilité des représentations et ressources langagières

Comme expliqué ci-dessus, le premier niveau d'interopérabilité est l'interopérabilité représentationnelle. Nous passons en revue, pour chaque type de ressource, les solutions existantes pour l'interopérabilité représentationnelle. Nous mettons ensuite en avant les solutions particulières retenues pour le présent travail et les justifications derrière ces choix. L'interopérabilité des annotations nécessite de définir un cadre et une hiérarchie de types d'annotation communs. Il faut faire la distinction entre les ressources à base de textes, les ressources à base d'entrées et les ressources dynamiques.

1.1.1 Interopérabilité des ressources dynamiques

Pour les ressources dynamiques, il existe des cadres logiciels d'annotation bien développés et robustes tels que ***U**nstructured **I**nformation **M**anagement **A**rchitecture* (UIMA) (Ferrucci et Lally, 2004) et ***G**eneral **A**rchitecture for **T**ext **E**ngineering* (GATE) (Cunningham, 2002), qui rendent possible l'établissement d'une chaîne de traitement qui permet d'enchaîner les outils dynamiques pour réaliser le traitement et l'annotation d'un texte tout en définissant un schéma d'annotation commun à toutes les ressources.

GATE est un produit commercial qui est plutôt utilisé par les linguistes alors qu'UIMA a été conçu par IBM et a ensuite été libéré et confié à la fondation Apache. Cette dernière le maintient et le met à disposition sous licence libre (licence Apache).

Ces dernières années, UIMA a commencé à devenir le standard de fait pour réaliser des chaînes de traitement. Initialement, UIMA était peu commode à mettre en place et nécessitait la création manuelle des chaines de traitement sous forme de fichiers *e**X**tensible **M**arkup **L**anguage* (XML) et ne possédait pas une documentation compréhensive et à jour. UIMA-fit fut ainsi créé pour pallier certains des problèmes de UIMA. UIMA-fit permet de spécifier les composants programmatiquement sans avoir à créer et à empaqueter les fichiers XML (Ogren et S. J. Bethard, 2009).

UIMA-fit rendit possible une création plus aisée de chaînes de composants et d'interfaces permettant un déploiement facile (par exemple sous la forme de services Web), ce qui a rendu à son tour possible la création de grandes collections de composants standardisés et aisées à importer. Un autre avantage est la possibilité récente de déployer des composants avec des systèmes de gestion de dépendances tels que `maven`.

C'est la finalité du projet *dkpro* (R. d. Castilho et Gurevych, 2014), qui offre une hiérarchie unifiée de types d'annotation et qui propose des adaptateurs pour la plupart des outils d'annotation disponibles et couramment utilisés en TAL (Stanford Parser (Klein et C. D. Manning, 2003), Open NLP [1], Mate (Bohnet, 2010), etc.).

dkpro offre une gestion automatique des dépendances au travers de `maven`, ainsi qu'un téléchargement et déploiement automatique de modèles pré-entrainés pour certaines langues (par exemple anglais, Allemand, Français, Russe).

dkpro connait une très forte popularité et est utilisé de manière relativement intensive en recherche. Par ailleurs, les campagnes d'évaluation telles que SemEval

1. https://opennlp.apache.org

(NAKOV et al., 2015), tentent de promouvoir la reproductibilité des expériences et promeuvent l'utilisation d'outils et de formats standards.

Dans le cas de SemEval il s'agit notamment de l'architecture UIMA, dont dkpro est la distribution la plus utilisée, comme en attestent les actes des dernières éditions de la campagne d'évaluation (S. BETHARD et al., 2016).

1.1.2 Interopérabilité des ressources statiques à base de textes

Pour les ressources statiques à base de textes, des initiatives telles que ***M**ultilevel **A**nnotation, **T**ools **E**ngineering* (MATE) (MCKELVIE et al., 2001) et NITE, (CARLETTA et al., 2003) proposent des systèmes d'annotation multi niveaux et de requêtes combinés, à la fois pour des corpus de texte mais aussi pour des corpus de parole.

Des systèmes tels que GATE et UIMA, de par leur nature, possèdent aussi des capacités d'annotation de texte, puisque les traitements qu'ils réalisent le long de la chaîne de traitement sont marqués sur le texte comme des annotations qui se chevauchent. Cependant, les schémas d'annotation dépendent de chaque chaine de traitement et de ses composants. De plus, il n'y a pas de garantie d'interopérabilité entre des composants qui utilisent des schémas d'annotations différents qui représentent pourtant les mêmes éléments.

Les formats d'annotation d'UIMA et de GATE ne sont pas prévus comme des formats d'échange qui permettent de transférer et de réutiliser les données d'annotation. À cette fin, il convient d'utiliser d'autres formats spécifiquement prévus pour l'échange d'annotations.

Le ***L**inguistic **A**nnotation **F**ramework* (LAF) (IDE et SUDERMAN, 2014) a été développé par le sous comité TC37 SC4 chargé de la gestion des ressources langagières de «l'***I**nternational **S**tandards **O**rganization* (ISO)» et a été standardisé en tant que ISO 24612.

L'objectif est de fournir une architecture pour l'annotation de ressources langagières qui répond aux besoins de toutes les activités en rapport aux ressources langagières en TAL (IDE et SUDERMAN, 2014). LAF possède également un format de sérialisation XML dénommé «***G**raph **A**nnotation **F**ramework* (GrAF)» qui permet un échange aisé des données d'annotation.

L'une des finalités de LAF est de pallier les manquements d'interopérabilité des architectures d'annotation et des chaînes de traitement telles que GATE et UIMA (voir ci-dessus).

Une autre finalité est de permettre de réconcilier les hiérarchies de types d'annotation entre les différents outils, qui pourtant jouent le même rôle [2].

Un désavantage de LAF est qu'il ne possède pas de sérialisation en ***W**eb **O**ntology **L**anguage* (OWL) ou ***R**esource **D**escription **F**ramework* (RDF) qui lui permettrait d'être compatible avec des annotations à partir d'ontologies et d'être interrogé au travers du langage de requêtes ***S**PARQL **P**rotocol and **R**DF **Q**uery **L**anguage* (SPARQL).

2. On peut donner l'exemple d'une annotation en «jetons» (*tokens*), où GATE et UIMA vont avoir des hiérarchies de types d'annotation qui jouent le même rôle (segmentation en phrases, en mots, lemmes, étiquettes de catégorie grammaticale) et qui ont la même sémantique, mais qui ont une représentation différente. Même au sein d'UIMA, des composants différents qui jouent le même rôle (par exemple Mate Parser et Stanford Parser) peuvent utiliser des types d'annotation différents. Pour pallier cette deuxième situation il existe d'ailleurs des tentatives d'abstraction telles que dans VERSPOOR et al. (2009).

Avec l'émergence actuelle des données lexicales liées ouvertes, les formats d'annotation supportant une représentation compatible avec OWL sont nécessaires.

Du fait de l'absence de solution viable, ouverte et disponible, s'est développé le format d'annotation ***N**LP **I**nterchange **F**ormat* (NIF) sous la forme d'une ontologie OWL (Hellmann, Lehmann et al., 2013), qui permet à toutes les applications utilisant des données lexicales liées de réutiliser les annotations à des fins diverses, par exemple l'alignement de ressources statiques (et en particulier à base d'entrées).

Ce travail de thèse se plaçant dans un contexte imposant de travailler avec des données lexicales liées, NIF est particulièrement intéressant pour exporter des annotations faites avec les entrées des ressources lexico-sémantiques produites.

1.1.3 Interopérabilité des ressources statiques à base d'entrées

Avant l'avènement de XML, les ressources lexico-sémantiques à base d'entrées n'étaient pas représentées de manière standardisée, et chaque ressource avait son propre format de représentation. Que ce soit des formats du type Microsoft Word ou Excel, des fichiers textes (séparés par des tabulations par exemple, comme c'est le cas pour ***P**rinceton **W**ord**N**et* (PWN) (Fellbaum, 1998)), ou encore des bases de données [3].

Par la suite, XML a pris de l'ampleur comme méta-format *de facto* pour la représentation des ressources au sein de nombreux projets et outils. Cependant, XML n'est qu'un méta-format et ne garantit en rien l'interopérabilité de la représentation si les spécifications du format sont différentes. Par ailleurs, beaucoup de ressources lexico-sémantiques sont des réseaux sémantiques représentés sous forme de graphe, ce que XML n'est pas adapté pour représenter.

Même s'il est possible de représenter des graphes en XML, cela nécessite une adaptation coûteuse qui nuit à la lisibilité du format. De nombreuses ressources sont encore représentées dans des formats XML *ad hoc*. L'accès à ce type de ressources nécessite l'écriture d'analyseurs et d'interfaces de programmation spécifiques à chaque ressource, ce qui mène à une multitude d'interfaces de programmation qui doivent être utilisées ensemble et autour desquelles il faut construire des convertisseurs manuellement afin d'assurer une compatibilité des ressources au niveau de la représentation.

Il existe des méta-spécifications visant à permettre de représenter tout type de ressources lexico-sémantiques dans un formalisme abstrait commun, en particulier, le formalisme LMF proposé par Francopoulo et al. (2006) et standardisé par l'ISO en tant que **ISO-24613 :2008**.

LMF vise à couvrir tous les formalismes de représentation communs et concurrents sans faire de choix : c'est à l'utilisateur de LMF de choisir quoi inclure dans chaque application.

LMF est composé d'un modèle noyau, ainsi que d'un ensemble d'extensions visant à modéliser divers phénomènes linguistiques (expressions polylexicales, sémantique par Frames, syntaxe, morphologie, etc. (Voir Figure 1.1), et en particulier

3. PWN peut également être représenté sous forme de bases de données : par exemple le *WordNet* italien issu du projet *MultiWordNet* (Pianta, Bentivogli et Girardi, 2002).

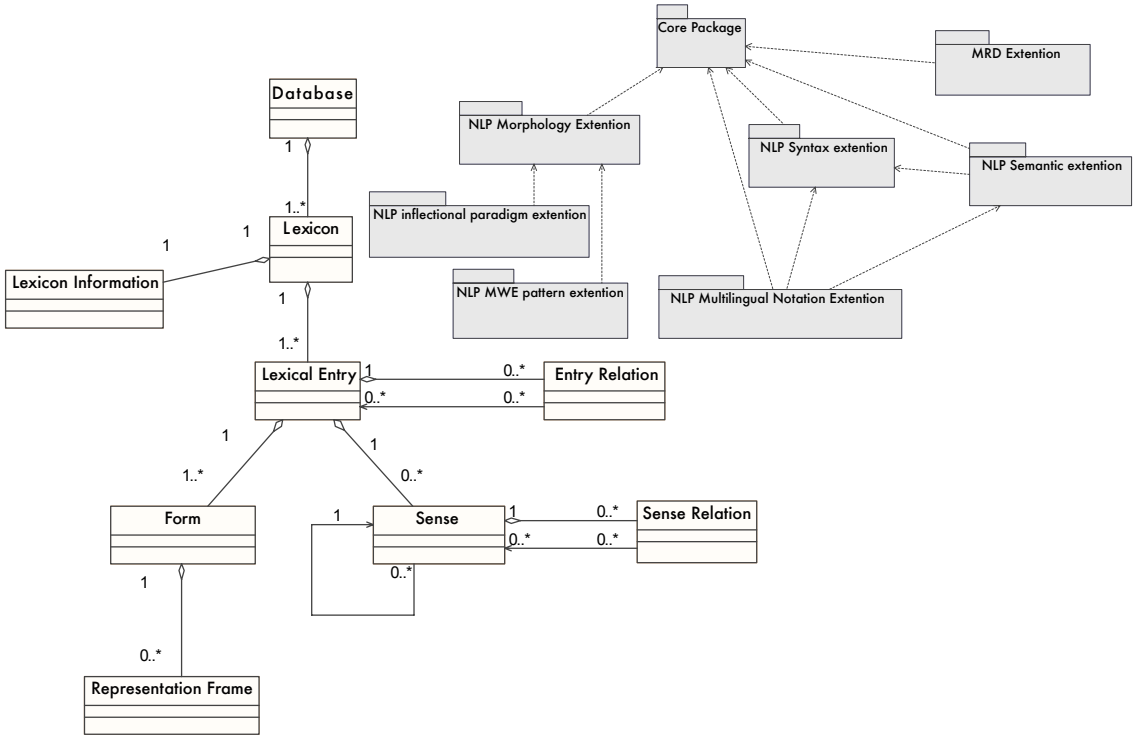


FIGURE 1.1 – Le modèle noyau LMF et ses extensions.

un module pour les notations multilingues, qui permet d'aligner des ressources multiples (multilingues ou non) ensemble à la fois par transfert et par pivot interlingue (Voir section 1.2).

Le module noyau (Figure 1.1) permet de définir un ou plusieurs lexiques (*Lexicon*) qui sont stockés dans une base de données.

À chaque lexique est associé un ensemble de méta-données (*Lexicon Information*). Un lexique (**Définition 1.1**) est composé d'entrées lexicales (*LexicalEntry*).

Définition 1.1 Lexique

«Linguistique : Ensemble des unités significatives d'une langue, excluant généralement les unités grammaticales et donc en inventaire ouvert, envisagé abstraitement comme un des systèmes constitutifs de cette langue.» [a].
En pratique, d'un point de vue informatique, il s'agit d'une liste d'entrées correspondant aux lexèmes de la langue.

a. http://www.cnrtl.fr/lexicographie/lexique

Les entrées lexicales peuvent correspondre aux lexèmes (`Word` dans LMF) (**Définition 1.2**) ou syntagmes (**Définition 1.3**) contenus dans le lexique.

Définition 1.2 **Mot-Vocable**

«Linguistique : Unité significative indépendante, ne pouvant pas toujours être déterminée selon un critère de séparabilité fonctionnelle ni par un critère de délimitation intonative» [a].

a. http://www.cnrtl.fr/definition/mot

Définition 1.3 **Syntagme**

«Linguistique : Combinaison de morphèmes ou de mots qui se suivent et produisent un sens acceptable. Le syntagme se compose donc toujours de deux ou plusieurs unités consécutives (par exemple : contre tous ; la vie humaine ; Dieu est bon ; s'il fait beau temps, nous sortirons, etc.)» (Sauss.1916, p. 170) [a].

a. http://www.cnrtl.fr/definition/syntagme

À chaque entrée lexicale sont associées une ou plusieurs formes (Par ex. : forme de surface, lemme, variantes orthographiques, etc. (**Définition 1.4**).

Définition 1.4 **Mot-Forme**

«Linguistique : Aspect sous lequel se présente un mot ou un énoncé. Forme canonique d'un mot» [a]. Un mot d'un point de vue graphique uniquement (sans aspect sémantique) [b].

a. http://www.cnrtl.fr/definition/forme
b. (Steinlin et al., 2004)

Les formes peuvent elles-mêmes potentiellement être associées à plusieurs trames de représentation (*Representation Frame*, voir **Définition 1.5**). Les entrées lexicales peuvent être reliées à une ou à plusieurs autres entités lexicales par des relations d'entités (*Entity Relation*).

Définition 1.5 ***Frame* – Cadre**

Une cadre est une structure cohérente de concepts reliés d'une manière qui impose de les connaitre tous pour en comprendre un (K. Allan, 2001). Chaque trame se compose d'un ensemble d'éléments de cadre (*Frame Elements*) qui peuvent être fondamentales (*Core Elements*) ou non (*Non-core elements)*. Chaque élément représente en quelque sorte des rôles sémantiques. C'est un formalisme de représentation pour la sémantique lexicale d'une langue. La ressource de référence pour la langue anglaise est FrameNet. L'accès aux cadres se fait au travers d'Unités Lexicales (UL), qui sont des mots dans un sens particulier. Chaque UL est illustrée par un ensemble exhaustif d'exemples d'usage annotées avec des éléments de cadre [a].

a. Source – https://en.wikipedia.org/wiki/FrameNet

Un ensemble non vide de sens (*Sense*. **Définition** 1.6) est associé à une entrée lexicale et contient les différents sens du mot ou de l'expression en question (lexemes ou phrasemes). Un sens peut lui même être divisé en sous ensembles qui correspondent à des distinctions sémantiques ou de domaine plus fines. Enfin, des sens peuvent être reliés entre eux par des relations de sens (*Sense Relation*).

Définition 1.6 **Sens – Acception**

Sens : «*Linguistique, Sémiotique : Effet de sens. Signification spécifique déterminée par le contexte et la situation. À chaque unité significative minima, correspond, dans la langue, un et un seul sens, et cela, malgré l'infinité de significations (ou effets de sens) qu'il peut avoir en fait dans le discours, et dont chacune représente un point de vue partiel, une visée particulière sur le sens.*» (Ducrot-Tod.1972, p. 160)[a].

Acception : «*Linguistique : Sens variable, nuance sémantique d'un mot suivant ses conditions d'emploi ou d'interprétation*[b]».

«*Une acception est un sens particulier d'un mot admis et reconnu par l'usage. Il s'agit donc d'une unité sémantique propre à une langue donnée (parfois appelée sémantème). C'est souvent l'unité d'un lexique monolingue.*[c]»

a. http://www.cnrtl.fr/definition/sens
b. http://www.cnrtl.fr/definition/acception
c. Sérasset, 1994

LMF assure une interopérabilité au niveau du modèle de données[4] (Francopoulo et al., 2006). Au niveau de la représentation de stockage des données, LMF propose un format de sérialisation en XML uniquement.

Il y a eu des tentatives de concevoir un format de stockage basé sur des technologies du Web sémantique, à savoir RDF et OWL, cependant le site Web de LMF indique que les efforts n'ont jamais abouti[5].

Les données liées ouvertes et la mise à disposition de données publiques sous forme d'ontologies se développent très rapidement et sont accompagnées d'une infrastructure robuste pour le stockage (bases de données de triplets/graphes) mais aussi pour l'accès aux données (langage de requêtes SPARQL).

Ainsi, un mouvement focalisé sur les données lexicales liées s'est également développé. Il vise à représenter des ressources et des données lexico-sémantiques (**Définition** 2) sous la forme d'une ontologie, de manière à pouvoir les utiliser avec d'autres ontologies existantes.

Ce mouvement a mené à la création du format lemon (McCrae, Spohr et Philipp Cimiano, 2011), qui porte le module noyau de LMF dans un langage d'ontologie (voir la Figure 1.2 pour le modèle noyau Lemon/Ontolex), puis à la création du groupe de travail W3C Ontolex qui vise une implémentation plus complète des modules présents dans LMF[6].

Cependant, un manque notable est que le support des ressources multilingues se limite à une architecture par transfert qui vise à établir des liens de traductions bilingues au niveau du sens. Ce manque se justifie par des motivations philosophiques de la communauté du Web sémantique qui considère que les concepts représentés au niveau de l'ontologie peuvent directement servir de représentation

4. LMF se contente de définir un modèle de données en ***U**nified **M**odeling **L**anguage* (UML).
5. http://lexicalmarkupframework.org/
6. http://www.w3.org/community/ontolex/wiki/Final_Model_Specification

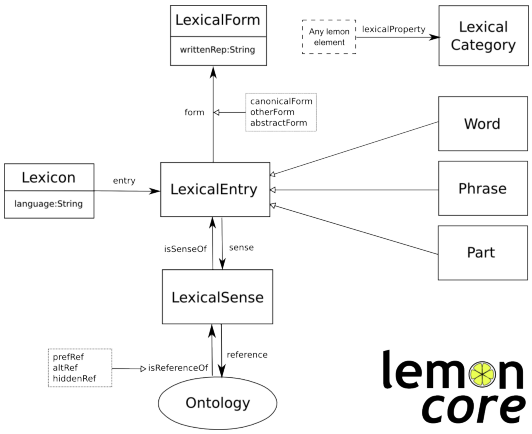


FIGURE 1.2 – Le modèle noyau lemon.

pivot pour les sens de mots. Malgré cela, le modèle Ontolex permet un stockage, une représentation et un accès interopérable aux données des ressources lexico-sémantiques à base d'entrées.

1.2 Architectures pour l'interopérabilité sémantique

Lorsque l'on veut aligner *deux* ressources, et si l'on suppose que l'interopérabilité au niveau de la représentation est assurée (schémas représentationnels compatibles), il faut faire face à deux problèmes fondamentaux.

D'une part, il faut un moyen de déterminer quels éléments sont équivalents entre les deux ressources. D'autre part, si jamais la granularité des distinctions sémantiques n'est pas la même, il faut pouvoir trouver une équivalence partielle hiérarchique entre les sens des deux ressources.

Pour illustrer ces deux problèmes, nous allons prendre l'exemple de l'entrée anglaise `race#noun` dans *WordNet* et dans l'***O**xford **D**ictionary of **E**nglish* (ODE) (STEVENSON, 2010), comme le fait Roberto NAVIGLI (2006). Mais avant, il convient de cerner les différences de structure entre *WordNet* (FELLBAUM, 1998) et ODE, en sachant que les ressources seront introduites et présentées de manière plus détaillée dans la section 1.3 du présent chapitre. Dans l'ODE, la structure des sens est hiérarchique. La distinction des sens principaux[7] se fait entre autres sur la base de domaines d'usage différents, et pour chacun d'eux, il y a des distinctions de sens plus fines (souvent d'ordre pragmatique), parfois illustrées par des exemples d'usage.

Il se peut que ces distinctions plus fines correspondent à des domaines particuliers alors que le sens principal est de domaine général (**Définition 1.7**)[8].

En ce qui concerne *WordNet*, la plupart des sens correspondent au domaine général et il y a peu de distinctions plus fines de sens. *WordNet* en lui-même ne contient pas d'étiquettes de domaine attachées aux sens et il faut utiliser des ressources externes pour pallier cette limitation (MAGNINI et CAVAGLIÀ, 2000).

Chacun des sens de *WordNet* aura potentiellement une granularité bien plus fine que les sens de ODE. Si l'on souhaite aligner les deux ressources, il faut donc

7. Sens listés au niveau le plus haut directement sous l'entrée du dictionnaire.
8. Nous pouvons voir cela comme un partitionnement en sous ensembles du sens. Certains des sous ensembles correspondent à des distinctions sémantiques spécifiques de domaine particuliers.

Définition 1.7 Domaine général

Un texte appartenant au domaine général est écrit en langue générale, telle qu'elle est employée dans la vie courante sans jargon particulier. Cela inclut habituellement le registre de langue employée dans la presse grand public, à la télévision, dans les conversations de tous les jours.

pouvoir faire correspondre les sens de *WordNet* à un ou plusieurs sous sens dans ODE. Une fois cette identification faite, nous aurons également une correspondance entre les sens *WordNet* et les sens de niveau supérieur.

La Figure 1.3 présente l'exemple utilisé par Roberto NAVIGLI (2006) auquel nous sommes venus ajouter un alignement possible entre les deux ressources. Ici nous pouvons voir que plusieurs sens issus de *WordNet* correspondent au même sens principal dans ODE. Mais en même temps, dans *WordNet* il n'y a pas de sens correspondant aux sens principaux de ODE correspondant aux domaines Nautique et Botanique, ainsi que toutes les sous distinctions de sens.

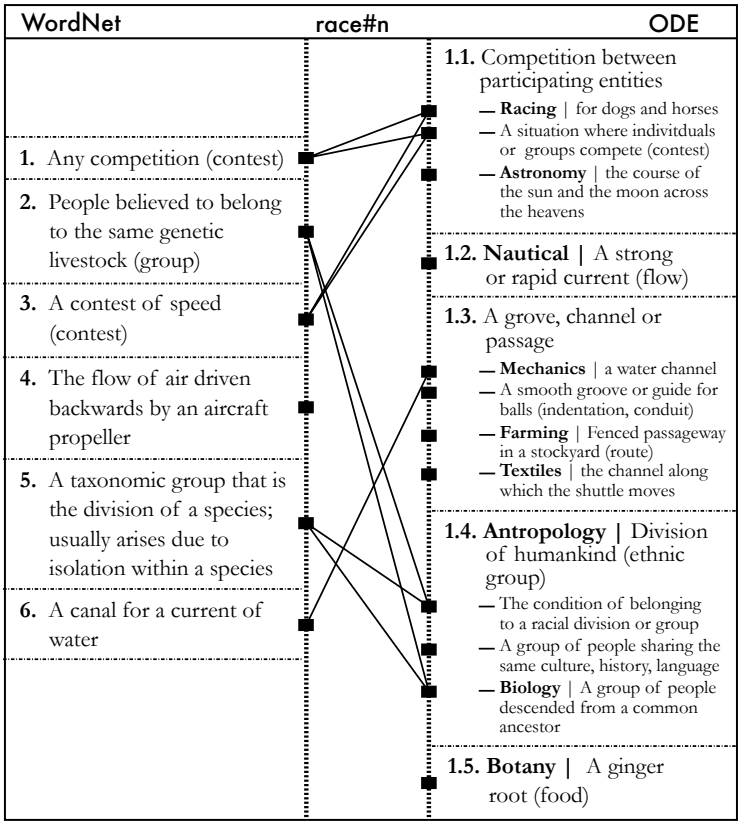


FIGURE 1.3 – Un exemple de granularités divergentes pour l'entrée correspondant au nom commun anglais *race* provenant de deux ressources lexicales en langue anglaise (*WordNet* 2.1 et Oxford Dictionary of English), avec un alignement potentiel entre les sens des deux ressources.

L'ODE n'a pas de sens principal issu du domaine de l'aviation contrairement à *WordNet*. Si jamais *WordNet* avait la même granularité d'ODE, *WordNet* n'aurait que 4 sens.

Si un lien d'équivalence existe entre deux sens principaux, il n'y a aucune garantie qu'il s'appliquera forcemment à toutes les distinctions plus fines et en par-

ticulier aux distinctions portant sur des domaines particuliers. Ainsi il est utile de définir des sous-liens d'équivalence pour les distinctions plus fines afin de préserver les nuances [9].

Dans ce cas on dit que l'un des sens contient [10] l'autre du point de vue de l'information sémantique qu'il porte.

Lorsque les deux ressources sont dans deux langues différentes, les problèmes de granularité et d'équivalence sémantique partielle persistent. Vient s'ajouter le fait que nous ne pouvons plus directement comparer toutes les informations sémantiques portées par les éléments des ressources. Il faut une étape supplémentaire qui visera à projeter les informations sémantiques des deux langues dans un espace commun où elles sont comparables (traduction, système translingue, équivalences structurelles).

Les problèmes de granularité et de divergence sémantique sont amplifiés par le fait que les conceptualisations dans une langue (ainsi que les lexicalisations desdits concepts) se forment de manière différente au fil du temps et ne se correspondent souvent pas, surtout entre des langues s'étant développées dans des environnements culturels très différents.

Dès que l'on s'intéresse à des alignements de trois ressources ou plus, il faut de plus définir un paradigme d'alignement pour gérer l'ensemble des ressources alignées et garder une consistance entre tous les alignements. Il existe deux philosophies/architectures pour l'alignement entre plus de deux ressources. Une architecture par transfert (alignements bilingues entre toutes les paires de ressources, une généralisation naïve de l'alignement de deux ressources) et une architecture par pivot interlingue qui impose à trouver une représentation interlingue commune pour représenter les alignements des sens entre toutes les ressources à la fois. Les sous-sections suivantes détaillent les deux approches ainsi que leurs limitations et définissent la portée du présent travail.

1.2.1 Architecture par transfert

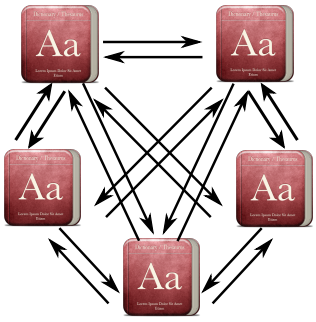


FIGURE 1.4 – Une architecture par transfert pour l'alignement de trois ressources lexicales ou plus.

L'architecture par transfert (Figure 1.4) consiste à établir des liens de correspondance directs entre les ressources deux à deux, par transposition directe de

9. C'est le cas ici quand des sens *WordNet* ne couvrent pas tous les sous sens d'un sens dans ODE.

10. « Action de penser le particulier sous le général ; résultat de cette action » – http://www.cnrtl.fr/definition/subsumer, ce qui ici veut dire que l'information sémantique contenue dans l'un est englobée par l'information sémantique contenue dans l'autre.

l'alignement de deux ressources entre elles. Il est nettement plus aisé d'éviter les difficultés liées à l'alignement de plus de deux ressources en faisant comme si il n'y en avait que deux localement. Pour chaque nouvelle ressource que nous ajoutons à la ressource alignée, il fait établir, pour chacun des élément, l'alignement avec toutes les autres ressources, déjà présentes dans la base.

Pour un petit nombre de ressources qui évoluera pas ou peu, une approche par transfert est envisageable. Cependant, dans le cas où l'on veut mettre en correspondance un grand nombre de ressources, qui évoluent régulièrement, cette philosophie est sous-optimale, car le nombre de combinaisons à établir croit en fonction du carré du nombre de ressources, comme nous le verrons ci-après.

Les corpus parallèles/comparables et les dictionnaires bilingues sont des exemples d'interopérabilité par transfert.

1.2.1.1 Inconvénients des architectures par transfert

Dans le cas général, si l'on veut aligner n ressources, il faut relier chaque ressource à $(n-1)$ autres ressources. À savoir qu'il faut procéder aux alignements sur toutes les paires d'éléments.

Dans le cas particulier de l'alignement de ressources lexico-sémantiques structurées (réseaux sémantiques) au niveau du sens, le problème prend une envergure encore plus importante, car il est nécessaire d'établir $(n-1)$ liens pour chaque sens de mot avec des liens de traduction unidirectionnels et $\frac{n(n-1)}{2}$ liens de traduction bidirectionnels, où n est le nombre de langues ou le nombre de ressources monolingues à aligner.

Si nous nous plaçons au niveau de l'alignent des sens appartenant à deux mots dans deux langues différentes et en supposant un nombre de sens p égal pour les deux mots, il y aura $p!$ alignements potentiels de sens. Ainsi à l'échelle de la ressource, pour un mot donné, on se retrouve avec $(p!)(n-1)$ liens potentiels à évaluer. Nous supposons que chaque mot de la langue source à au moins une traduction.

N.B. Ce n'est qu'une hypothèse pour simplifier le calcul et qui n'est donnée qu'à titre d'illustration : ce nombre varie grandement de langue en langue et en fonction de la granularité des ressources, du domaine et du texte lui même.

Le degré de polysémie moyen dans Wordnet est de 1,5 (toutes parties du discours confondues), cependant, en pratique, dans un texte, il peut être bien plus élevé pour certains types de catégories grammaticales (par exemples les verbes) et surtout pour les mots les plus courants (exemple extrême, le verbe être en anglais, *to be*, qui contient plus de 30 sens). Pour un exemple, voir (Roberto NAVIGLI, LITKOWSKI et HARGRAVES, 2007).

Non seulement il y a un grand nombre de paires à traiter, mais plus les langues que l'on traite sont proches, plus il y aura de duplication au niveau des alignements, car dans de nombreux cas, les alignements risquent d'être identiques. Par conséquent, quand le nombre de ressources est grand, la construction manuelle d'alignements par transfert n'est pas réaliste.

1.2.1.2 Avantages des architectures par transfert

Un avantage indéniable des architectures par transfert, est que lors de la construction manuelle, il suffit d'avoir des experts bilingues pour obtenir des alignements de qualité.

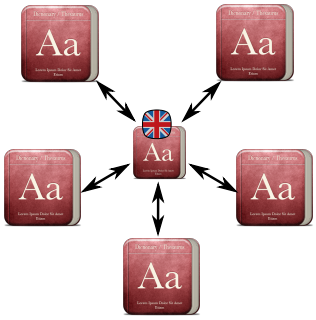


FIGURE 1.5 – Une architecture par pivot naturel pour l'alignement de trois ressources lexicales ou plus.

Cela est particulièrement vrai dans un contexte de production participative (**Définition** 1.8), il est plus aisé et moins cher de trouver des travailleurs compétents pour réaliser la tâche d'alignement par rapport à une architecture à base de pivot interlingue comme nous le détaillerons ci-dessous.

Définition 1.8 Production participative

La production participative (*crowdsourcing* en anglais) consiste à exploiter les efforts de contributeurs (bénévoles ou faiblement rémunérés) en grand nombre au travers d'Internet.

1.2.2 Architecture par pivot naturel

Une alternative à une architecture par transfert est d'utiliser une représentation par pivot interlingue qui va mettre en relation les éléments équivalents venant des différentes ressources à aligner. Un pivot est une généralisation sur les représentations sémantiques des éléments alignés issus des différentes ressources (WITT et al., 2009).

Il existe deux types de pivots, le pivot naturel, qui utilise une ressource dans une langue particulière comme pivot (par exemple *WordNet*) et un pivot interlingue (dit artificiel) qui ne dépend d'aucune des langues alignées et qui est traité dans la section suivante.

L'utilisation d'une langue riche en ressources comme pivot (pivot naturel) est une option souvent adoptée à défaut d'avoir les outils et les moyens de produire un vrai pivot interlingue (artificiel). La Figure 1.5 illustre schématiquement une architecture par pivot naturel (à comparer avec la Figure 1.4 pour une architecture par transfert).

1.2.2.1 *Inconvénients de l'architecture par pivot naturel*

La sémantique de chaque langue se développe en fonction de son histoire, de sa culture et de sa société et les sens de mots vont avoir des lexicalisations divergentes (CAMPENHOUDT, 2000). Ainsi, l'utilisation d'un pivot naturel introduit des phénomènes contrastifs artificiels (**Définition 1.9**).

Définition 1.9 **Phènomène contrastif artificiel**

Un **phénomène contrastif** est un ensemble d'éléments qui permettent de faire apparaitre des distinctions entre des sens qui ne sont pas complètement équivalents d'une ressource à une autre.
Un **phénomène contrastif artificiel** correspond à une perte d'information discriminatoire lors d'un alignement qui ignore ces distinctions lorsque l'on utilise un pivot naturel (langue naturelle). En effet une langue naturelle aura une conceptualisation et des lexicalisations divergentes par rapport aux autres langues contenues dans la ressource multilingue (SÉRASSET, 1994). Voir exemple et Figure 1.6.

Nous allons maintenant illustrer les problèmes de phénomènes contrastifs artificiels. En effet, la granularité des distinctions de sens est souvent liée à la prévalence des usages de certains mots et concepts dans la vie quotidienne. Pour les distinctions les plus fréquentes, des mots dédiés sont créés. Il y a de nombreux exemples de lexicalisations divergentes répertoriés par les linguistes (CAMPENHOUDT, 2000) [11].

Prenons l'exemple du mot riz. En français et en anglais et dans d'autres langues romanes, il correspond à un mot et à un concept unique, dont les différents contextes d'usage déterminent les distinctions en sens du mot (les grains de la plante, les plants de riz, le riz en tant que denrée alimentaire) alors que dans de nombreuses langues asiatiques, il y a des concepts et des lexicalisations différentes pour chacun de ces sens.

Ainsi, en japonais, riz cuit (l'aliment) se transcrit par `gohan` alors que le riz cru (la graine) se transcrit par `kome`.

La même distinction se rencontre dans la langue bangalaise (parlée au Bangladesh et au Bengale Oriental en Inde) où riz cuit se transcrit par `bhat` alors que riz cru se transcrit par `dhan`.

Si les concepts et sens de l'anglais sont choisis comme pivot, et que l'on aligne à ce pivot anglais des sens venant de plusieures langues asiatiques, alors, nous perdrons les distinctions de sens entre les lexicalisations divergentes d'un même mot et nous ne pourrons plus savoir que la lexicalisation `bhat` du bengali correspond à la lexicalisation japonaise `gohan`. La Figure 1.6 illustre le problème.

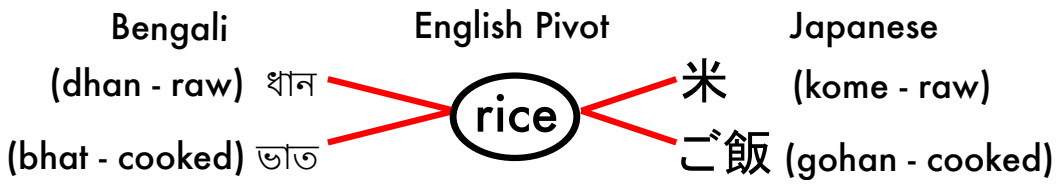


FIGURE 1.6 – Un exemple de phénomène contrastif artificiel.

11. Il n'en existe cependant pas de quantification précise.

Ainsi, les contrastes artificiels introduits rendront la ressource inutilisable dans nombre d'applications, notamment la traduction automatique. En effet pour la traduction automatique, si le contraste entre les lexicalisations divergentes est perdu (Figure 1.6), le choix lexical que fera le système de traduction aura une chance sur n (pour n lexicalisations possibles) d'être erroné (`kome` aura deux traductions possibles en bengali) sans moyen de savoir laquelle est la bonne), ce qui peut radicalement changer le sens d'une phrase.

1.2.2.2 Avantages de l'architecture par pivot naturel

L'utilisation d'un pivot naturel, surtout dans une langue fortement dotée (**Définition 1.10**), rends la construction beaucoup plus aisée car il n'est en théorie nécessaire que d'avoir des locuteurs bilingues entre chacune des langues et la langue du pivot pour établir le lien. Un autre avantage du pivot naturel est que l'on évite l'explosion de combinaisons pour l'alignement de ressources entre elles par rapport à une architecture par transfert mais aussi (de manière moindre) par rapport à un vrai pivot interlingue.

Définition 1.10 **Degré de dotation d'une langue**

Le **degré de dotation** d'une langue quantifie la disponibilité de ressources langagières informatisées dans cette langue vis-à-vis de différents critères allant de la saisie et de l'affichage des caractères, à la disponibilité de dictionnaires ou encore de ressources dynamiques (détection des catégories grammaticales, analyseurs morphologiques, etc.).
Une **langue peu dotée**, de manière générale manque de certains outils élémentaires. Même si dans beaucoup de cas la saisie de caractères est possible, des ressources telles que des corpus, des données d'entrainement ou même des dictionnaires électroniques peuvent être manquantes. Nous pouvons citer la langue somalie comme un exemple typique d'une langue peu dotée.
À l'opposé, une **langue fortement dotée** (par exemple l'anglais) va posséder divers types de ressources langagières en abondance.
La notion de dotation est dépendante de l'application considérée. Si l'on explore un nouveau type d'application en langue anglaise pour laquelle il n'existe aucune ressource, alors l'anglais est considéré comme une langue peu dotée vis-à-vis de cette application en particulier (Berment, 2004).

1.2.3 Architecture par pivot artificiel

Une architecture par pivot interlingue artificiel, n'utilise pas une langue en particulier comme pivot mais construit une structure hiérarchique interlingue prenant en compte les divergences de lexicalisations afin d'éviter les contrastes artificiels. La Figure 1.7 illustre d'une manière schématique le principe d'une architecture par pivot interlingue (à comparer à la Figure 1.4 pour les architectures par transfert et à la Figure 1.5 pour une architecture à pivot interlingue naturel).

1.2.3.1 Inconvénients des architectures par pivot artificiel

L'inconvénient principal des architectures à base de pivot interlingue artificiel, est qu'il est beaucoup plus difficile de construire un alignement entre N sens.

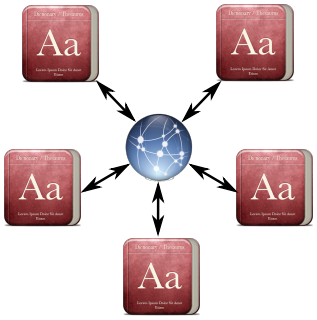


FIGURE 1.7 – Une architecture par pivot interlingue pour l'alignement de trois ressources lexicales ou plus.

En effet, pour pouvoir juger de l'équivalence entre les sens, il faut de préférence recourir à des contributeurs qui parlent plusieurs langues simultanément, même si il existe des moyens détournés de construire ce pivot indirectement. Par ailleurs, il est difficile d'animer une communauté de collaborateurs pour les pousser à participer activement.

1.2.3.2 Avantages des architectures par pivot artificiel

Une ressource lexicale fondée sur une représentation par pivot interlingue artificiel, peut jouer un rôle dans la construction de systèmes de traduction fondés sur des interlingua intermédiaires (**Définition** 1.11).

Définition 1.11 Interlingua

En traduction automatique une interlingua est une langue artificielle intermédiaire entre la langue source et la langue cible[a].

a. https://fr.wikipedia.org/wiki/Interlangue_(traduction_automatique)

Nous pouvons citer l'exemple de systèmes de traduction fondés sur l'***U**niversal **N**etworking **L**anguage* (UNL) [12].

UNL est un langage d'hypergraphes qui permet d'encoder du texte en langue naturelle dans une représentation interlingue indépendante de toute langue et qui nécessite la présence d'une base de connaissances (***U**niversal **N**etworking **L**anguage **K**nowledge **B**ase* (UNLKB)).

Cette dernière permet de représenter des informations lexico-sémantiques, indépendamment d'une langue en particulier. Les lexèmes multilingues qui forment l'espace lexical d'UNL sont des ***U**niversal **W**ords* (UW), qui sont alignés à la manière d'un pivot interlingue (MARTINS et GRAÇAS VOLPE NUNES, 2005).

Ainsi, l'UNLKB est une ressource multilingue fondée sur une architecture de pivot interlingue. De plus, une de ces évolutions, *UNL Ontology* est à l'origine de la plateforme Jibiki-Pivax (NGUYEN, BOITET et SÉRASSET, 2007), qui est un ensemble d'outils, permettant la mise en place de bases de données lexicales au travers d'une architecture pivot.

12. http://www.undl.org/ (UCHIDA et ZHU, 2001)

1.3 Ressources langagières multilingues existantes et interopérabilité

Il existe un grand nombre de ressources langagières et particulièrement lexicales et sémantiques, que ce soit sous une forme papier traditionnelle ou sous forme électronique.

Mise à part la classification des types de ressources langagières présentée dans la section précédente, il convient de faire une autre distinction majeure, qui est le mode de construction de la ressource.

Nous pourrons ainsi distinguer deux familles de méthodes de construction : la construction manuelle et la construction automatique.

Pour les ressources de chaque catégorie, nous passerons en revue quelques exemples saillants de l'état de l'art, et nous procéderons à une analyse de l'interopérabilité potentielle avec d'autres ressources ainsi que les architectures d'interopérabilité utilisées pour les ressources multilingues en particulier.

1.3.1 Ressources construites manuellement et collaborativement

Ce type de ressource est traditionnellement construite par des experts (e.g. lexicographes ou psychologues cognitifs). La construction manuelle par des experts coûte très cher, mais est aussi limitée par la taille ainsi que par le nombre de langues qui pourront être couvertes.

En effet, le nombre de langues que parle l'expert est limité. Une alternative à la construction par des experts, est la construction par production participative (**Définition 1.8**) au travers d'Internet. Ainsi, nous présenterons d'abord les ressources traditionnelles que sont les dictionnaires (monolingues/bilingues), puis nous nous intéresserons aux *WordNet*s , aux ressources collaboratives et enfin au projet Kyoto, qui offre une architecture hybride (manuelle/automatique).

1.3.1.1 Dictionnaires et thesauri

Les dictionnaires (**Définition 1.12**) sont typiquement des ressources pouvant aisément être représentés par des normes telles que LMF, ainsi que ses implémentations en OWL dans Ontolex. L'alignement de deux dictionnaires, en particulier dans des langues différentes, produit des dictionnaires bilingues, selon une architecture par transfert.

Définition 1.12 **Dictionnaire**

«Recueil des mots d'une langue ou d'un domaine de l'activité humaine, réunis selon une nomenclature d'importance variable et présentés généralement par ordre alphabétique, fournissant sur chaque mot un certain nombre d'informations relatives à son sens et à son emploi et destiné à un public défini»[a]

a. http://www.cnrtl.fr/definition/dictionnaire

Les thésaurus (**Définition** 1.13) sont une autre sorte de dictionnaire, qui listent les relations entre les mot dans une langue particulière (par exemple des synonymes, des antonymes, etc.).

Définition 1.13 Thésaurus

«Documentologie : Langage documentaire fondé sur une structuration hiérarchisée d'un ou plusieurs domaines de la connaissance et dans lequel les notions sont représentées par des termes d'une ou plusieures langues naturelles et les relations entre notions par des signes conventionnels» [a]

a. http://www.cnrtl.fr/definition/thésaurus

La combinaison d'un thésaurus avec une dictionnaire multilingue permet sous certaines conditions d'obtenir une équivalence entre des relations (**Définition** 1.14) dans différentes langues en exploitant les alignements bilingues (transfert).

Définition 1.14 Relation lexico-sémantique

Lien de nature sémantique entre deux entités lexico-sémantiques d'une langue, par exemple la synonymie, l'hyperonymie, la métonymie, etc.).

Dans un cadre plus général, des thésaurus pourraient être ajoutés à de nombreux types de ressources, y compris des ressources alignées selon une architecture par pivot interlingue, ce qui viendrait enrichir le graphe en relations.

L'ODE (Stevenson, 2010), le dictionnaire cité comme exemple dans la Section 1.2, est un dictionnaire populaire qui suit le modèle de construction lexicographique classique et qui existe aussi sous forme électronique. Même si de nombreux dictionnaires dans des formats électroniques sont fondés sur des représentations en XML, un nombre important de projets passent à des représentations fondées sur Ontolex/lemon.

C'est le cas de ODE, mais aussi des dictionnaires de la collection Kdictionary, notamment avec la participation d'équipes de Kdictionary et d'Oxford University Press à des événements tels que le datathon SD-LLOD 2015, dont l'objectif était précisément de convertir des ressources en lemon/Ontolex et de les aligner avec d'autres ressources [13].

Il existe d'autres formalismes pour la représentation des dictionnaires, par exemple fondés sur la théorie sens-texte et son implantation dans le **D**ictionnaire **E**xplicatif et **C**ombinatoire (DEC) du Français (Mel'čuk, 1999). Les données de base nécessaires au formalisme sens-texte peuvent être représentées avec des standards tels que LMF.

Des initiatives de ressources électroniques sur la base de la théorie sens-texte existent également, et ont été construites à l'initiative de A. Polguère, collaborateur de I. Mel'čhuk, le créateur du formalisme. Nous pouvons citer DiCo, qui est un projet de dictionnaire combinatoire électronique.

DiCo est basé sur une structure lexicographique très classique (Steinlin et al., 2004), où les entrées sont des vocables, qui contiennent eux même des lexies/significations. Il en existe des traductions dans un modèle de données objet spécifique

13. Ces travaux n'ont pas encore fait l'objet de publications, cependant les informations relatives aux objectifs du Workshops ainsi que de nombreuses ressources sont disponibles sur le site Web http://datathon.lider-project.eu/

à DiCo (DiCObject), ainsi qu'une conversion en XML dans un format compatible avec LMF dans le contexte du projet Papillon (Lapalme et Sérasset, 2003), une base multilingue à base de pivot interlingue.

1.3.1.2 Wordnets et EuroWordnet

PWN est une base de données lexicale (Fellbaum, 1998), dont le but était initialement de modéliser le lexique mental humain par association de mots (amorçage sémantique). PWN s'organise autour de la notion de *synsets*, qui contiennent un ensemble de mots, qui mis ensemble, évoquent un concept mental.

PWN n'était pas initialement destiné à être utilisé comme ressource lexico-sémantique, mais uniquement pour des recherches en psychologie cognitive. Cependant, PWN a rapidement été récupéré par des chercheurs en traitement automatique des langues et utilisé comme ressource lexico-sémantique pour un grand nombre de projets.

Les distinctions de sens n'existaient pas dans les premières versions de PWN, mais ont été ensuite ajoutées comme l'intersection d'un mot et du *synset* contenant ce mot. On dit que les mots du *synset* lexicalisent le concept dans un contexte lexical défini par les autre mots du *synset*.

La dernière version de PWN, 3.0, contient 155.287 entrées avec 117.659 *synset*s et 206.941 paires sens-mots. Pour la représentation des données, WordNet utilise un ensemble de fichiers texte séparées par des tabulations, mais il s'agit en réalité d'un schéma relationnel qui peut aisément être stocké dans un système de gestion base de données du même type.

En terme d'interopérabilité sémantique, PWN ne propose pas directement de mécanisme d'alignement, cependant, il est utilisé pour des annotations sémantiques de texte et de corpus, ce qui implique que les corpus annotés sont uniquement compatibles avec d'autres corpus et ressources également annotés avec WordNet.

WordNet a rapidement été utilisé dans de nombreuses applications et en particulier lors des premières campagnes d'évaluation SenseEval pour la désambiguïsation lexicale, comme inventaire de sens et ressource de référence pour produire des corpus d'évaluation annotés en sens.

C'est en particulier le cas de SemCor (Fellbaum, 1997), qui est une version du Brown Corpus (Francis et Kucera, 1979) annotés en sens issus de *WordNet*. Les corpus ont ensuite été réutilisés pour de nombreuses tâches nécessitant d'avoir du texte désambiguïsé ainsi que pour la création de systèmes de désambiguïsation lexicale.

WordNet est ainsi devenu un standard *de facto*, et de nombreuses initiatives visant à produire des WordNets dans d'autres langues sont apparues. EuroWordNet (Piek. Vossen, 1998) en est un exemple. Le projet visait à financer la construction de *WordNet*s dans plusieures langues européennes (hollandais, espagnol, italien, anglais, français, allemand, tchèque, estonien) [14].

Vis-à-vis de l'interopérabilité représentationnelle, les EuroWordNets sont tous représentés sous la forme d'une base de données avec un schéma commun. Les éditions de langues sont toutes alignées entre elles et le choix a été fait d'utiliser

14. http://www.illc.uva.nl/EuroWordNet/objectives-ewn.html

PWN comme pivot interlingue. Les alignements (relations d'équivalence) entre *synset*s sont regroupés dans un index translingue (***I**nter**L**ingual **I**ndex* (ILI) anglais) et peuvent correspondre à des relations sémantiques.

Il s'agit ici d'une architecture interlingue par pivot naturel (d'anglais). Lorsque des concepts n'existent pas en Anglais alors le concept de l'une des langues où il est présent est utilisé comme pivot. Au travers OpenMultilingual WordNet, une partie des données d'EuroWordNet sont disponibles au format lemon.

1.3.1.3 Ressources collaboratives

Un inconvénient de la construction manuelle de ressources lexicales par des experts est qu'elle coûte cher et qu'elle prend du temps. Par ailleurs, il n'y a jamais de consensus absolu parmi les lexicographes vis-à-vis des mots présents dans la ressource ou vis-à-vis des distinctions de sens.

Il en va de même pour les encyclopédies. Cela rend l'accès à ce type de ressources potentiellement difficile. Le développement du Web a permis l'apparition de projets tels que Wikipedia et Wiktionary, qui ont profondément changé la manière dont nos sociétés accèdent à l'information (VANDENDORPE, 2008).

La logique derrière de tels projets est que même si les contributeurs ne sont pas des experts et même si les contributions individuelles ne sont pas d'aussi bonne qualité, la dynamique collaborative de l'ensemble permettra de produire un tout meilleur que l'ensemble de ses parties (notamment par des corrections en masse de nombreux utilisateurs).

Le succès de ces initiatives ainsi que l'apparition de travaux collaboratifs exploitant la collaboration participative massive (plateformes de production participative telles que Amazon Mechanical Turk), on menées à des évolutions positives dans la recherche en TAL [15] (SABOU, BONTCHEVA et SCHARL, 2012).

D'une part, les projets de la Wikimedia Foundation ont pris une place prépondérante comme ressources pour la recherche, mais nous avons également pu assister à une explosion dans les approches à base de jeux qui permettent d'exploiter des internautes non experts (LAFOURCADE, JOUBERT et BRUN, 2015).

1.3.1.4 Wikipédia et DBPedia

Wikipedia est une encyclopédie collaborative qui compte des éditions dans plus de 250 langues et avec plus de 35,9 millions d'articles au total [16] et dont les éditions les plus développées contiennent individuellement plusieurs millions d'entrées [17] :

- anglais – 4.957.985 articles ;
- suédois – 1.993.354 ;
- allemand – 1.849.647 ;
- néerlandais – 1.833.841 ;
- français – 1.657.535,
- etc.

De plus, les pages équivalentes qui existent dans plusieures langues sont reliées entre elles et il est possible de passer de l'une à l'autre aisément.

15. Il convient de noter que certains usages dérégulés de production participative posent des problèmes éthiques desquels il convient d'être informé afin de les éviter (FORT et al., 2014).

16. Statistiques datant d'août 2015 – https://stats.wikimedia.org/EN/TablesWikipediaZZ.htm

17. Statistiques au 31 août 2015 – https://stats.wikimedia.org/EN/Sitemap.htm

FIGURE 1.8 – Exemple d'une page Wikipédia et de sa structure.

Le problème majeur qui se pose avec Wikipedia et avec les projets liés, est qu'elles se présentent sous un format Wiki. Wikipédia fixe des recommandations générales sur la structure souhaitée des pages. Cependant, rien n'est imposé et on ne peut pas s'assurer que l'information soit correcte ou consistante du point de vue de la représentation.

La Figure 1.8 montre l'exemple d'une page Wikipédia et illustre sa structure : chaque page a un titre, un résumé court du sujet de la page, une boite d'information permettant de mettre en avant des données caractéristiques en fonction du domaine (par exemple, population, superficie, pour une ville), un sommaire avec des liens de navigation généré automatiquement, le contenu, structuré à l'aide d'une hiérarchie d'en-têtes (niveaux 1, 2,3 et plus), une boite de liens vers les pages traitant du même sujet dans d'autres langues. Toutes ces informations sont contribuées par les utilisateurs, qui doivent directement les saisir en format code wiki source.

Certains éléments ont un format prédéfini et sont générés par des macros (par exemple la boite d'information) alors que le reste de la structure est complètement défini par les utilisateurs et sa conformité aux recommandations relève de la bonne volonté et d'une modération suffisamment proactive.

Quand on veut utiliser les données de Wikipedia dans une application informatique ou de TAL, il faut réaliser au préalable une opération d'extraction dans un format structuré mais qui permet aussi de supprimer au moins une partie des données invalides. Une telle opération est de par sa nature non-interopérable.

Ce problème a motivé l'apparition de projets tel que DBPédia (LEHMANN, ISELE, JAKOB, JENTZSCH, KONTOKOSTAS, MENDES, HELLMANN, MORSEY, KLEEF et AUER, 2014) qui vise à extraire l'information d'une manière systématique et de rendre le résultat de cette extraction disponible dans un format accessible. DBPédia en particulier a été extrait sous la forme d'une ontologie en OWL qui est librement et publiquement accessible. Du fait de cette ouverture, le projet DBPédia a connu beaucoup de succès.

Mis à part DBPedia, Wikipédia est également très utilisé comme :

— source de texte parallèle et comparable ;
— moyen d'obtenir des taxinomies de catégories (FLATI, VANNELLA et al., 2014 ; MELO et WEIKUM, 2010 ; WEI et al., 2013) ;
— comme source pour la génération d'autres ressources (voir par exemple la Section 1.3.2.1 pour BabelNet ci-après) ; l'enrichissement de ressources existantes (GUREVYCH, ECKLE-KOHLER, HARTMANN, MATUSCHEK, Christian M MEYER et al., 2012) (voir la Section 1.3.2.2 pour Uby ci-après), pour l'extraction de terminologie bilingue ;
— ainsi que pour de nombreuses autres applications que nous n'énumérerons pas exhaustivement.

Il existe par ailleurs des initiatives telle que ***J**ava-based **W**iktpedia **L**ibrary* (JWPL) qui propose un accès programmatique direct aux données Wikipédia à partir de Java (ZESCH, MÜLLER et GUREVYCH, 2008a). Alors que DBPédia est construit entièrement sous forme d'ontologie et ne garde que les informations pertinentes dans ce contexte, JWPL, même si elle ne fonctionne qu'avec Java, offre un accès direct à toutes les données.

1.3.1.5 Wikitionary et DbNary

À l'instar de Wikipédia, Wiktionary est un projet de dictionnaire dans de nombreuses langues fondé sur les mêmes principes collaboratifs. Initialement, le projet avait pour vocation de servir de dictionnaire pour les utilisateurs de Wikipédia afin de pouvoir définir des termes inconnus et de pouvoir obtenir des traductions des termes.

Wiktionary est fondé sur un modèle collaboratif où les pages sont définies en langage Wiki. Il y a, pour les éditions de langue les plus développées des recommandations générales sur la syntaxe à utiliser pour représenter des types d'éléments communs (sens, définitions, traductions), cependant rien de force les utilisateurs à les respecter. Par ailleurs les recommandations sont différentes pour chaque édition.

Ainsi, si l'on veut utiliser les données de Wiktionary, il faut procéder à une extraction du contenu qui permet de représenter les données sous une forme systématique tout en gérant les nombreux cas d'exception spécifiques à chaque édition.

Il existe quelques librairies en Java qui visent à l'extraction des données de Wiktionary, mais elle se focalisent souvent sur les éditions principales de Wiktionary (l'anglais, le français, l'allemand, le russe) qui sont bien modérées et qui ont une structure relativement propre avec peu d'exceptions :

— Le projet ***J**ava-based **W**iktionary **L**ibrary* (JWKTL) développé à l'université de Darmstadt en Allemagne, à l'instar de JWPL, propose un accès programmatique aux données contenues dans Wiktionary (entrées, sens, définitions de sens, exemples, traductions, relations sémantiques) en anglais, allemand et russe (ZESCH, MÜLLER et GUREVYCH, 2008a).
— Pour l'édition russe de Wiktionary, un autre librairie, Wikokit (analyseur syntaxique pour les pages Wiktionary) est utilisée et permet d'extraire des informations semi-structurées pour ensuite les stocker dans une base de données (A. KRIZHANOVSKY et SMIRNOV, 2013).
— Il existe d'autres initiatives similaires, telles que GLAWI (SAJOUS et HATHOUT, 2015), qui extrait l'édition française de Wiktionary sous un format XML, et qui en plus des éléments extraits par JWKTL extrait aussi les

formes fléchies, l'étymologie la prononciation et des relations morphologiques.

En termes d'interopérabilité, JWKTL offre un accès au travers d'une API Java, ce qui limite son interopérabilité à des programmes écrit en Java utilisant cette librairie (le format de base de données utilisé est spécifique à la librairie). Quant à GLAWI, le format XML permet de réutiliser les données du moment que l'on peut écrire un adaptateur qui comprenne le format XML spécifique de la ressource.

La dernière initiative d'extraction de Wiktionary d'envergure, DBNary (Sérasset, 2015), suit la même philosophie que le projet DBPedia et cherche à représenter les données contenues dans Wiktionary dans un format d'ontologie, et plus particulièrement au format lemon que nous avons décrit dans la Section 1.1.3.

DBNary est ainsi interopérable avec les nombreuses autres ressources et outils qui gèrent ce format et permet aussi de réaliser des alignements avec des ontologies et d'autres ressources liées ouvertes. DBNary extrait les mêmes informations que JWKTL avec en plus les dérivations morphologiques des verbes (quand elles sont disponibles).

À l'heure actuelle 21 langues sont extraites (bulgare, hollandais, anglais, finnois, français, allemand, indonésien, italien, japonais, latin, lituanien, malagasy, grec moderne, norvégien, polonais, portugais, russe, serbo-croate, espagnol, suédois, turc).

DBNary utilise cependant des extensions pour les éléments non supportés par lemon (les entrées de haut niveau ou vocables, les relations de traduction, ainsi que certaines informations de catégorie grammaticale que ne peuvent être standardisés).

Bon nombre des extensions deviendront redondantes lors du passage à Ontolex. Du fait que DBNary contient de nombreuses éditions de langues, il est plus aisé d'obtenir des informations supplémentaires sur les traductions vers chaque langue, à condition de pouvoir lier les traductions (chaînes de caractère) aux entrées cibles dans les éditions de langue correspondantes.

La Figure 1.9 montre l'exemple du vocable `cat` en anglais ainsi que les entrées correspondantes mais aussi les sens associés à l'une des entrées lexicales de `cat`. Elle présente également l'exemple de la première entrée lexicale pour `cat` et présente une liste non-exhaustive des traductions dans plusieures langues.

1.3.1.6 Kyoto

Le Projet Kyoto (Piek Vossen et al., 2010), avait pour objectif de créer une plateforme collaborative semi-automatique pour l'extraction de connaissances.

Kyoto est une architecture axée autour d'une ontologie centrale permettant d'encoder les connaissances extraites, ainsi que PWN et d'autres EuroWordNets, qui sont liés à l'ontologie. Les *WordNet*s dans différentes langues sont représentés au format LMF-Wordnet, qui a été inventé pour l'occasion.

Le processus d'extraction est géré par un système multi-agents : des tybots extraient d'abord les termes pertinents qui sont ajoutés à l'ontologie de manière hiérarchique, puis des kybots vont détecter des motifs correspondant à des relations conceptuelles (ontologiques et morpho-syntaxiques) dans du texte à partir des informations de l'ontologie.

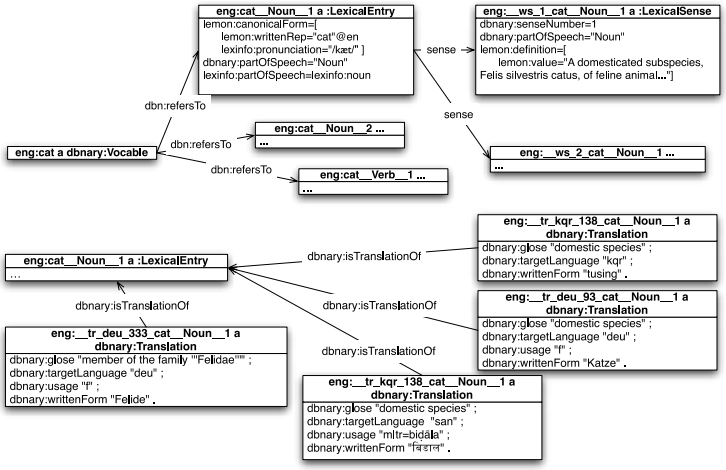


FIGURE 1.9 – Un exemple d'une entrée de haut niveau et de toutes les entités liées ainsi que d'une entrée lexicale et de ses traductions, tel que présenté par SÉRASSET (2015).

Le format de sortie est le ***K**YOTO **A**nnotation **F**ramework* (KAF), qui est une extension du format standard LAF et qui permet donc une interopérabilité au niveau de la représentation avec d'autres systèmes d'annotation.

La particularité de Kyoto qui le rend intéressant est qu'il est aussi prévu que des humains viennent apporter des corrections à l'ontologie et aux données lexicales récoltées, mais aussi puissent définir eux même les sens des termes extraits, ce qui met en jeu des problématiques de dynamicité des ressources, où il faut faire en sorte que les changements apportés par les utilisateurs ne soient pas écrasés par les processus automatiques des kybots. Ces mêmes problématiques se posent pour la construction de ressources lexico-sémantiques à base d'entrées et il convient de les garder à l'esprit.

1.3.1.7 Papillon

Le projet Papillon (BOITET, MANGEOT et SÉRASSET, 2002) est un projet visant a créer une ressource lexico-sémantique multilingue de manière collaborative. Papillon se compose de volumes monolingues suivant le format du dictionnaire DiCo [18] (STEINLIN et al., 2004) et étant composés de sens de mots, appelés lexies, en concordance avec la théorie sens-texte.

Papillon est fondée sur une architecture pivot interlingue artificiel (Section 1.2.3), et plus particulièrement sur le formalisme des acceptions interlingues proposé par SÉRASSET (1994). Ce dernier garantit, de par sa structure, que des phénomènes contrastifs artificiels ne se produisent pas.

L'idée de ce formalisme est de ne pas seulement créer des équivalences directes entre les sens, mais aussi de créer une hiérarchie d'équivalences dans les cas où des lexicalisations divergentes dans certaines langues définissent des distinctions sémantiques (voir l'exemple de `riz` présenté dans la Section 1.2.3.1).

On peut reformuler l'exemple de la Figure 1.6 afin d'illustrer les mécanismes de hiérarchie d'équivalences des acceptions interlingues dans la Figure 1.10.

18. Présenté dans la Section 1.3.1.1.

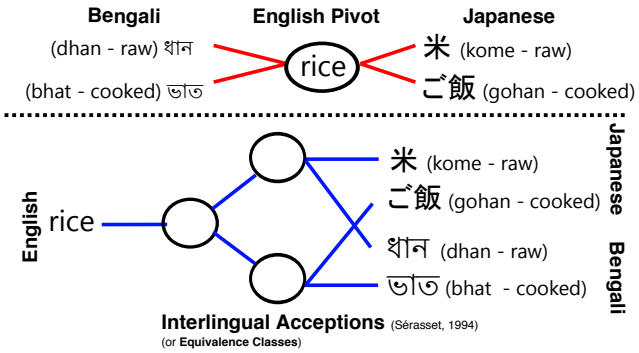


FIGURE 1.10 – Remédier à un phénomène contrastif artificiel avec les acceptions interlingues.

Le projet était basé sur une construction experte collaborative, ce qui avait le potentiel de produire une ressource très intéressante. Cependant, il s'est avéré difficile, de trouver des contributeurs experts qualifiés (c'est-à-dire qui maitrisent trois ou plus des langues [19]). Par ailleurs, faire participer les annotateurs de manière bénévole et volontaire sur le long terme est une tâche très difficile.

1.3.1.8 Jeux avec un but et Jeuxdemots

Les jeux avec un but (en anglais ***G**ames **W**ith **A** **P**urpose* (GWAP)) (LAFOURCADE, JOUBERT et BRUN, 2015) sont des jeux dans lesquels, jouer est un moyen d'arriver à un but qui a une finalité pratique et utile en dehors du jeu en lui même.

Les GWAP sont également utilisés en recherche, où ils constituent un moyen d'exploiter les capacités cognitives des humains en masse au travers d'un jeu afin de résoudre un problème de recherche difficile.

Ce type de jeu est aussi utilisé en TAL, notamment pour produire des ressources langagières.

La majeure partie des GWAP se présentent comme des sites internet proposant un jeu plutôt textuel tournant autour des mots, mais il existe aussi quelques jeux «locaux» [20] complets.

Les jeux de la première catégorie sont souvent des jeux d'annotation linguistique qui s'éloignent peu de la tâche d'annotation en elle même (JURGENS et Roberto NAVIGLI, 2014). JURGENS et Roberto NAVIGLI (2014) quant à eux proposent une série de deux jeux complets avec des graphismes classiques de jeu vidéo : un pour la désambiguïsation lexicale (voir **Définition** 1.15) et un pour l'alignement de « sens » *WordNet* avec des images. Ils rapportent une réduction des coûts de production de 73% par rapport à des approches de participation collaborative ainsi qu'un gain de 16.3% de précision sur l'état de l'art des jeux sérieux pour la désambiguïsation lexicale.

Une autre plateforme de jeux sérieux pour l'acquisition lexicale se démarque : il s'agit de Jeuxdemots (LAFOURCADE et JOUBERT, 2013). Jeuxdemots avait au départ avant tout l'objectif de construire un réseau lexico-sémantique représentant le lexique, ces sens et divers types de relations lexico-sémantiques.

19. L'un des inconvénients des architectures par pivot interlingue artificiel, telle que décrite dans la Section 1.2.3.1

20. Des jeux qui ne sont pas des application web, mais des programmes classiques qui tournent localement sur les machines des joueurs.

Définition 1.15 Désambiguïsation lexicale
La désambiguïsation lexicale consiste à assigner aux mots ambigus d'un texte les sens (issus d'un inventaire de sens) qui correspondent au contexte d'usage du mot. Il s'agit d'une tâche standard et fondamentale en TAL.

Jeuxdemots était d'abord un jeu d'association d'idées (amorçage sémantique) : il vous présente un terme ou une expression, et vous devez lister les termes qui s'y rapportent d'après vous.

Vos réponses sont évaluées en fonction des associations les plus fréquentes faites par les autres utilisateurs. Jeuxdemots existe dans plusieures langues, mais seulement la communauté française est vraiment active et contribue. C'est la seule pour laquelle il y a une animation permanente.

Un avantage de jeuxdemots est que la plateforme propose de nombreux jeux différents qui capturent différents types de données, ou qui les présentent simplement sous un format plus ludique pour certains publics [21].

Les données de Jeuxdemots sont librement et gratuitement téléchargeables et consultables en ligne, cependant elles se présentent comme des fichiers avec des champs séparés par des tabulations, ce qui signifie qu'il faut les convertir où écrire un adaptateur spécifique avant de les utiliser.

1.3.2 Ressources construites automatiquement

1.3.2.1 Babelnet

BabelNet (Roberto NAVIGLI et PONZETTO, 2012a) est une ressource lexicale multilingue se basant sur PWN et intégrant les informations contenues dans Wikipédia pour la rendre multilingue. BabelNet utilise un système de traduction automatique pour compléter les définitions dans les cas où elles n'ont pas pu être extraites de la page Wikipédia.

BabelNet complète les *synset*s de PWN avec les mots «synonymes» des autres langues au travers des liens de pages multilingues contenues dans Wikipedia afin d'obtenir des *BabelSynsets*.

Au fil du temps, BabelNet a connu trois versions majeures. La première version publique contenait 5 langues et a servi d'inventaire de sens pour une tâche de désambiguïsation lexicale multilingue dans SemEval 2013 (tâche 12) (Roberto NAVIGLI, JURGENS et VANNELLA, 2013).

La version 2.0 sortie en 2014 passe à 51 langues supportées et a été utilisée comme base pour la tâche de similarité multi-niveaux lors de la campagne d'évaluation SemEval 2014 (JURGENS, M. T. PILEHVAR et Roberto NAVIGLI, 2014).

Jusqu'à présent BabelNet avait son propre format de données avec une interface de programmation dédiée, de laquelle elle était dépendante. Le passage à la version 3.0 en 2015 fait passer le nombre de langues à 272 et BabelNet devient également disponible en lemon (EHRMANN et al., 2014) (voir Section 1.1.3).

Dans cette dernière version, il y a 13 801 844 *BabelSynsets*, soit :

— 119 036 997 sens ;
— 6 066 256 «concepts»
— 7 735 588 entités nommées ;

21. http://imaginat.name/JDM/Page_Liens_JDMv4.html

— 380 239 084 relations lexico-sémantiques [22]. Au fur et à mesure des versions, BabelNet a été enrichi par OmegaWiki [23], Wiktionary anglais, allemand et russe, Wikidata [24], Open Multilingual Wordnet [25].

La version 3.0 a servi de base à la tâche 13 de désambiguïsation lexicale et de détection d'entités nommées à Semeval 2015 (Moro et Roberto Navigli, 2015). BabelNet a également été amélioré par divers processus algorithmiques, par exemple l'inférence de nouvelles relations taxinomiques (Flati, Vannella et al., 2014).

Les créateurs de BabelNet sont en train de construire tout un écosystème d'outils et d'interfaces de programmation en ligne. Il s'agit notamment de Babelfy, la combinaison de BabelNet à un système de désambiguïsation lexicale multilingue à la pointe de l'état de l'art, qui permet à n'importe qui de désambiguïser un texte.

C'est aussi un moyen d'aligner d'autres ressources avec Babelnet au niveau des *synset*s en générant une description textuelle d'un sens (par exemple un exemple d'usage) qui est annotée avec des *BabelSynsets*, ce qui permettra ensuite d'établir un alignement avec le BabelSynset pour lequel les mots correspondent à un sens dans l'exemple d'usage.

Cette facilité d'usage vient cependant au prix d'une précision et rappel qui ne sont pas parfaits et limités par la performance du système de désambiguïsation. Évaluer les erreurs sans l'emploi d'un corpus de référence étant difficile, il n'y a pas de garantie que les résultats seront corrects sur tous les types de textes, en particulier dans des domaines spécifiques qui ne sont pas nécessairement bien couverts. Ce n'est cependant pas un problème limité à Bebelfy, mais à tout système de désambiguïsation lexicale appliqué de manière ouverte sur du texte brute.

En terme d'architecture multilingue, BabelNet utilise un pivot naturel, qui sont les «concepts» anglais contenus dans WordNet. Par conséquent, BabelNet souffre du problème de phénomènes contrastifs artificiels.

La philosophie derrière le développement de BabelNet est de privilégier une grande quantité de données et une grande couverture, avec un processus d'amélioration et de correction des erreurs incrémental au fur et à mesure que des solutions algorithmiques se développent dans l'état de l'art.

1.3.2.2 Uby

Uby est une ressource développée à l'université technique de Darmstadt en Allemagne par Gurevych, Eckle-Kohler, Hartmann, Matuschek, Christian M Meyer et al. (2012) et dont le nom signifie *A Large-Scale Unified Lexical-Semantic Resource* c'est-à-dire, une ressource lexico-sémantique unifiée à grande échelle.

22. Statistiques au 22 Octobre 2015, voir http://babelnet.org/stats pour des statistiques complètes et à jour.

23. Un dictionnaire multilingue basé sur Wiktionary mais imposant des règles de structuration strictes et homogènes. http://www.omegawiki.org.

24. Base de connaissances de la fondation Wikimédia. https://meta.wikimedia.org/wiki/Wikidata.

25. http://babelnet.org/about

Uby utilise le standard ISO LMF et plus particulièrement Uby-LMF, une extension qui offre un format de sérialisation. Uby combine 12 ressources lexico-sémantiques en anglais et en allemand :

— En anglais : PWN WordNet, Wiktionary anglais, OntoWiktionary [26] anglais, Wikipédia, FrameNet, VerbNet [27] ;
— et en allemand : Wikipédia, Wiktionary, OntoWiktionary, GermaNet [28], IMSLex-Subcat [29].

UBY et utilise une représentation pivot interlingue artificielle. Comme il n'y a que deux langues dans la ressource, la représentation pivot est équivalente à une architecture par transfert. La raison pour laquelle Uby se cantonne à deux langues est pour éviter les difficultés de construction du pivot interlingue dans le cas où il y a plus de deux langues et pour éviter de recourir à un pivot naturel qui mènerait à des phénomènes contrastifs artificiel. Uby est une ressource avec des alignements de qualité et une grande couverture sur les deux langues qu'il contient.

1.3.2.3 OpenMultilingualWordNet

Le projet Open Multilingual Wordnet (le WordNet multilingue ouvert) (BOND et FOSTER, 2013) a débuté sur la fondation de l'initiative *Global Wordnet Association* qui recensait l'ensemble des WordNets existant dans le monde [30].

Tous les WordNets ouverts disponibles ont été rassemblés et mis à disposition et ont ensuite été tous alignés à PWN de manière automatique. L'architecture d'alignement employée est celle d'EuroWordNet (Piek. VOSSEN, 1998), l'ILI où PWN est utilisé comme pivot interlingue naturel, et, dans les cas où il n'y a pas de concept correspondant dans PWN, le concept étranger est importé et utilisé comme pivot.

Malgré ce mécanisme, les problèmes de phénomènes contrastifs artificiels se manifestent également. OpenMultilingual WordNet, dans sa version étendue actuelle, comprends plus de 2 millions de sens pour 100 000 concepts et relie 140 millions de mots dans des centaines de langues.

Les WordNets individuels suivent le même format que PWN. OpenMultilingualWordNet fournit en réalité que les alignements entre les WordNets individuels sous la forme de fichiers csv avec des tabulations comme séparateur. Il est possible de convertir l'ensemble de la ressource vers LMF-Wordnet ou vers lemon, cependant ce travail est encore à venir.

1.3.2.4 Graphes de traduction

Un graphe de traduction est composé de nœuds qui correspondent à des lemmes dans différentes langues et qui sont reliés par des relations de traduction.

Parmi les ressources présentées auparavant, l'ensemble des Wiktionary extraits dans DBNary forment un graphe de traduction. Il en va de même pour BabelNet, car on est en mesure, pour un mot donné de trouver ces traductions dans d'autres langues.

26. Ontologie construite à partir de Wiktionary (Christian M. MEYER et GUREVYCH, 2012a).
27. Le plus grand lexique verbal en langue anglaise (KIPPER et al., 2006).
28. Le WordNet de l'allemand (HAMP et FELDWEG, 1997).
29. Un lexique morphologique de l'allemand (FITSCHEN, 2004).
30. http://globalwordnet.org/wordnets-in-the-world/.

Dans un tel graphe il peut potentiellement y avoir une distinction de sens, cependant elle se fera souvent en fonction des traductions dans les autres langues (comme c'est par ailleurs le cas pour les distinctions de sens dans les dictionnaires bilingues (CAMPENHOUDT, 2000)).

Un graphe de traduction est une ressource lexicale multilingue à base d'entrées où les entrées sont les nœuds dans le graphe (les mots). Les graphes de traduction imposent moins de contraintes que des ressources lexico-sémantiques complètes (par exemple des dictionnaires) et peuvent être construits à des échelles bien plus massives.

Nous pouvons citer le projet PanLex (KAMHOLZ, POOL et COLOWICK, 2014) qui vise à fournir des traductions de «concepts lexicaux» (lexies) dans un nombre maximal de langues. PanLex contenait en mars 2014[31] 20 millions de lexèmes et expressions, dans plus de 9000 langues pour un total de 1,1 milliards de paires de traductions entre lexèmes.

PanLex est disponible en RDF au format lemon[32]. PanLex est construit automatiquement (agrégation de corpus, de dictionnaires, de ressources lexicales, etc.) puis des correcteurs humains inspectent les données pour corriger d'éventuelles erreurs (bien entendu de manière non exhaustive du fait de la taille colossale des données.)

N.B.

Les glissements sémantiques corrélés arrivent lorsque des chemins de traduction dans le graphe partagent plusieurs sens avec d'autres nœuds du graphe. Il s'agit d'une forme de phénomènes contrastifs artificiels lors de l'emploi de mécanismes d'inférence transitifs, ce qui cause des erreurs.

Une autre initiative remarquable par MAUSAM et al. (2009) s'intitule PanDictionary, un graphe de traduction contenant plus de 10 millions de mots dans plus de 1000 langues. PanDictionary est construit par inférence probabiliste et les liens de traduction sont classés par degré de confiance en l'existence réelle du lien de traduction.

Ainsi, il y a quatre fois plus de liens de traductions que dans le Wiktionary anglais avec une précision de 90% et plus de 200 millions de paires de traductions dans 200 000 langues avec une précision de 80%. Au lieu de recourir à des correcteurs humains, MAUSAM et al. (2009) caractérisent les erreurs principales (erreurs dans les dictionnaires source et glissements sémantiques corrélés causés par la polysémie) et conçoivent un algorithme pour les détecter et les exclure du graphe.

1.3.2.5 Induction des sens de mots et sémantique distributionnelle

Une autre manière de créer des ressources lexico-sémantiques est d'exploiter de grandes quantités de corpus afin d'automatiquement inférer les distinctions de sens en fonction des contextes d'usage des mots et de leur voisinage lexical. D'une manière générale, c'est ce qu'on appelle les méthodes de sémantique distributionnelle (se basant sur l'hypothèse distributionnelle), et qui utilisent une représentation vectorielle qui modélise les relations entre les mots. La plupart des méthodes distributionnelles ne font pas de distinctions de sens en fonction des contextes d'usage.

Ainsi, l'idée est de compter les termes qui co-occurrent dans un contexte donné (phrase, paragraphe, document) issu d'un corpus. On construit ainsi une matrice terme-terme qui permet de mesurer les cooccurrences.

31. Dernières statistiques disponibles, comme présentés dans KAMHOLZ, POOL et COLOWICK (2014).

32. http://ld.panlex.org/rdf.html

On applique typiquement un algorithme de réduction de dimensionnalité car les vecteurs sont très creux et de dimensionnalité très grande, en projetant l'un des axes de termes de la matrice dans un espace à dimensionnalité bien plus petite (par exemple 300) en ne conservant que les informations sémantiques les plus pertinentes. Nous pouvons citer l'exemple de l'approche LSA qui applique une décomposition en valeurs singulières de la matrice et qui ne conserve que les composants contribuant le plus à la variance.

Un sous ensemble de ces méthodes prennent en compte les distinctions de sens au travers de la construction d'une matrice terme-contexte qui met en relation tous les termes d'un corpus avec tous les contextes d'usage (par exemple phrases ou paragraphes).

La construction d'une ressource lexico-sémantique (ou du moins un inventaire de sens) requiert de regrouper pour chaque terme tous les contextes où apparait ce terme qui sont suffisamment similaires les uns aux autres en utilisant une technique de regroupement automatique (*clustering*).

C'est ce que l'on appelle l'induction automatique de sens ou en anglais ***W**ord **S**ense **I**nduction* (WSI), qui a fait l'objet de plusieurs tâches d'évaluation lors des campagnes SemEval, notamment en 2007 (Agirre et Soroa, 2007), 2010 (Manandhar et al., 2010) et 2013 (Roberto Navigli et Vannella, 2013).

Les approches vectorielles existent depuis de nombreuses années et ont longtemps offert des résultats très intéressants pour mesurer les relations sémantiques entre mots et sens. Cependant elles avaient le désavantage d'une construction longue et complexe. On peut citer les vecteurs conceptuels et sémantiques (Lafourcade et Stanford, 1999 ; Schwab, Tze et Lafourcade, 2007) l'une des premières adaptations des modèles vectoriels en recherche de document (Salton, 1983) pour le TAL.

La montée en puissance des processeurs (centraux et graphiques) a rendu l'utilisation de techniques de calcul sur des réseaux de neurones profonds possible, qui offrent un calcul de la projection en dimensionnalité réduite bien plus rapide.

Cela permit la création de modèles de langue vectoriels basées sur la sémantique distributionnelle qui permettent des améliorations pour beaucoup de tâches existantes en TAL. Nous pouvons citer le plus utilisé, Word2Vec (Mikolov et al., 2013), produit par Google Research et qui fournit des modèles pré-entrainés sur des corpus comprenant des milliards de mots (corpus Google News).

Conclusions

Dans ce chapitre, nous avons présenté les éléments de base portant sur l'interopérabilité des ressources lexico-sémantiques, et plus particulièrement les trois niveaux d'interopérabilité : représentationnelle, sémantique et dynamique avant de nous focaliser particulièrement sur les deux premiers. Nous avons ainsi présenté les différents standards et systèmes de représentations existants qui garantissent au moins partiellement une interopérabilité et avons ensuite présenté les ressources lexico-sémantiques actuelles les plus utilisées et intéressantes ainsi que la manière dont elles se placent par rapport aux critères d'interopérabilité à la fois représentationnelle mais aussi sémantique.

Comme nous l'avons vu dans la Section 1.1.3, le format lemon/Ontolex est un pas en avant pour l'interopérabilité représentationnelle des ressources et offre toute une infrastructure de stockage et d'accès aux données performant et éprouvé provenant des technologies du Web sémantique. Les principales ressources du domaine sont également disponibles. Ainsi l'utilisation de ce format de représentation est un choix naturel pour s'assurer que les productions de ce travail soient réutilisables et pérennes.

Quant à DBNary, c'est une ressource avec beaucoup de traductions et de relations mais qui a une constance et une qualité très variable selon les langues. De plus les relations et traductions portent sur des entrées lexicales ou des vocables et non sur des sens, ce qui est un pré-requis pour aider à l'alignement des sens au niveau multilingue.

DBNary recense une grande partie des cas de figure problématiques pour l'interopérabilité ainsi que beaucoup de langues différentes. Les algorithmes développés sur DBNary devront faire preuve d'une robustesse et d'une capacité de généralisation supérieure. Nous éviterons ainsi le risque de produire des algorithmes qui ne fonctionnent qu'avec un petit ensemble de langues et qui ne sont pas valides ou testés pour le reste.

Chapitre 2

Alignement automatique de sens & interopérabilité sémantique

Sommaire

Introduction

L'interopérabilité sémantique de ressources lexico-sémantiques passe avant tout par l'alignement des ressources entre elles au niveau des sens de mots. Dans le cas le plus simple, l'alignement de deux ressources dans la même langue, des difficultés se présentent déjà.

En effet, comme nous l'avons montré dans l'exemple de l'alignement de WordNet et de ODE dans la Section 1.2 et dans la Figure 1.3, la granularité des sens n'est pas forcément la même, et certains sens présents dans un dictionnaire ne sont pas nécessairement présents dans l'autre.

Quand il s'agit de ressources dans différentes langues, des problèmes supplémentaires se posent : la comparabilité des informations sémantiques, des lexicalisations et des conceptualisations divergentes (il n'y a pas de correspondance 1 à 1) pour l'alignement de plus de deux ressources.

C'est un facteur décisif pour les distinctions faites entre les architectures d'interopérabilité sémantique multilingue. Comme nous l'avons décrit auparavant, le mode de construction manuel pose un problème de couverture et de taille du

fait des coûts associés, ce qui signifie qu'il est avantageux de prendre l'approche d'une construction automatique. Ainsi plusieurs ressources lexico-sémantiques multilingues construites automatiquement existent et peuvent atteindre des tailles sans commune mesure avec les ressources à construction manuelle (Section 1.3).

Le processus est schématisé dans la Figure 2.1 et se déroule en deux phases. Nous avons au début un ensemble de ressources partiellement alignées ou non alignées. Il faut d'abord compléter les alignements entre toutes les ressources deux à deux, puis dans une deuxième étape créer une représentation interlingue pour les lier toutes. Ce chapitre se focalise sur la première étape, c'est-à-dire l'alignement automatique de sens.

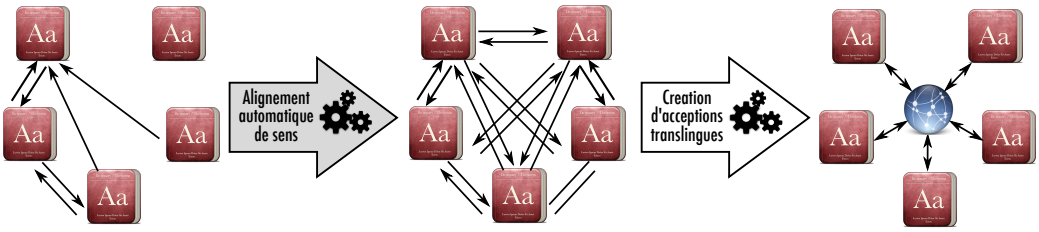


FIGURE 2.1 – Processus général de création des ressources lexico-sémantiques multilingues – alignement automatique de sens.

Les algorithmes d'AAS ont pour but de relier les sens de mots équivalents provenant de plusieurs ressources lexicales (MATUSCHEK, 2015, Chapitre 2). Ainsi, pour des ressources n'ayant pas exactement la même granularité ou le même découpage sémantique, des algorithmes d'AAS permettront de les réconcilier et de les rendre sémantiquement interopérables à condition de pouvoir échapper aux phénomènes contrastifs artificiels par le choix de la bonne représentation.

L'objectif de la thèse étant la construction automatique d'acceptions interlingues, il convient de passer en revue les méthodes d'alignement automatique de sens existantes puis de s'intéresser à la représentation qui permettra d'éviter les phénomènes contrastifs artificiels. Ainsi nous nous intéresserons aux acceptions interlingues d'un point de vue formel et présentons les travaux antérieurs portant sur leur construction automatique et verrons quelles sont les problèmes supplémentaires qui se posent par rapport aux approches classiques d'AAS.

2.1 Algorithmes pour l'alignement automatique de sens

Un algorithme d'AAS doit déterminer une équivalence entre des sens de mots. Il faudra pouvoir déterminer la proximité ou l'éloignement entre l'information sémantique dénotée par les sens puis de pouvoir décider si la proximité est suffisante afin de considérer les sens équivalents. Une manière de déterminer cette proximité est d'utiliser des mesures de similarité sémantique qui permettent de déterminer la proximité sémantique en comparant des informations qui sont supposées être représentatives du sens pour un humain (TCHECHMEDJIEV, 2012), par exemple les définitions et les exemples (gloses) présents dans un dictionnaire. Lors de l'alignement de deux dictionnaires entre eux les seules informations disponibles sont les définitions et les exemples, ce qui impose le choix de la mesure de similarité.

Quand on manipule des RLS plus complexes représentées sous forme de graphe (par exemple un dictionnaire augmenté d'un thésaurus) il est alors possible d'ex-

ploiter les liens entre les nœuds ainsi que la topologie des graphes des ressources pour déterminer l'équivalence des sens.

Nous allons tout d'abord présenter des tâches similaires à AAS dans d'autres domaines ainsi qu'en TAL, puis nous présenterons les approches pour l'AAS en général suivi d'une étude des différentes approches utilisées pour permettre d'aligner des ressources dans différentes langues, avec une insistance particulière sur les manquements dans le contexte de l'alignement simultané de plus de deux ressources.

2.1.1 Tâches similaires à l'alignement automatique de sens

Des techniques similaires à l'AAS sont utilisées dans d'autres domaines ainsi que pour d'autre tâches en TAL, comme répertorié par MATUSCHEK (2015, Chapitre 2). Nous allons reprendre les tâches décrites par ce dernier, mais en proposant notre propre analyse des similarités et différences avec l'AAS.

2.1.1.1 Autres domaines que le TAL

ALIGNEMENT D'ONTOLOGIES. Le but de l'alignement d'ontologies est de mettre en relation deux ontologies avec des schémas de types différents et d'identifier si il y a des instances de classes ou des relations qui portent la même (ou du moins très proche) sémantique formelle.

L'alignement d'ontologie exploite typiquement soit la structure de l'ontologie (MAEDCHE et STAAB, 2002), soit les termes présents dans l'ontologie (approche terminologique) autour des éléments à aligner (William W. COHEN, RAVIKUMAR et Stephen E. FIENBERG, 2003), soit des approches hybrides qui les combinent (PIRRÓ et EUZENAT, 2010).

Ces méthodes ont certains points communs avec l'AAS, en particulier du point de vue des mesures qui permettent de déterminer la proximité sémantique des entités. En effet, certaines des nombreuses mesures utilisées en TAL (TCHECHMEDJIEV, 2012) sont également utilisées sur des ontologies (PIRRÓ et EUZENAT, 2010).

Cependant, alors que dans une ontologie toute interprétation est définie par une sémantique formelle qui permet un raisonnement et une inférence logique, ce n'est pas le cas pour les RLS dans le cas général. En effet, la sémantique en langue naturelle n'est pas vraiment consistante, ni transitive (par exemple, deux synonymes d'un même mot ne sont pas synonymes entre eux).

Par ailleurs, il n'y a aucune polysémie dans une ontologie. Même si les données d'une RLS sont stockées comme une ontologie, les raisonnements formels ne mèneront pas à des conclusions ayant une validité linguistique.

Le seul cas où il peut y avoir une convergence des deux approches est dans le cas de RLS purement terminologiques qui ne contiendraient pas ou peu de polysémie, ce qui mènerait aussi à des relations de synonymie transitives.

Des problèmes communs à l'AAS existent cependant : la granularité des représentations sémantiques peut varier entre ontologies ; des ontologies dans des langues différentes (LESNIKOVA, DAVID et EUZENAT, 2015) ont des représentations textuelles (noms des classes, étiquettes, etc.) qui ne sont pas comparables (dans des langues ou des systèmes d'écriture différents).

Le problème de granularité porte bien entendu ici sur des logiques formelles strictement différentes de la sémantique lexicographique décrite dans le Chapitre 1.

ALIGNEMENT DE SCHÉMAS DE BASES DE DONNÉES. L'alignement de schémas de bases de données a des similarités avec l'alignement d'ontologie. En effet, les relations entre les entités sont strictement définies de manière formelle (par exemple les contraintes de clefs étrangères). Il est possible d'exploiter certaines méthodes d'alignement d'ontologies pour l'alignement de bases de données (BERLIN et MOTRO, 2002).

Une base de données représente elle aussi le monde réel tout comme le fait une ontologie, cependant, alors que l'ontologie encode explicitement la sémantique des données stockées, ce n'est pas le cas d'un schéma de bases de données. En effet, la sémantique est implicite et encodée dans le nom des tables et des attributs et une spécification qui nécessaire pour connaitre leur signification exacte (KANG et NAUGHTON, 2003).

De plus les relations dans les bases de données SQL ne sont pas explicitement nommées et rien ne garantit que les noms des tables et des attributs auront une signification ou même qu'elles seront consistantes entre deux bases de données (différents schémas possibles pour une même signification) avec la même sémantique (MATUSCHEK, 2015, Chapitre 2). Cette tâche est ainsi plus éloignée de la tâche de AAS que l'alignement d'ontologies, même si quelques problèmes similaires se posent.

ALIGNEMENT DE GRAPHES. L'alignement de graphes a pour but de trouver une correspondance structurelle (isomorphisme) entre deux graphes, et plus particulièrement de pouvoir déterminer, pour deux nœuds provenant de graphes distincts, si ils ont la même position dans les topologies respectives des deux graphes. Dans le cas général, on sait uniquement si deux nœuds sont liés ou non, ce qui n'est habituellement pas une information suffisante si nous voulions adapter des algorithmes d'alignement de graphes pour des tâches telles que l'AAS.

En effet, pour aligner les sens de mots, il est utile d'ajouter certaines contraintes liées au différents types de nœuds (un sens, un *synset*, un mot, etc) ou d'arcs (relations sémantiques différentes) et utiliser des sources de savoir supplémentaires (heuristiques à base de mesure de distance) afin de d'amoindrir problèmes posés par la différence potentielle dans la structure des ressources lexicales (MATUSCHEK, 2015, Chapitre 2). Il existe cependant des travaux qui proposent des mesures de distance fondées uniquement sur les topologies des ressources lexico-sémantiques en exploitant des techniques de marche aléatoire (GAUME et al., 2014).

2.1.1.2 En TAL

DÉSAMBIGUÏSATION LEXICALE. La désambiguïsation lexicale (***W**ord **S**ense **Dis**ambiguation* (WSD)) est une tâche fondamentale en TAL depuis les débuts du domaine dans les années 50. La désambiguïsation lexicale consiste à déterminer pour un lexème utilisé dans un contexte donne (phrase, texte), le sens particulier dans lequel ce mot est utilisé à partir des sens répertoriés dans un inventaire de sens (par exemple un dictionnaire ou WordNet).

Dans son expression la plus générale, la désambiguïsation lexicale consiste à annoter tous les mots d'un texte d'une manière cohérente (TCHECHMEDJIEV, 2012, All-words WSD). Il existe cependant aussi des tâches de WSD portant sur

une d'échantillon lexical (*lexical sample*) qui visent à annoter un mot cible contenu dans une phrase (contexte d'usage particulier), sans annoter le reste du texte.

Il y a de nombreuses approches de WSD, en allant des approches supervisées apprenant un modèle de classification à partir de données d'entraînement déjà existantes aux approches à bases de connaissances exploitant uniquement les informations contenues dans des ressources lexico-sémantiques et des mesures de similarité sémantique.

En WSD, on assigne à un ou plusieurs mots une étiquette correspondant à un sens de ce mot tel que défini dans une ressource lexico-sémantique en tenant compte du contexte d'usage du mot, alors qu'en AAS nous devons assigner une étiquette correspondant à un sens dans la même ou une autre ressource lexico sémantique en exploitant les informations sémantiques portées par le sens ainsi que son voisinage dans le graphe.

Il est dans ce sens possible de directement transposer des méthodes de WSD pour faire de l'AAS en remplaçant le mot à annoter par un mot ou un sens de mot dans une autre ressource lexico sémantique que celle qui sert d'inventaire de sens. Il faut alors établir la proximité sémantique entre le sens de la ressource source et le ou les sens de la ressource cible afin de trouver en ensemble de sens suffisamment proches pour être considérés comme équivalents du point de vue d'un humain.

Une solution de calculer cette similarité est d'utiliser des approches comparant les descriptions textuelles des sens, comme les définitions ou les exemples d'usage qui sont souvent présents dans les ressources lexico-sémantiques. Nous pouvons citer la similarité de Lesk (1986) et ses extensions, comme celle utilisée l'alignement de relations de traduction aux sens source dans Wiktionary (Tchechmedjiev et al., 2014).

Un autre type d'approche de WSD exploite la structure de la ressource lexico-sémantique (par exemple (R. Navigli et Lapata, 2010 ; Roberto Navigli et Velardi, 2005)) et peut également s'appliquer à l'AAS.

Une différence fondamentale entre la WSD et l'AAS est due au fait que dans un texte, du moins localement, les usages successifs du même mot provenant de la même partie du discours portent souvent le même sens («One sense per discourse»), or ce n'est pas le cas dans une ressource lexico-sémantique. Ce qui se rapproche le plus de cette notions dans les ressources lexico-sémantiques sont des sens reliés par une relation de synonymie qui dénote à peu près le même contexte d'usage(Matuschek, 2015). Par ailleurs, la WSD "All-words" n'a pas vraiment adapté pour l'alignement des ressources lexico-sémantiques du fait que l'on ne travaille que sur un sens source particulier à la fois.

SIMILARITÉ TEXTUELLE. La similarité textuelle consiste à calculer un score de similarité entre des textes et des passages de texte afin de déterminer la proximité sémantique et/ou thématique des textes entre eux. Les approches de similarité textuelles sont essentielles pour la classification de texte ou la recherche d'information (par exemple recherche sémantique floue).

Les approches de similarité textuelles sont très proches de certaines approches de WSD et de AAS, en particulier celles consistant à comparer les descriptions textuelles des sens de mots. Cependant il y a une différence fondamentale : un sens est la lexicalisation en contexte d'un concept. Quand on compare les mots

des définitions de plusieurs sens on sait que les mots vont être représentatifs du sens. Par contre, pour la similarité textuelle, ce n'est pas le cas.

Ainsi, une similarité textuelle devra être capable de faire bien plus qu'une simple comparaison des symboles dans le texte. Il faudra donc également capturer des distinctions sémantiques de manière implicite (ou de créer une correspondance avec une ressource lexico-sémantique) et de pouvoir déterminer les mots du texte plus saillants sémantiquement, notamment au travers de connaissances préalables sur la syntaxe.

Aussi, ce type d'algorithmes est certainement plus adapté pour de l'induction automatique de sens ou encore pour construire des ressources lexico-sémantiques avant de pouvoir les aligner.

DÉTECTION DE PARAPHRASE ET IMPLICATION TEXTUELLE. Une implication textuelle est une relation orientée entre deux fragments de texte, tel que, au travers d'un raisonnement logique, la valeur de vérité d'un des fragments dépend de la valeur de vérité de l'autre. La tâche de détection d'implication textuelle en TAL consiste à partir d'un fragment source et d'un ensemble d'affirmations (appelées hypothèses dans le domaine), à classifier les affirmations selon si il y a ou non une relation implication sémantique entre le fragment source et l'affirmation en question.

Par exemple pour la phrase «Jean déteste l'homme qui a frappé Marie» quelles sont parmi les hypothèses suivantes celles qu'elle implique ou non : «Marie a été frappée par l'homme qui est haï par Jean» (*Vrai*), «Jean hait un homme» (*Vrai*), «Jean déteste Marie» (*Faux*).

C'est une tâche essentielle pour déterminer si une phrase constitue la bonne réponse à une question ou encore pour éliminer des phrases redondantes (résumé automatique) ainsi que pour générer des paraphrases valides. (BEDARIDE et GARDENT, 2008).

La similarité de cette tâche par rapport avec l'AAS est surtout liée au fait qu'il s'agit d'un problème de classification binaire où comme pour l'AAS on décide de relier ou non deux éléments de manière stricte plutôt que de calculer une similarité textuelle. En effet lors du calcul d'une similarité, on ne prend pas de décision, on ne fait que quantifier le proximité, alors qu'en implication textuelle il faut un critère de décision strict.

En contrepartie, même si dans les RLS il y a des sens dont nous devons comparer les définitions à celles d'autres sens, il y a des différences fondamentales dans la nature du texte en question. En effet l'identification de l'«implication textuelle» s'opère habituellement sur des phrases complètes sur la base d'une inférence, alors que dans les RLS, les définitions sont des phrase servant à illustrer le contexte où le mot est utilisé dans ce sens (elles sont descriptives) et non pas de valeur de vérité particulière.

Il est rare, par exemple, que la définition d'un hyponyme soit une phrase cohérente que l'on pourrait identifier comme découlant logiquement (de manière stricte) de la définition de son hypéronyme. Ainsi, la définition et les objectifs de la tâche d'implication textuelle et donc les critères de décision sont trop stricts par rapport à ce qui est attendu en AAS, même si les mêmes informations caractéristiques et techniques peuvent intervenir dans les deux cas. (MATUSCHEK, 2015, Chapitre 2).

PROXIMITÉ SÉMANTIQUE. Les algorithmes de proximité sémantique permettent de calculer la similarité (pas toujours une similarité au sens d'une métrique) de l'information sémantique portée par deux sens de mots (ou plus) tels que distingués dans une ressource lexico-sémantique. On peut distingues les mesures de similarité (au sens mathématique du terme) qui utiliseront uniquement les relation de type est-un (synonymie) alors que les mesures de proximité [1] pourront utiliser tout type de relations (y-compris de l'antonymie et de la meronymie).

Une mesure de proximité détermine pour deux sens d'une même ressource leur proximité sémantique, caractérisée par un réel entre -1 et 1 ou alors entre 0 et 1. Ici, -1 ou 0 sont les valeurs minimales de la proximité (les sens ne sont pas similaires du tout/sont opposés) et 1 est la valeur maximale de la proximité (les sens sont jugés identiques). La proximité sémantique est une tâche ancienne et essentielle dans le développement de systèmes de WSD à base de similarité.

Même avec un simple dictionnaire (peu de structure), il est possible de calculer cette proximité entre sens en utilisant l'algorithme de Lesk (LESK, 1986) qui consiste à compter le nombre de mots en commun entre les définitions. Cette approche et ses nombreuses extensions sont toujours très utilisées de nos jours et restent compétitives avec d'autre types de mesures de proximité sémantique.

Il existe cependant un large éventail de mesures de proximité sémantique classiques, développées principalement autour de *WordNet*, mais applicables dans nombre de RLS structurées contenant le même type d'information.

Ces mesures peuvent exploiter :

- les mots communs ou divergents des définitions – Lesk, Coefficient Dice, Indice de Jaccard/Tanimoto (ROGERS et TANIMOTO, 1960), Indice de Tverski (TVERSKY, 1977a), etc.,
- la structure du graphe des relations lexico-sémantiques (G. HIRST et ST-ONGE, 1998 ; LEACOCK et CHODOROW, 1998 ; RADA et al., 1989 ; Z. WU et PALMER, 1994),
- le contenu en information estimé par un corpus (annoté en sens ou non) (JIANG et CONRATH, 1997 ; D. LIN, 1998a ; RESNIK, 1995),
- ou une combinaison hybride (LI, BANDAR et MCLEAN, 2003 ; T. M. PILEHVAR, JURGENS et Roberto NAVIGLI, 2013 ; PIRRÓ et EUZENAT, 2010).

Il existe également des mesures fondées sur des informations multilingues CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI, 2015 ou encore sur des plongements sémantiques [2] (CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI, 2015 ; IACOBACCI, T. M. PILEHVAR et Roberto NAVIGLI, 2015 ; LAFOURCADE, 2006).

Les mesures de proximité sont calculées au sein d'une RLS spécifique alors qu'en AAS il est nécessaire de calculer la proximité des sens entre différentes RLS et potentiellement dans plusieures langues. Cependant, beaucoup de mesures peuvent également être appliquées entre différentes ressources, notamment les mesures comparant les définitions des sens. Les approches à base de plongements citées (CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI, 2015 ; IACOBACCI, T. M. PILEHVAR et Roberto NAVIGLI, 2015) sont fondées spécifiquement sur BabelNet, mais peuvent être adaptées pour fonctionner sur des RLS différentes.

1. *Semantic relatedness measures* dans la littérature.
2. En anglais *sens embeddings*, ou la représentation (plongement) des sens de mots identifiés dans une ressource lexico-sémantique sous forme de vecteurs dans un espace vectoriel.

Ainsi, les différentes types de mesures de proximité se calquent assez directement sur les différentes approches de l'AAS : Les approches dites "à base de similarité" vont utiliser des mesures de proximité sémantique non-spécifiques à une ressource particulière ; les approches à base de graphes exploitent les structures respectives des RLS à aligner et se rapprochent conceptuellement de certaines mesures de proximité qui exploitent la structure du graphe ; il y a enfin des approches hybrides qui combinent les deux. L'utilisation de techniques de plongements sémantiques pour la AAS n'est par contre pas très développée.

2.1.2 Techniques d'alignement automatique de sens

Nous passerons ici brièvement en revue les différents types d'approches pour l'AAS monolingue. Nous nous intéressons aux approches multilingues et par conséquent nous ne donnons pas le détail de tous les (nombreux travaux) d'AAS monolingue mais juste quelques exemples principaux. Pour une revue plus complète, le lecteur est invité à se référer à Matuschek (2015, Chapitres 4-6).

2.1.2.1 AAS à base de similarités

Les RLS admettent des structures et des paradigmes très différents, cependant il y a une constante qui est presque toujours présente : les sens de mots comportent des définitions et potentiellement des exemples d'usage. Les définitions et les exemples d'usage d'un sens ainsi que les définitions et exemples de sens reliés (par des relations lexico-sémantiques) constituent alors une source d'information que l'on pourra (presque) toujours exploiter dans le contexte de l'alignement de deux RLS dans la même langue.

L'idée générale peut être résumée en quelques étapes standards (Christian M. Meyer et Gurevych, 2011).

— Pour une entrée source dans l'une des ressources es_1, identifier la ou les entrée(s) cible(s) correspondante(s) ec_i (par exemple, même lemme, même catégorie grammaticale) et, pour chaque sens de l'entrée ou des sous-entrées de l'entrée $(ss_1, ss_2, \ldots, ss_n)$ on génère une liste de sens candidats à l'alignement correspondant aux entrées cibles identifiées $(sc_1, sc_2, \ldots, sc_k)$.
— Calculer la similarité de toutes les paires de sens source et cible $sim(ss_i, sc_j)$, où sim est une mesure de similarité sémantique.
— Pour chacune des paires de sens, si le score de similarité est supérieur à un certain seuil θ, un alignement entre les sens source et cible est créé.

La sélection du seuil est capitale pour obtenir un alignement de qualité, et est spécifique à chaque mesure de similarité. Le seuil est habituellement estimé manuellement lorsque l'on ne dispose pas d'un jeu de données d'évaluation (par exemple dans Christian M. Meyer et Gurevych (2011)). Mais l'estimation manuelle ou experte du seuil sont des tâches très fastidieuses, et il est préférable de pouvoir apprendre le seuil de manière automatique. Pour ce faire, il faut construire, pour un sous ensemble des données, un étalon de référence qui pourra ensuite servir à estimer le seuil optimal, en faisant varier la valeur du seuil et en conservant le seuil qui mène à la meilleure correspondance avec l'étalon de référence (habituellement exprimé en termes de précision et de rappel, voir Section 2.4.2.2). Cette approche est utilisée universellement, et le y compris lors des expériences préliminaires de la Section 4.1 du Chapitre 4.

Cet algorithme général peut être étendu de diverses manières, par exemple, comme le fait Matuschek (2015, Chapitre 4) en utilisant plusieurs mesures de

similarité en conjonction et en estimant autant de seuils d'alignement, et en n'acceptant l'alignement que si toutes les mesures de similarité dépassent le seuil en conjonction : $(sim_1(ss, sc) > \theta_1) \wedge (sim_2(ss, sc) > \theta_2) \wedge \ldots \wedge (sim_n(ss, sc) > \theta_n)$. Ceci aura l'effet de supprimer certains alignements incorrects à condition que les mesures de similarité expriment des informations complémentaires pertinentes pour la discrimination des sens, et aura mécaniquement comme conséquence l'augmentation de la précision et potentiellement une baisse du rappel.

Beaucoup de mesures de similarité et de proximité sémantiques qui peuvent être utilisées ici, notamment celles mentionnées à la fin de la Section 2.1.1.2. Les mesures spécifiques que l'on pourra utiliser vont entièrement dépendre des ressources qu'il faudra aligner en fonction des informations disponibles.

Les mesures fondées sur un calcul des mots en commun entre les définitions sont populaires car on peut les utiliser avec pratiquement toutes les ressources lexico-sémantiques, alors que les mesures à bases de taxonomie et de contenu informationnel, ne peuvent être utilisées qu'avec des ressources structurées comme *WordNet*.

De manière plus générale, au delà de la comparaison simple des recouvrements, une approches populaire pour l'AAS en particulier consiste à comparer les distributions de fréquence des termes décrivant chacun des sens. Cette comparaison peut se faire par la mesure cosinus sur les vecteurs de fréquence, qui est souvent utilisée comme mesure de référence.

$$\cos(s_1, s_2) = \frac{BoW(s_1) \cdot BoW(s_2)}{\|BoW(s_1)\| \|BoW(s_2)\|}$$

Cette approche a par exemple été employée par Ruiz-Casado, Alfonseca et Castells (2005) pour l'alignement des *synsets* Wordnet avec des pages Wikipédia ou encore par Christian M. Meyer et Gurevych (2011) et Niemann et Gurevych (2011) pour l'alignement Wordnet/Wiktionary.

L'utilisation de cette approche nécessite d'avoir des descripteurs (par exemple définitions, exemples d'usage, texte de la page Wikipédia) de sens suffisamment longs afin d'obtenir des distributions représentatives. Par ailleurs, pour que la comparaison ait un sens if faut que les deux vecteurs soient dans la même base. C'est ici un espace vectoriel où chaque dimension est un mot de l'union des deux descripteurs de sens.

Dans le cas ou les descripteurs sont courts il faut les enrichir avec une source d'information extérieure, par exemple un modèle sémantique vectoriel calculé à partir d'un corpus, où un vecteur est associé chaque sens[3] Schwab (2005).

Une autre technique populaire, initialement adaptée pour la WSD est l'utilisation de PageRank (Mihalcea, Tarau et Figa, 2004) et plus particulièrement d'une amélioration admettant des poids personnalisés (Agirre et Soroa, 2009). Cette approche a été largement utilisée, en particulier dans les travaux de Matuschek (2015), Christian M. Meyer et Gurevych (2011) et Niemann et Gurevych (2011). Cette technique fonctionne bien lorsque l'une des ressource est richement structurée (comme *WordNet*), cependant elle est difficile à appliquer directement pour aligner deux ressources sans structure ou avec une structure limitée, par exemple la paire (Wikipédia, Wiktionary).

3. par exemple moyenne des vecteurs associés à chaque mot de la définition, ou par contextualisation faible comme expliqué ultérieurement dans la Section 2.4.1.2

2.1.2.2 Algorithmes d'alignement automatique de sens structurel

Dans les approches à base de similarité, on peut être amené à utiliser des informations provenant de la structure des ressources, cependant, cette information est utilisée pour permettre une extension des définitions ou simplement utilisée comme une liste plate de symboles pour le calcul du recouvrement entre deux sens.

Il y a cependant des approches qui utilisent uniquement la structure du graphe pour arriver à un alignement de sens, sans exploiter les informations présentes dans les définitions ou les exemples d'usage. Ces approches sont apparues comme une généralisation des mesures de similarité exploitant des distances dans la structure du graphe taxinomique (Leacock et Chodorow, 1998 ; Rada et al., 1989 ; Z. Wu et Palmer, 1994), ou des constructions plus complexes telles que les chaînes lexicales, qui exploitent des enchainements transitifs dans la taxinomie pour modéliser le contexte d'usage des mots. L'utilisation de la structure a également montré son potentiel pour des ressources autres que WordNet, telles que Wiktionary (Zesch, Müller et Gurevych, 2008d) et BabelNet (Roberto Navigli et Ponzetto, 2012b), ce qui suggère qu'elles sont particulièrement convenables pour l'alignement de sens avec des ressources hétérogènes.

L'une des premières utilisation faite de ce type d'approche a été pour aligner des synsets de *WordNet* à la hiérarchie de catégories de Wikipédia (Toral, Muñoz et Monachini, 2008), avec comme objectif d'enrichir les synsets *WordNet* par des entités nommées provenant de Wikipédia. Dans le cas de mots polysémiques, ils utilisent des motifs de graphe pour comparer les taxonomies de WordNet et des catégories Wikipédia afin de trouver des correspondances entre les hypéronymes dans WordNet et les sous catégories dans Wikipédia. Cette approche est très spécifique aux deux ressources et se limite aux relations "instance-de" qui sont une infime partie de la structure des ressources et qui ne sont jamais présentes pour certaines catégories grammaticales, telles que les verbes. Par ailleurs, le graphe de catégories de Wikipédia n'est qu'une infime partie de l'ensemble de toutes les relations et informations présentes dans Wikipédia. Ponzetto et Roberto Navigli (2009) proposent une méthode qui a la même finalité, mais qui utilise cette fois les relations "est-un" et construit un sous graphe de WordNet pour chaque catégorie de Wikipédia et alignent le synset qui a la meilleure correspondance. Les limitations sont également les mêmes que pour Toral, Muñoz et Monachini (2008).

SSI-Dijkstra+ (Laparra, Rigau et Cuadros, 2010) est la première approche à pleinement exploiter la structure du graphe en appliquant un algorithme de recherche du chemin le plus court dans un graphe. Leur objectif est d'aligner les UL de FrameNet (voir **Définition** 1.5) avec les *synsets* WordNet. Ils identifient d'abord les UL monosémique pour les aligner aux *synsets* WordNet équivalents et puis, ils alignent chaque UL dans le même cadre (**Définition** 1.5) qu'une UL monosémique avec les *synsets* les plus proches dans le graphe WordNet enrichi de eXtended WordNet. Cette approche est limitée car si le cadre ne contient que des UL polysémiques, alors ils alignent l'UL avec le sens le plus fréquent dans WordNet. D'autre part seule la structure de WordNet est considérée, les UL sont considérées comme étant uniquement du texte, car *SSI-Dijkstra+* est initialement un algorithme de WSD qui utilise WordNet comme ressource. Pour ces raisons, cet algorithme est difficile à appliquer à d'autres ressources. PAr ailleurs les ex-

tensions eXtended WordNet n'existent pas pour d'autres ressources telles que Wiktionary.

Un autre type d'approche peut être qualifié comme étant interne à une ressource, c'est-à-dire visant à aligner des éléments dans une même ressource afin de l'enrichir, par inférence vis-à-vis des éléments existants. C'est notamment le cas du travail de Roberto NAVIGLI (2009a) qui vise à annoter les gloses associées aux *synsets* WordNet en sens en exploitant les cycles dans le graphe *WordNet*.

FLATI et Roberto NAVIGLI (2012) étendent cette approche à un contexte translingue qui vise à désambiguïser les traductions dans un dictionnaire bilingue[4]. Ces deux approches sont très similaires à une tâche d'AAS.

C'est sur la base de cette idée de recherche du chemin le plus court que MATUSCHEK (2015) développe l'algorithme *Dijkstra-WSA* qui est une généralisation de cette approche afin d'être indépendant de la langue et au type de ressources.

Le principe de l'algorithme est de d'abord traiter les cas triviaux (entrées de lemme équivalent issus de part et d'autre des deux ressources ayant chacune un seul sens[5]) puis de passer aux cas des mots polysémiques. Pour cela, pour deux lemmes équivalents, ils récupèrent tous les sens et pour chaque sens pas encore aligné, ils calculent tous les chemins possibles vers les sens candidats au travers de la structure des deux ressources ainsi que des alignements déjà existants et sélectionnent le candidat se trouvant le plus près.

Ce calcul du chemin le plus court s'effectue grâce à l'algorithme de Dijkstra, d'où le nom de la méthode. Cette approche à l'avantage de ne pas faire de suppositions sur les ressource (notamment sur la présence ou non de définitions). Cet algorithme, combiné avec une approche de repli par similarité (pour les sens qui n'ont pu être aligner) a permis d'aligner de nombreuses ressource, notamment dans le cadre de la construction de UBY (MATUSCHEK, 2015).

2.1.2.3 Algorithmes d'alignement automatique génétiques

Les techniques d'AAS génériques vont combiner des approches à base de similarité et à base de structure en les traitant comme des traits qui permettent ensemble de décider si oui ou non deux sens doivent être alignés. Une telle combinaison est différente d'une utilisation par défaut d'une deuxième technique (stratégie de repli) comme utilisée dans MATUSCHEK (2015, Chapitre 5) par exemple. Ici, des éléments provenant à la fois de similarité et de structure peuvent être combinés directement afin de bénéficier des informations complémentaires apportées par chaque mesure. Globalement nous pouvons distinguer les méthodes non supervisées à base de connaissance et le méthodes d'apprentissage supervisé.

Les travaux principaux entrant dans la première catégorie ont produit des ressources massives et utilisées à très grande échelle, à savoir Open Multilingual Wordnet (BOND et FOSTER, 2013) et BabelNet Roberto NAVIGLI et PONZETTO, 2012a, qui ont été décrites quantitativement dans la Section 1.3.

BOND et FOSTER (2013) utilisent Wiktionary comme base pour leur permettre d'aligner de nombreux WordNets entre eux. Ils utilisent fondamentalement une

4. Dans un dictionnaire bilingue les traductions sont données entre les mots indépendamment des sens spécifiques ; les désambiguïser c'est les ramener aux sens de mots correspondants.

5. Soit le même lemme dans la même langue des deux côtés ou un lemme équivalent par traduction

approche de similarité des gloses, cependant ils utilisent aussi une mesure exploitant les liens de traduction, en considérant que deux sens qui partagent beaucoup de traductions communes sont équivalents.

Comme le remarque MATUSCHEK (2015, Chapitre 6), cette approche souffre de la faible densité du graphe de traduction, en particulier pour les éditions de langues moins riches, où la complémentation par un système de traduction automatique n'aide pas car ce sont souvent des langues peu dotées.

Roberto NAVIGLI et PONZETTO (2012a) utilisent le graphe des relations WordNet aligné avec Wikipédia comme base et l'étendent à plusieures langues en exploitant les liens de traduction entre les pages Wikipédia.

Ils créent d'abord des contextes de désambiguïsation à partir du texte des pages Wikipédia mais aussi certains liens hypertextes vers d'autres pages Wikipédia afin de calculer les probabilités de correspondance entre les *synsets* WordNet et les pages Wikipédia par un processus de désambiguïsation lexicale utilisant à la fois des recouvrements de type sac-de-mots mais aussi sur la base d'une recherche de chemin de longueur maximale dans le graphe.

Lorsque l'alignement est effectué, les *synsets* vont être enrichies pour devenir des Babel*synsets*, en exploitant les liens de traduction inter-pages, les liens de redirection en anglais, ainsi que dans la langue cible d'un lien de traduction. Les gloses des *synsets* sont enrichis avec des phrases des pages Wikipédia et *SemCor*. Tous les trous sont comblés avec un système de traduction automatique.

BabelNet a depuis beaucoup grandi et évolué ; sa version actuelle contient plus de 256 langues. Les techniques d'alignements se sont perfectionnées, en particulier le système de M. T. PILEHVAR et Roberto NAVIGLI (2014), qui est une généralisation hybride du processus d'alignement de ressources à base de similarité où PPR est utilisé de manière générique pour la génération de signatures sémantique hybrides, à la fois structurelles et à base de définitions.

La Figure 2.2 présente un traduction du schéma issu de M. T. PILEHVAR et Roberto NAVIGLI (2014) illustrant leur approche d'alignement. Elle permet d'exploiter soit les définitions, exemples d'usage ou tout autre texte portant sur le sens, potentiellement enrichis par des relations lexico-sémantiques provenant soit de la même ressource, soit d'une ressource externe, dénotée par "Graphe H" dans la ressource.

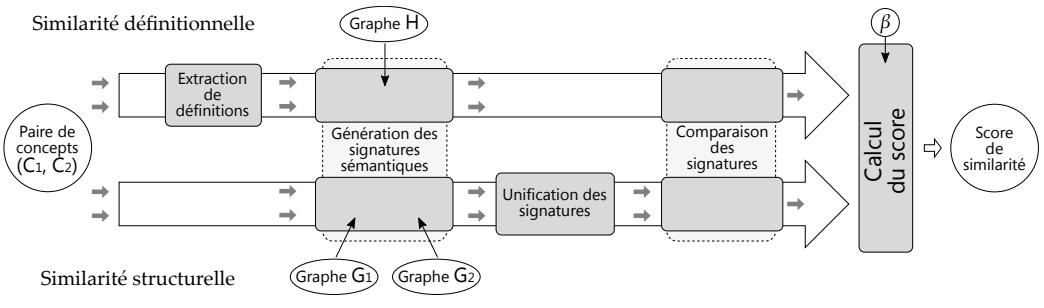


FIGURE 2.2 – Le modèle d'alignement de RLS hétérogènes de M. T. PILEHVAR et Roberto NAVIGLI (2014).

En parallèle, il est possible d'exploiter des propriétés structurelles du graphe (degré de centralité, cliques, partions, coloriages). On pourra ainsi générer une signature sémantique pour chaque type de traits sous la forme d'une matrice, sur

laquelle nous pourrons appliquer des traitements (enrichissement, PPR, réduction de dimensionnalité, etc.).

La similarité structurelle pouvant contenir des informations provenant de plusieurs graphes, il faudra la normaliser pour s'assurer par exemple que tous les vecteurs soient exprimés dans le même repère. Il devient ainsi possible de comparer les signatures afin de calculer un score pour les deux types de signatures puis d'unifier les scores en un score global en utilisant un paramètre $\beta \cdot score_1 + (1 - \beta) \cdot score_2$ donnant l'importance relative d'une score par rapport à l'autre.

Les approches supervisées extraient des traits, d'une manière similaire à M. T. PILEHVAR et Roberto NAVIGLI (2014), cependant les différences principales avec les approches à base de connaissances est qu'un algorithme d'apprentissage supervisé est utilisé pour estimer les poids optimaux pour la combinaison des traits.

FERRÁNDEZ et al. (2010) proposent une technique semi supervisée pour l'alignement de FrameNet et de WordNet. La technique d'alignement en elle même est similaire à *Dijkstra-WSA* de MATUSCHEK, 2015. Pour chaque entrée de *WordNet* correspondant à une entrée FrameNet par son lemme et sa partie du discours puis pour chaque sens WordNet et UL FrameNet des entrées correspondantes, ils construisent un graphe de voisinage autour des sens qui contient toutes les relations desquelles le sens est la source ou la cible.

Ils calculent un score de similarité normalisé correspondant à la moyenne des distances entre les éléments du voisinage et l'élément central. Ils combinent cela à une mesure de similarité textuelle. Ils utilisent une technique d'apprentissage à partir de 100 exemples d'alignements manuels pour estimer les poids de chaque relation à la fois dans FrameNet et dans WordNet.

MELO et WEIKUM (2009) proposent une méthode pour créer une ressource dans une autre langue (et non pour aligner deux ressources existantes) en utilisant la structure de WordNet comme base.

Pour chaque *synset*, un ensemble de traductions candidates est rassemblé, et un modèle de classification est entrainé à partir de données annotées manuellement pour décider si une traduction particulière peut donner lieu à la création d'un *synset* équivalent dans une autre langue.

D'après MATUSCHEK (2015, Chapitre 6) Cette tâche est plus facile que l'alignement de ressources existantes, sans compter que cette approche utilise des traits spécifiques à WordNet.

La technique proposée par MATUSCHEK (2015, Chapitre 6) est plus générale, car elle n'utilise que des traits pouvant se trouver dans une grande partie des ressources lexico-sémantiques, en particulier la catégorie grammaticale, l'ordre d'apparition du sens (indice), le nombre de relations de traduction en commun, et des motifs dans les phrases d'exemple.

Ainsi, pour chaque sens dans une ressource, une liste de candidats est créée, et pour chaque candidat, les traits caractéristiques des sens sont extraits. Un sous ensemble des candidats alignés manuellement par des experts est utilisé pour entrainer un modèle de classification par validation croisée à 10 croisements qui est ensuite utilisé sur l'ensemble des candidats pour prendre une décision d'alignement.

De nombreux algorithmes de classification standards sont évalués, *Naïve Bayes*, Réseaux bayésiens, perceptrons, **S**éparateur à **V**aste **M**arge (SVM), arbres de décision. Selon les objectifs (rappel maximal, précision maximale, etc.). Le choix de l'algorithme le plus performant dépendra des paires de ressources . Le désavantage principal de cette approche ainsi que de toutes les approches pleinement supervisées, c'est qu'elles sont limitées au niveau de la couverture, car une annotation manuelle est quelque chose de coûteux.

2.1.3 Passage au multilingue

L'AAS dans un contexte multilingue emploie tous les types de techniques décrites dans la section précédente, mais nécessitent l'utilisation d'une source de savoir translingue pour pouvoir établir des correspondances entre les sens de différentes langues.

On distingue différentes approches en fonction des trois types d'algorithmes d'AAS pour l'établissement de cette correspondance translingue :

1. Pour l'AAS à base de similarité, une approche naïve standard, qui semble plutôt bien fonctionner, est l'emploi d'un système de traduction automatique. Le système de traduction peut ainsi être utilisé pour traduire les gloses de description des sens afin de permettre à une approche de similarité à base de recouvrement de comparer les différents sens. C'est notamment le cas pour la construction d'Open Multilingual WordNet (Bond et Foster, 2013) ou encore dans les cas d'alignements dans deux langues différentes dans toutes les approches proposées par Matuschek (2015). Les gloses multilingues de BabelNet sont également enrichies avec de la traduction automatique, ce qui permet de potentiellement étendre la ressource ultérieurement avec des approches à base de similarité, même si ce n'est pas le mode de construction principal de BabelNet.

   Certaines approches qui visent à la construction de ressources pouvant potentiellement être étendues à l'alignement de sens, utilisent un système de désambiguïsation translingue, en désambiguïsant par exemple les gloses des sens d'une ressource avec les sens d'une autre. C'est le cas dans les travaux de (Apidianaki et Sagot, 2012), et c'est aussi une approche recommandée par les créateurs de Babelfy (Moro, Raganato et Roberto Navigli, 2014) et déjà utilisée dans des cas d'applications pour l'alignement de sens (Maarouf et al., 2015), en particulier dans le contexte des données lexicales liées, ainsi que pour l'alignement d'ontologies (Lesnikova, David et Euzenat, 2015).

2. Pour les AAS à base de graphe, un ensemble de liens de traduction est typiquement utilisé. Cet liens proviennent soit d'un graphe de traduction comme celui présent dans Wiktionary (Hanoka et Sagot, 2012 ; Sajous, Navarro et al., p.d.), soit d'une graphe induit, comme dans les projets PanLex (Kamholz, Pool et Colowick, 2014) et PanDictionary (Mausam et al., 2009).

Un type d'approche sous exploité en AAS est l'utilisation de plongements sémantiques multilingues en combinaison avec des traits issus de la structure du graphe ou des gloses. Même si l'idée n'est pas nouvelle, comme nous le verrons dans la Section 2.3, la création de véritables plongements multilingues est relative-

ment récente et est restée principalement limitée à deux langues (GOUWS, BENGIO et CORRADO, 2015 ; VULIĆ et MOENS, 2015b ; ZOU et al., 2013) et confinée à des applications de traduction automatique.

CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI (2015) proposent une technique qui permet d'obtenir une représentation vectorielle multilingue avec des vecteurs comparables, en calquant les dimensions des vecteurs sur les Babel*synsets*. La construction des vecteurs se fait au travers d'un sous corpus de pages Wikipédia associées à toutes les pages liées au BabelSynset, sur la base d'une mesure de spécificité lexicale. Ces vecteurs sont principalement utiles pour calculer la similarité entre des sens multilingues présents dans BabelNet, ce qui en théorie permettrait de les utiliser pour de l'AAS de manière spécifique à BabelNet.

Une approche plus générale est celle récemment proposée en pré-publication sur arXiv par AMMAR et al. (2016). Elle n'utilise que des corpus monolingues et des dictionnaires bilingues (potentiellement construits à partir de corpus parallèles) devraient être beaucoup plus adaptée à une utilisation d'AAS, à condition de pouvoir projeter le modèle vectoriel sur les ressources à aligner, comme par exemple avec la méthode proposée par FARUQUI et al. (2015).

Une limitation habituelle des approches d'AAS multilingues, est soit d'une part le fait qu'elles sont limitée à deux langues, soit si elle ne sont pas limitées à deux langues l'utilisation d'une pivot en langue naturelle. Ces limitations tiennent également pour les techniques vectorielles détaillées ci-dessus, puisque pour CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI (2015) les vecteurs sont calqués sur les Babel*synsets*, qui sont un pivot en langage naturel et pour AMMAR et al. (2016), les vecteurs anglais sont explicitement choisis comme pivots. Cependant, pour AMMAR et al. (2016), il est expliqué comment on pourrait ne pas utiliser de langue particulière comme pivot. Ces limitations engendrent des problèmes de perte de contraste, comme nous le détaillerons ci-dessous.

2.2 Contrastes et relations de traduction

Comme présenté dans la **Définition** 2.1, un contraste dans une ressource lexico-sémantique peut être de nature sémantique ou linguistique. Les sous-sections qui suivent donnent respectivement des exemples de contrastes sémantiques puis linguistiques et tentent d'apporter des éléments issus de la recherche en linguistique sur la prévalence de ces contrastes dans les données langagières en général et dans les ressources lexico-sémantiques en particulier.

2.2.1 Contrastes sémantiques

Dans les cas de problèmes cités dans la Section 1.2.3 du Chapitre 1, on ne peut pas directement lier les acceptions (voir le Chapitre 3 pour une définition formelle) provenant de toutes les langues ensemble même si il y a une correspondance sémantique partielle. La Figure 2.3 prend l'exemple de l'alignement des acceptions des mots/sens correspondant à "riz" dans quatre langues (français, anglais, japonais, bengla).

En bengla et en japonais, riz dans le sens du grain de riz comestible peut être traduit de deux manières distinctes, ce qui corresponds à une distinction séman-

Définition 2.1 Contraste

Un contraste dans une base lexico-sémantique est une distinction à faire lors de l'alignement entre des volumes monolingues quand une entité dans l'un des volumes ne peut être alignée à exactement une entité dans un autre volume du fait d'une différence de granularité au niveau de l'information portée par les éléments. En d'autres termes, l'une des entités englobe plus d'information et doit être aligné à plusieurs entités plus spécifiques dans les autres volumes.
Un contraste peut porter sur l'information sémantique à proprement parler (**contraste sémantique**[a]) ou sur la granularité de distinctions de nature linguistique (**contraste linguistique**[b]).

a. Voir la Section 2.2.1 pour des exemples portant sur les contrastes sémantiques.
b. Voir la Section 2.2.2 pour des exemples portant sur les contrastes linguistiques.

tique plus fine qu'en français ou en anglais. Ainsi, relier les acceptions distinctes qu'on trouve dans les deux langues asiatiques avec l'unique acception correspondant au sens français ou anglais pour signifier qu'elles sont équivalentes voudrait dire que toutes les acceptions monolingues sont reliées à la même acception interlingue, ce qui est une erreur qui induit une perte de contraste. En d'autres termes nous ne pourrons pas obtenir une correspondance exacte entre le sens de "bhat" et le sens de "gohan" si on considère ces sens strictement équivalents à l'acception *"occidentale"*.

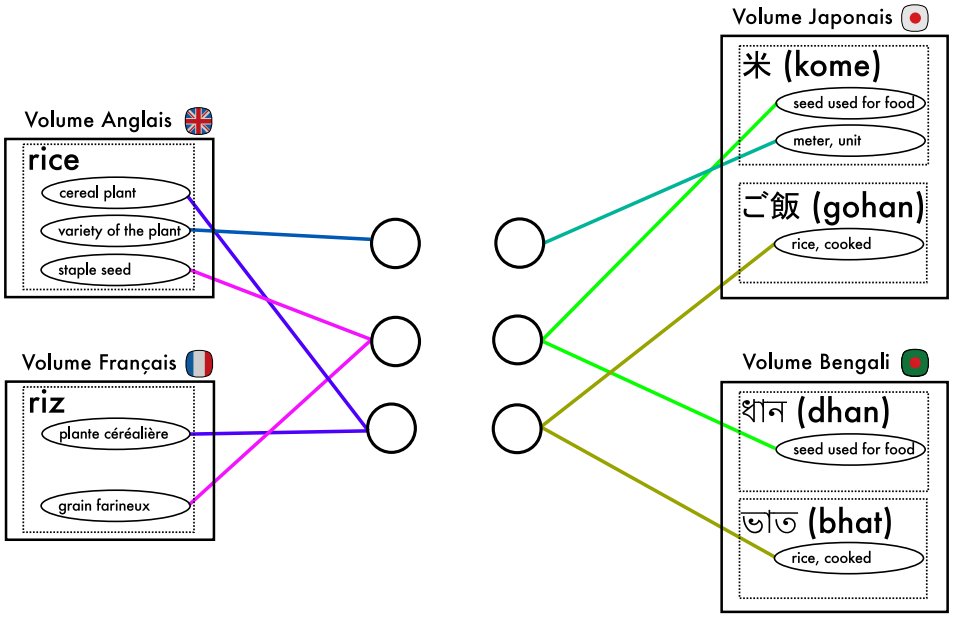


FIGURE 2.3 – Illustration d'acceptions interlingues reliant des sens jugés équivalents entre eux. Les sens bengalis et japonais ne peuvent être directement reliés aux sens français et anglais pour 'grain céréalier' car on ne pourrait plus distinguer les lexicalisations divergentes.

2.2.2 Contrastes linguistiques

Cependant, il se peut également qu'une relation de raffinement n'ait pas une interprétation sémantique, mais plutôt linguistique. En effet, il peut arriver qu'il y ait des phénomènes contrastifs d'ordre linguistique et non sémantiques, par exemple au niveau des distinctions grammaticales sur les parties du discours.

REPRENONS l'exemple de Sérasset (1994) pour le mot Chinois 工作 (Gōngzuó) qui peut correspondre à l'acception française "travailler (de manière générale)" qui correspond à un aspect verbal ou alors à l'acception "travailler de ses mains/travail manuel#nom", qui en français a un aspect nominal. Or, dans les volumes d'acceptions monolingues, l'un des points de distinction entre les entrées lexicales d'un même vocable sont les catégories grammaticales des mots, ce qui implique qu'il faudra choisir entre aligner 工作 à l'entrée verbale ou a l'entrée nominale, ce qui mènerait à une perte de contraste au détriment de nombreuses applications, notamment de traduction.

2.2.3 Contrastes artificiels et pivot naturel

La présence de tels cas de contrastes, à la fois sémantiques comme linguistiques ne posent pas de problèmes lors de l'alignement de ressources issues uniquement de deux langues, car le contraste est préservé par un alignement direct des sens plus spécifiques au sens plus général. Dans un tel cas nous ne savons lequel des sens est le plus général sans un calcul préalable.

Dès lors qu"il s'agir d'aligner des ressources multilingues (plus de deux), le choix de la représentation pivot est déterminant. En effet, se contenter d'un alignement direct de tous les sens plus spécifiques au sens le plus général, implique que le sens le plus général (et de la/des langues auxquelles il appartient) prends le rôle pivot : il n'y as pas de lien direct entre les sens plus spécifiques dans les langues autres que cette du pivot. Ainsi dans l'exemple de "riz" donné dans la section précédente, utiliser le sens anglais ou français comme pivot mène à une perte artificielle de contraste (voir Figure 2.4).

2.3 Acceptions interlingues – vers un objectif idéal ?

En se basant sur les travaux de Farwell, L. Guthrie et Wilks (1992, 1993) sur le projet ULTRA mais aussi sur les travaux de linguistes en sémantique lexicale tels que Mel'čuk (1999, 2006) avec le DEC, Sérasset (1994) introduit la notion d'acception interlingue comme une représentation interlingue mettant des acceptions monolingues (**Définition** 1.6) issues de volumes monolingues de référence et partageant le même signifié (sens jugés «équivalents») en relation.

Du fait des problèmes de contraste sémantiques et linguistiques prévalants dans les RLS par pivot naturel, il est impossible de relier tous les sens alignées par transfert entre eux par une classe d'équivalence unique. C'est ce que Sérasset (1994) appelle les problèmes de raffinement de sens entre différentes langues. Ainsi Sérasset (1994) propose l'établissement d'une relation de raffinement qui permet d'indiquer qu'il y a une correspondance partielle entre des acceptions interlingues mettant en relation les acceptions des différentes langues, comme

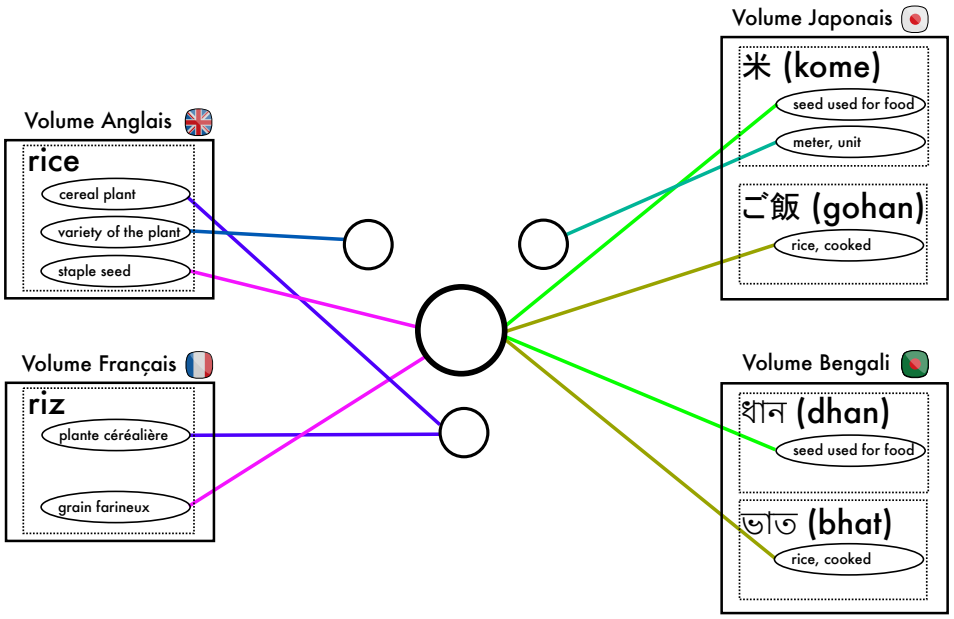


FIGURE 2.4 – Illustration d'un pivot en langue naturelle qui mène à une perte artificielle de contraste.

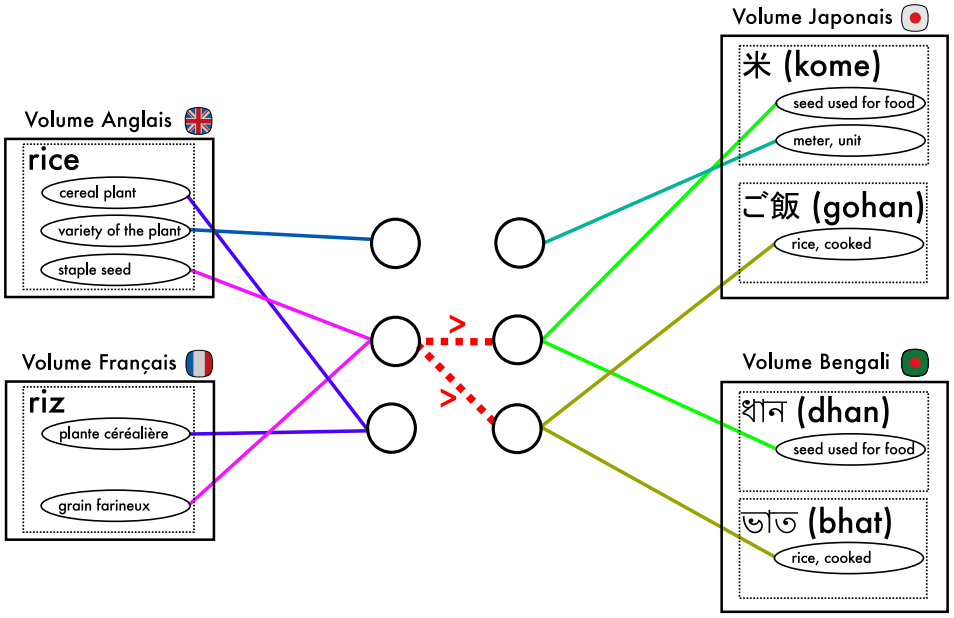


FIGURE 2.5 – Illustration d'acceptions interlingues reliant des sens jugés équivalents entre eux avec l'addition de la relation de raffinement. Les sens bengalis et japonais sont maintenant reliés aux sens français et anglais pour 'grain céréalier' sans perte de contrastes.

l'illustre la Figure 2.5. Dans ce cas l'acception occidentale correspond à l'union des acceptions liés par un lien de raffinement.

Dans son travail, SÉRASSET (1994) se focalise sur la manipulation et la représentation de RLS d'un point de vue logiciel, ainsi la construction des acceptions en elles-mêmes doit se faire de manière manuelle. Le système génère plusieurs alignements possibles et laisse alors des experts faire le choix le plus adéquat. Le système original ne peut s'appliquer à une très grande échelle du fait de limitations

techniques. Cependant, ce système a donné naissance au Projet Papillon-Nadia, que nous avons déjà présenté dans le Chapitre 1, qui offre la possibilité d'une construction collaborative de RLS à plus grande échelle sur la base du paradigme d'acceptions interlingues.

Pour des raisons de compétences, il fut difficile de trouver des experts parlant suffisamment de langues pour être à même de construire des acceptions interlingues cohérentes. En effet, lors de l'alignement de trois langues c'est une tâche relativement aisée, cependant au delà de 4-5 langues on atteint les limites du multilinguisme humain et très peu d'experts seront à même de maitriser les particularités sémantiques de chaque langue de manière suffisamment avancée, ce qui mène inévitablement à une construction très fastidieuse et potentiellement erronée.

Ainsi, il est essentiel de pouvoir automatiser la construction de telles ressources, du moins à un degré qui puisse faciliter le travail des experts et restreindre le nombre de langues de travail.

2.4 Vers une construction et validation automatique ?

Sérasset (1994) présente quelques pistes pour assurer la consistance et la validité des données. De par la nature des acceptions interlingues, des sens alignés sur la même acception peuvent être considérés comme synonymes. La structure formée par les liens entre acceptions doit être cohérente avec les relations de synonymie présentes dans chaque ressource. Dans le cas contraire, il existe une incohérence au niveau de la ressource qui peut être imputée soit à une erreur ou un manquement au niveau des acceptions ou alors à des erreurs dans les volumes monolingues.

Un autre signe d'incohérence qu'il identifie est la présence de boucles dans le réseau des relations de synonymie. En effet, si toutes les acceptions sont équivalentes entre elles, pourquoi sont-elles des acceptions différentes plutôt qu'une acception fusionnée les englobant toutes ?

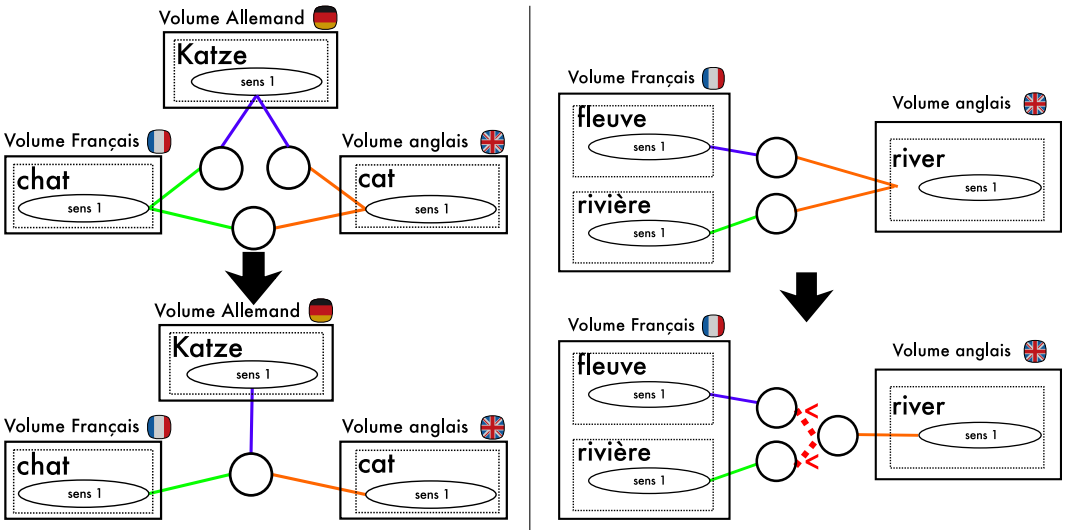
2.4.1 Approches existantes pour la construction d'acceptions

Suites aux travaux de Sérasset (1994), l'objectif de la thèse de Teeraparbseree (2005) était de reprendre le concept d'acceptions multilingues et de proposer des méthodes et outils pour la construction automatique et l'évaluation de RLS à base d'acceptions.

2.4.1.1 Maintien de la cohérence

Dans ce travail, quelques opérations de base pour la cohérence et la construction itérative d'acceptions sont définies, qui correspondent à certains cas «patron» qui pourraient apparaître dans un alignement en acceptions de RLS :

- **Groupage d'acceptions** Lorsqu'un sens est lié à plusieurs acceptions qui elles même sont liées à des sens correspondant à des lexicalisations différentes, alors les acceptions peuvent être fusionnées (Figure 2.6a).
- **Raffinement d'acceptions** Dans une base d'acceptions, si N sens monolingues sont reliés deux à deux de manière de former un cycle transitif, alors les acceptions en question sont équivalentes et peuvent être fusionnées (Figure 2.6b).



(a) Groupage d'acceptions. **(b)** Raffinement d'acceptions.

FIGURE 2.6 – Opérations de TEERAPARBSEREE (2005) sur des acceptions existantes.

Ces opérations pourront être appliquées à n'importe quelle RLS multilingue alignée par d'acceptions interlingues. Cependant deux difficultés peuvent se présenter :

— de construire des acceptions partielles entre différentes langues,
— puis de détecter les patrons en questions et de quantifier leur prévalence dans les différentes ressources alignées.

2.4.1.2 Construction initiale d'acceptions

Une réponse partielle est donnée à la première question dans le contexte de l'état de l'art de l'époque, au travers de trois approches possibles pour la création d'une ressource à base d'acceptions interlingues :

APPROCHE <<ONTOLOGIQUE>> L'approche ontologique consiste à utiliser une ontologie comme pivot et à relier les sens des volumes monolingues vers les concepts correspondants dans l'ontologie. TEERAPARBSEREE (2005) décrit quelques approches de l'époque, sans toutefois justifier le fait que ce n'est pas l'approche retenue pour le travail en question. Cette approche pose cependant de nombreux problèmes fondamentaux.

D'une part la granularité des concepts dans une ontologie est très grande, ce qui veut dire qu'il est probable d'arriver à des problèmes de contraste artificiels comme dans une approche à pivot en langue naturelle (WITT et al., 2009).

De plus, les concepts dans une ontologie relèvent d'une représentation en sémantique formelle fondée sur les logiques de description, qui permet un raisonnement sur les instances de l'ontologie. Les axiomes définissant cette sémantique formelle ne sont cependant pas applicables à des ressources lexico-sémantiques (transitivité inexistante dans le cas général, incohérences logiques inhérentes aux langues naturelles).

Ainsi, l'inférence de nouvelles relations dans l'ontologie par un raisonnement pourraient résulter à des incohérences et à des erreurs dans les RLS.

Depuis 2005, cette approche ontologique a évoluée et a pourtant aboutie, paradoxalement, aux initiatives autour des données lexicales liées et à la création de

lemon et d'Ontolex. En effet la finalité de ces projets est de lier les sens de mots et les données lexicales en général à une ontologie.

Cependant, cette philosophie n'impose pas que l'ontologie serve de pivot, ce qui permet le «détournement» de n'utiliser ces approches uniquement comme des outils pour l'accès et la représentation des données et non pour le raisonnement.

APPROCHE LINGUISTIQUE ET TRADUCTIONNELLE TEERAPARBSEREE (2005) se focalise principalement sur l'acquisition de données lexicales à partir de dictionnaires bilingues multiples. L'idée est d'utiliser les dictionnaires bilingues pour trouver des liens de traduction mot à mot et d'utiliser une inférence transitive dans certains cas, quand il n'y a pas de dictionnaire bilingue direct. Cette approche peut mener à de nombreuses erreurs dans des situations de lexicalisations divergentes (MAUSAM et al., 2009).

C'est pourquoi il est proposé d'utiliser une consultation inverse pour éviter de diverger. Considérons l'exemple suivant : nous avons un dictionnaire français vers anglais et un dictionnaire japonais vers anglais à disposition, mais nous voulons un dictionnaire japonais vers français. Ainsi pour connaitre la traduction du mot *kyousou* en français, nous consultons d'abord le dictionnaire japonais -> anglais et obtenons l'ensemble de traductions $T_{en} = \{$ *race*, *contest*, *competition* $\}$, ensuite nous consultons les traductions de chacun des mots de l'ensemble pour obtenir un nouvel ensemble de traductions en français cette fois $T_{fr} = \{$ race, course, compétition, concours, hâte $\}$.

Cependant cet ensemble est beaucoup trop grand et que le mot japonais ne se traduit que par certains de ces mots, mais pas tous. Ainsi la dernière étape consiste à faire une consultation inverse de chaque mot de l'ensemble T_{fr} vers l'anglais : $Ti_{en} = \{$ ancestry, race, contest, competition, match $\}$, enfin on réalise une deuxième consultation inverse pour les mots de Ti_{en} vers le japonais $Ti_{jp} = \{$ *senzo*, *jinshu*, *kyousou*, *kyougi* $\}$. L'étape suivante est de déterminer quels sont les mots de Ti_{en} se traduisent par *kyousou* et on ne conserve que ceux là (*contest*, *competition*, *race*), puis on garde uniquement les mots français qui se traduisaient par les mots anglais gardés après la deuxième consultation inverse (concours, course, compétition). La Figure 2.7 illustre l'approche.

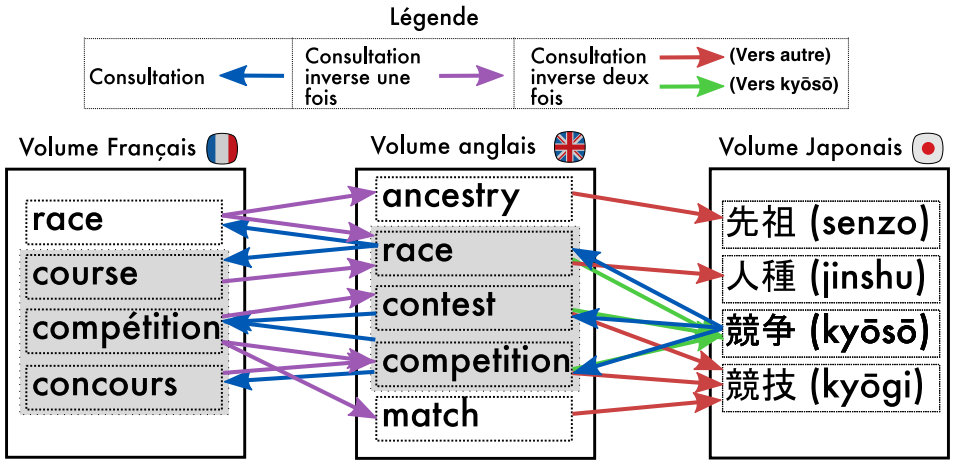


FIGURE 2.7 – Illustration du principe de consultation inverse pour des paires de langues non couvertes.

Cette approche est utile dans un contexte général pour enrichir un graphe de traduction, cependant elle est limitée à une triangulation entre trois langues et

se limite uniquement à la production d'un graphe au niveau des mots et non au niveau des sens. Les données présentes dans une ressource telle que DBNary contiennent déjà de nombreuses paires de langues au niveau des traductions et pour un certain nombre peuvent être ramenées aisément au sens. Ainsi l'extraction de Wiktionary dans DBNary constitue clairement un meilleur point de départ. Par ailleurs, les approches de triangulation pour des graphes de traduction sont de nos jours bien plus avancées et correspondent à des généralisations à un nombre arbitraire de langues, comme nous l'avons en partie vu dans la Section 1.3.2.4. Nous détaillerons ce point d'avantage dans le Chapitre 4, comme une étape importante de pré-traitement.

APPROCHE <<SÉMANTIQUE>> L'approche sémantique est présentée comme exploitant les projections de données linguistiques dans des espaces vectoriels, appelés maintenant des plongements sémantiques. Cette approche est assez commune, notamment par l'utilisation de techniques telles que **A**nalyse **S**émantique **L**atente (ASL). TEERAPARBSEREE (2005) propose d'utiliser ce type de représentation en conjonction avec les techniques à base de dictionnaires bilingues vues précédemment afin de générer des acceptions interlingues. Le type particulier de représentation vectorielle utilisée, repose sur les vecteur conceptuels (LAFOURCADE, 2006).

Les vecteurs conceptuels ont une construction différente à l'utilisation de ASL ou de modèles récents à base de plongements : ils sont calculés à partir de l'arbre résultat de l'analyse morpho-syntaxique d'un corpus de manière hiérarchique, afin de capturer le pouvoir discriminant de la syntaxe et de la morphologie du contexte d'usage des mots en plus de simple cooccurrences. Le calcul se fait de manière itérative, en montant et descendant successivement dans l'arbre d'analyse et en appliquant une pondération dépendante des fonctions morpho-syntaxiques (par exemple en affectant un poids différent à la fonction «objet» par rapport à la fonction «modalité»), comme détaillé par LAFOURCADE et GUINAND (2005).

Axiome 2.1 **Similarité cosinus** – $\cos(\widehat{XY})$

Soient deux vecteurs, X, Y sémantiques ou conceptuels. La distance cosinus est le cosinus de l'angle entre les deux vecteurs qui se calcule par le produit scalaire entre les deux vecteurs divisé par le produit des normes. La distance cosinus n'est pas une vraie distance métrique d'un point de vu mathématique

$$\cos(\widehat{XY}) = \frac{X \cdot Y}{\|X\| \times \|Y\|}$$

Contrairement à la distance cosinus standard utilisée sur de tels espaces vectoriels (**Axiome 2.1**), la distance choisie pour les vecteurs conceptuels est la distance angulaire qui se calcule à partir de la distance cosinus (**Axiome 2.2**).

Sont également définies deux opérations, la somme normée et le produit normé (**Axiome 2.3**).

Ainsi, TEERAPARBSEREE (2005) propose une méthode pour la création d'une base d'acceptions multilingues entre deux langues l_1 et l_2, qui nécessite de disposer d'un dictionnaire bilingue $Dict_{l_1 \to l_2}$ et d'une base de vecteurs conceptuels pour l_1 et l_2.

1. Associer un vecteur conceptuel $V(sens_i)$ issu d'une base de vecteurs de l_1 à chaque sens de chaque mot. Si les sens en question sont issus d'un dic-

Axiome 2.2 **Distance angulaire thématique** – $D_A(X,Y)$

Soient deux vecteurs, X, Y issus d'une base de vecteurs conceptuels. La distance angulaire est l'arc-cosinus de la similarité cosinus :

$$D_A(X,Y) = \arccos(\widehat{XY})$$

Du fait des propriétés d'$\arccos$, pour des vecteurs thématiquement proches, $D_A(X,Y) \leqslant \frac{\pi}{4}$ et pour des vecteurs ayant peu de point communs, $D_A(X,Y) \geqslant \frac{\pi}{4}$.

Axiome 2.3 **Opérations normées** – $X \oplus Y$ – $X \otimes Y$

Soient deux vecteurs, X, Y issus d'une base de vecteurs conceptuels. Nous pouvons définir deux opérations normées s'appliquant aux deux vecteurs. La somme normée s'exprime par :

$$V = X \oplus Y : \forall i, v_i = \frac{(x_i + y_i)}{\|V\|}$$

Le produit normé s'exprime par :

$$V = X \otimes Y : \forall i, v_i = \sqrt{(x_i y_i)}$$

Axiome 2.4 **Contextualisation faible** – $\Gamma(X,Y)$

Soit deux vecteurs, X, Y issus d'une base de vecteurs conceptuels.
La contextualisation faible est une opération définie sur les vecteurs conceptuels qui permet d'amplifier les composants similaires entre le deux vecteurs et d'atténuer les éléments divergents :

$$\Gamma(X,Y) = X \oplus_{\|\cdot\|} (X \otimes_{\|\cdot\|} Y)$$

tionnaire monolingue qui fournit des définitions alors le vecteur $V(sens_i)$ peut être construit grâce à une opération dite de contextualisation faible (Lafourcade et Guinand, 2005) entre le vecteur moyen formé par les vecteurs de chaque mot $V_g(sens_i)$ de la définition du sens et le vecteur du mot du sens $V(mot)$ (**Axiome 2.4**), ce qui permet de construire à l'échelle d'une sens un vecteur contextualisé qui correspond à une acception particulière et donc d'un sens du mot.

2. Créer une acception interlingue associée à chaque acception monolingue de la langue source.
3. Relier les acceptions de la langue source correspondant à chaque sens de chaque mot de la langue aux sens correspondants dans la langue cible. Ainsi pour chaque mot mot_x dans le dictionnaire bilingue et chaque sous-entrée qui correspond à des distinctions de catégories grammaticales POS et à des traductions différentes $equivalents_{l_2}$ dans la langue cible $mot_{xi} = (POS, glose, equivalent_{l_2})$ où $glose$ est un ensemble de mots décrivant le signifié de la sous entrée :
   a) Relier les vecteurs de sens de la base de vecteurs de l_1 avec les sous-entrées mot_{xi}. Dans le dictionnaire bilingue il n'y a pas de distinctions

de sens autres que celles crées par des traductions divergentes et il est ainsi nécessaire de relier les vecteurs de sens de la base de vecteurs de l_1 avec les sous entrées mot_{xi}. Il faut d'abord prendre le vecteur V_g, qui est la moyenne des vecteurs conceptuels de chaque mot de la glose et calculer la contextualisation faible avec le vecteur $V(mot_x)$ de mot_x : $V(mot_{xi}) = \Gamma(V_g, V_{mot_x})$. Enfin, on compare la distance angulaire D_A de chaque $V(sens_i)$ de mot_x avec le vecteur du mot $V(mot_x)$. On sélectionne ensuite le sens du mot source dont le vecteur est le plus proche. Si la distance est supérieure à $\frac{\pi}{4}$ on passe à la sous-entrée suivante, sinon, on relie $sens_i$ à la sous entrée mot_{xi} et on continue avec l'étape suivante.

b) On calcule la distance entre $V(mot_{xi})$ et le vecteur $V(sens_{yi})$ de chaque sens de chaque traduction possible de mot_{xi} dans $equivalent_{l_2}$. On sélectionne le sens dont le vecteur $V(sens_{yi})$ est le plus proche de $V(mot_{xi})$. Si la distance est supérieure à $\frac{\pi}{4}$ on passe au mot_{xi} suivant, sinon on relie le sens ($sens_{yi}$ sélectionné à l'acception associée au sens $V(sens_i)$ de la langue source calculée à durant l'étape précédente.

Cette méthode pose deux problèmes, d'une part, lors de l'étape 3.c, on calcule la distance entre des vecteurs issus de la base de la langue l_1 avec des vecteurs issus de la base de la langue l_2. Or, les espaces vectoriels ne sont pas comparables car les dimensions ne représentent pas la même chose puisque aucune information lexicale ne fait le lien entre les acceptions. Les dimensions des vecteurs conceptuels représentent une hiérarchie de concepts issus de thésaurus et Teeraparbseree (2005) fait la supposition que les vecteurs sont comparables du moment que les hiérarchies de concepts aient une correspondance ordonnée exacte. Or, cette hypothèse ne peut jamais être réalisée à moins que les deux langues soit extrêmement proche, si proche en réalité pour être des dialectes mutuellement intelligibles. Si l'on appliquait cette méthode à des langues éloignée, la distance entre les vecteurs n'aurait absolument aucun sens.

Pour utiliser cette méthode il faudrait supposer d'avoir une mesure de distance translingue qui change les espaces des vecteurs afin de pouvoir les rendre comparables, ce qui est en réalité la difficulté principale dans l'alignement de ressources dans différentes langues, en particulier lorsque l'on veut utiliser les informations présentes dans les définitions.

2.4.2 Approches pour l'évaluation

L'évaluation de la qualité est une part importante de la construction d'une RLS multilingue et plus particulièrement au développement d'une méthode de construction et fiable.

2.4.2.1 Critères d'évaluation

Teeraparbseree (2005) fais la distinction entre trois critères d'évaluation, nous les décrirons brièvement, et apporterons par la même occasion une critique des critère vis-à-vis de leur désavantages et des limitations de leur application, en particulier au vu du contexte actuel.

COMPARAISON À UNE RÉFÉRENCE. La comparaison à une référence consiste à prendre une RLS multilingue déjà alignée avec des acceptions interlingues de ma-

nière manuelle et à comparer la ressource produite automatiquement afin d'évaluer la correspondance. L'objectif de la construction automatique est justement de ne pas avoir à construire une ressource manuellement du fait de la difficulté et du coût associé, il semble ainsi improbable d'utiliser une telle évaluation à grande échelle. Il faudra ainsi réserver ce critère à une petite échelle sur des acceptions individuelles pour vérifier la viabilité générale des méthodes appliqués, sans que ça ne constitue une évaluation représentative de l'ensemble de la ressource, comme nous le verrons dans la Section 4.3.

QUALITÉ STRUCTURALE. TEERAPARBSEREE (2005) définit la qualité structurale par rapport à l'état des connexions entre les acceptions interlingues et les sens qui les composent. Ainsi sont définies trois états dans lesquels l'alignement d'un sens monolingue peut se trouver :

— Incomplet : Un sens n'est connecté à aucune acception interlingue.
— Correct et complet : Un sens est connecté a une et unique acception multilingue.
— Incorrect : Un sens est connecté à plusieurs acceptions interlingues.

Ce critère est en effet indicatif d'un certain nombre de problèmes de structuration et peut être utile dans le contexte d'une construction manuelle. Pour une construction automatique, les cas d'incomplétude et d'incorrection peuvent être détectés et corrigés, potentiellement par une autre approche qui n'est pas susceptible de reproduire ces erreurs. Il est dans ce cas préférable d'incorporer un tel critère comme une contrainte lors de la construction afin d'éviter que ces erreurs ne soient commises.

JUGEMENT HUMAIN. Un critère basé sur le jugement humain consisterait à faire corriger à des experts (linguistes, traducteurs, etc.) une partie de la ressource et de compter le nombre d'erreurs qui ont été corrigées par rapport à la taille de la ressource. Du fait de la difficulté et du coût nécessaire pour faire une telle correction à l'échelle de toute la ressource, TEERAPARBSEREE (2005) propose de réaliser un échantillonage représentatif de la ressource en fonction du budget en temps et financier.

Un autre problème qui se pose est bien entendu le fait que chaque expert ne peut effectuer qu'une correction partielle, à moins qu'il ne parle toutes les langues présentes dans la base. Il est essentiel d'avoir un moyen de sélectionner et de ne montrer à chaque expert que des éléments qu'il est habilité à corriger, ce qui est une tâche assez difficile.

INFORMATION SÉMANTIQUE. TEERAPARBSEREE (2005) présente ce critère comme consistant à utiliser des vecteurs conceptuels pour s'assurer de la proximité effective des sens qui sont reliés à une acception d'une manière similaire à la construction par vecteur conceptuel. Cependant, si l'on utilise la même base de vecteurs pour la construction et pour l'évaluation de la ressource, on obtiendra une mesure de qualité très haute, sans que cela ne corresponde à la réalité, car les vecteurs ne sont qu'une approximation de l'information sémantique réelle et ne sont pas eux même d'une qualité parfaite. Si les vecteurs ne sont pas comparables, alors une évaluation avec ce critère risque de ne pas du tout être représentative de la réalité.

2.4.2.2 Mesures d'évaluation

TEERAPARBSEREE (2005) conçoit l'évaluation comme une combinaison linéaire de mesures de qualité, afin de proposer un critère global, qui est ainsi une somme pondérée de k différentes mesures :

$$Q = \sum_{i=1}^{k} = p_i \cdot Q_i$$

La valeur de Q pour une ressource sans erreur est zéro. Cette valeur est 1 pour une ressource complètement incorrecte. Il doit en être de même pour l'ensemble des mesures Q_i. Les poids p_i sont présentés comme étant à définir par des linguistes selon leurs besoins spécifiques.

Nous allons maintenant brièvement présenter et critiquer chacune des mesures proposées.

MESURE DE QUALITÉ PAR RAPPORT À UNE RÉFÉRENCE Si l'on considère une référence comme étant un ensemble d'acceptions notée A_{ref} et si l'on note les acceptions produites comme un ensemble A_{algo} nous pouvons définir quatre valeurs caractérisant les éléments corrects ou incorrects à l'instar d'une évaluation par étalon de référence :

- Vrais Positifs – $VP = A_{ref} \cap A_{algo}$
- Vrais Négatifs – $VN = \{tn_i | tn_i \notin A_{ref} \wedge tn_i \notin A_{algo}\}$
- Faux Négatifs – $FN = \{tn_i | tn_i \in A_{ref} \wedge tn_i \notin A_{algo}\}$
- Faux Positifs – $FP = \{tn_i | tn_i \notin A_{ref} \wedge tn_i \in A_{algo}\}$

Pour qu'une acception soit considérée identique à une autre il faut que les sens monolingues qui la composent soient les mêmes. On peut ainsi définir des mesures standards telles que le Rappel, la Précision, le Silence et le Bruit :

- Rappel – $R = \frac{VP}{VP+FN}$
- Précision – $P = \frac{VP}{VP+FP}$
- Silence – $S = \frac{FN}{FN+VP}$
- Bruit – $B = \frac{FP}{FP+VP}$

Contrairement à ce qui est présenté par TEERAPARBSEREE (2005), pour obtenir une mesure où une plus petite valeur correspond à une meilleure qualité, il faut inverser la précision et le rappel et non le silence et le bruit. Ainsi, $Q_i(P) = \frac{1}{P}$, $Q_i(R) = \frac{1}{R}$, $Q_i(S) = S$, $Q_i(B) = B$.

MESURE DE QUALITÉ STRUCTURALE. La mesure de qualité structurale se base sur le critère du même nom présenté auparavant, c'est-à-dire sur le nombre de sens monolingues dans trois états : incomplets, incorrects, corrects et complets. On définira ainsi trois mesures correspondantes, où $|\ sens\ |$ est le nombre de sens monolingues dans la base et où $nbaccep(sens)$ donne le nombre d'acceptions dont le sens fait partie :

- $Q_i(sens_{incomplets}) = \frac{|\{sens_i | nbaccep(sens_i)=0\}|}{|sens|}$
- $Q_i(sens_{incorrects}) = \frac{|\{sens_i | nbaccep(sens_i)>1\}|}{|sens|}$
- $Q_i(sens_{corr.comp.}) = \frac{|sens|}{|\{sens_i | nbaccep(sens_i)=1\}|}$

MESURE DE QUALITÉ VECTORIELLE. La mesure vectorielle est définie à l'échelle de la base comme étant la distance moyenne entre les sens monolingues composant une acception multilingue où n désigne le nombre d'acceptions dans la base

et où $nbps_i$ désigne le nombre de paires de sens associées à une acception et où $distance_k$ est la distance entre les sens constituant la paire k :

$$Q_i(vec) = \frac{1}{\frac{1}{n}\sum_{i=1}^{n}(\frac{1}{nbps_i}\sum_{k=1}^{nbps_i} distance_k)}$$

Cette mesure pose exactement le même problème que la construction par vecteur, le fait que les vecteurs ne sont pas comparables entre différentes langues et que l'hypothèse fait ne peut jamais être vraie dans le contexte d'application.

Conclusions

Nous avons en premier lieu défini la tâche d'Alignement Automatique des Sens(AAS) et présenté les tâches similaires dans d'autres domaines ainsi que les approches existantes d'AAS à la fois pour des ressources dans une même langue mais aussi pour des ressources dans des langues différentes. Nous avons constaté les limitations de ces approches, c'est-à-dire la limitation à deux langues ou l'utilisation d'un pivot en langue naturelle, ce qui nous a mené à illustrer les problèmes posés par ce type de pivot.

Ceci nous à conduit à présenter le formalisme des acceptions interlingues, qui semble en théorie offrir une solution à ce problème, mais qui à le désavantage d'être très difficile à créer manuellement et à évaluer. Nus nous intéressons ainsi à une construction automatique de cette représentation interlingue. Ainsi, nous passons en revue le travail existant sur la création d'acceptions et en soulignons les problèmes, qui se révèlent être très similaires aux problèmes qui se posent dans le contexte plus général des algorithmes d'AAS.

Cependant, nous notons tout de même que l'apparition de véritables plongements sémantiques pourrait être un outil permettant d'utiliser certains des algorithmes proposés par Teeraparbseree (2005) et de les étendre afin de réellement adresser les problèmes de perte de contraste, qui sont l'un des verrous scientifiques majeurs, et qui ne soit pourtant pas traité et étudié dans le contexte de l'alignement de sens.

La première étape sur le long chemin de cette étude nécessite cependant de fournir une définition plus formelle de ce qu'est une acception ainsi que les contraintes mathématiques qui y sont associées, ce qui est le sujet du chapitre qui suit.

Deuxième partie

ALIGNEMENT DE RESSOURCES VIA DES ACCEPTIONS INTERLINGUES

Chapitre 3

Algorithmes et axiomatisation pour l'alignement par acceptions

Sommaire

Introduction

Avant de s'atteler à la construction automatique d'acceptions, il conviendra en premier lieu de formaliser les différents éléments des RLS à aligner, et aussi ce que sont exactement les acceptions interlingues et les relations de raffinement, qui n'ont été définies que de manière informelle par le passé, comme nous avons pu le voir dans la Section 2.3 du Chapitre 3.

Si nous n'avons pas déjà une base d'acceptions avec des sens que nous savons équivalents, c'est-à-dire pour la construction initiale d'une base d'acceptions, il est nécessaire de savoir quels sens sont équivalents, tout en distinguant les divergences de lexicalisation.

Ces dernières peuvent apparaître entre toutes les paires de langues que nous cherchons à aligner, ce qui implique qu'il faudra absolument d'abord évaluer *tous*

les alignements de sens bilingues deux à deux avant de pouvoir déterminer les équivalences multilingues initiales.

Comme présenté dans le Chapitre 2, il existe de nombreuses méthodes d'alignement automatique de sens pour le cas bilingue, et comme le processus initial de construction d'acceptions a besoin des alignements bilingues, il est inutile de réinventer de nouveaux algorithmes d'AAS. Nous pourrons donc faire l'hypothèse que l'ensemble des alignements doit exister en amont de la construction initiale des hiérarchies d'acceptions (Figure 3.1).

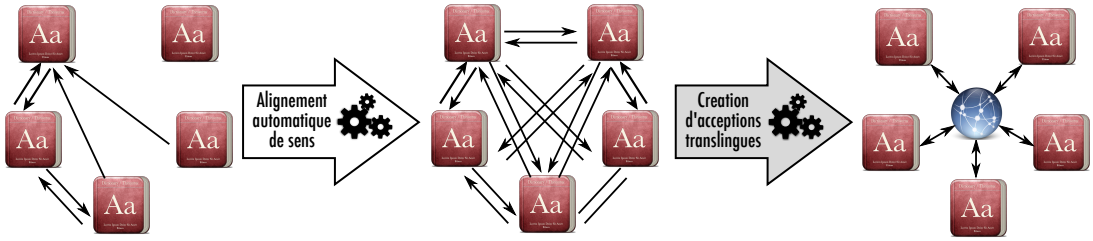


FIGURE 3.1 – Processus général de création des ressources lexico-sémantiques multilingues – création d'acceptions.

C'est une fois que la construction initiale sera terminée que nous devrons proposer des techniques de mise à jour des hiérarchies construites (suppression ou ajout de sens), qui dans le cas de l'ajout devront employer une ou des technique(s) d'alignement direct de sens aux hiérarchies qu'il conviendra de proposer.

Ainsi dans ce chapitre nous formaliserons dans un premier temps la structure des RLS ainsi que les acceptions, les relations de raffinement et les hiérarchies d'acceptions. Nous proposerons ensuite deux algorithmes de construction initiale d'acceptions sur la base du calcul des cliques dans le graphe des alignements bilingues et du degré des nœuds du graphe.

Le premier est un algorithme récursif naïf dans le sens où il correspond à la compréhension intuitive du problème, mais qui a une complexité prohibitive en temps de calcul dans le pire des cas ($n^n \cdot n!$). C'est pourquoi nous proposons ensuite un deuxième algorithme, itératif cette fois, qui exploite les propriétés de dégénérescence du graphe afin d'arriver au même résultat mais avec une complexité cette fois dominée par le calcul des cliques ($n^2 \cdot 3^n$).

Ensuite, nous présenterons les techniques d'ajout et de suppression de sens aux hiérarchies d'acceptions. Alors que la suppression s'avère assez simple, l'ajout est loin d'être trivial, en particulier si nous souhaitons éviter de calculer l'ensemble des alignements bilingues, ce qui, rappelons-le, est l'un des avantages évoqués pour les architectures à base de pivot interlingue. C'est pourquoi nous présenterons d'abord une technique d'ajout à base de noyau qui établit un squelette minimal de la hiérarchie en fonction des ressources disponibles.

Nous proposerons ensuite une ébauche de méthode d'alignement direct par calcul d'une divergence entre le sens à aligner et la hiérarchie d'acceptions, en définissant les propriétés et contraintes formelles régissant la construction de la mesure de divergence. Enfin nous ferons quelques observations sur le maintien de la validité axiomatique par les différents algorithmes proposés.

Nous conclurons finalement sur l'impact de la formalisation proposée et sur la portée potentielle des techniques. Nous ouvrirons ensuite l'horizon vers une

première expérimentation empirique sur les prospects de l'application des techniques proposées sur DBNary.

3.1 Formalisation mathématique des acceptions interlingues

Avant de formaliser la notion d'acception en tant que "classe d'équivalence" entre des sens provenant de différentes ressources lexico-sémantiques, il convient de présenter une structure formelle définissant tous les éléments d'une ressource lexico-sémantique ainsi que leur hiérarchie en fonction de la définition du format Ontolex, et dont la structure abstraite est suffisante pour servir de base à notre axiomatisation de la construction d'acceptions.

Ensuite, nous définirons les acceptions interlingues (axies), les relations de raffinement entre axies et les hiérarchies d'acceptions, puis nous inférerons certaines propriétés et théorèmes utiles à la fois pour la construction d'acceptions, mais aussi pour la vérification et le maintien de la validité formelle d'une base d'acceptions.

3.1.1 Une vue formelle d'une ressource lexico-sémantique

Le lexique d'une langue (**Définition 3.1**) est composé de vocables/mots-vedette (**Définition 3.2**).

Définition 3.1 **Lexique (formel) – $\mathcal{L}$**

Soit L l'ensemble de toutes les langues.
Un lexique $\mathcal{L}$ dans une langue donnée $l \in L$ est l'ensemble des ses vocables/mots-vedette[a] $\mathcal{V}_{\mathcal{L}_l}$.

a. Voir **Définition 3.2**

Définition 3.2 **Vocable/Mot-vedette (formel) – $v_i \in \mathcal{V}_{\mathcal{L}_l}$**

Un mot-vedette $v_i \in \mathcal{V}_{\mathcal{L}_l}$ représente un mot de la langue $l \in L$ qui sert de point d'entrée dans la ressource. À tout mot-vedette est associée une chaîne de caractères contenant le mot et identifiant le mot-vedette de manière unique.
Soit $\mathcal{C}hs$ l'ensemble de toutes les chaînes de caractères. La fonction $\texttt{mot} : \mathcal{V}_{\mathcal{L}_l} \mapsto \mathcal{P}(\mathcal{C}hs)$ retourne le lemme associé au mot-vedette.
À tout mot-vedette est associé un ensemble d'entrées lexicales $\mathcal{E}l$. La fonction suivante renvoie l'ensemble des entrées lexicales[a] associées au mot-vedette : $\texttt{EntrLex} : \mathcal{V}_{\mathcal{L}_l} \to \mathcal{P}(\mathcal{E}l)$.

a. Voir **Définition 3.3**

Chaque vocable se compose d'entrée lexicales qui correspondent au mot représenté par le vocable, auquel est ajoutée une distinction étymologique et de partie du discours (**Définition** 3.3).

Définition 3.3 **Entrée lexicale (formel) –** $el \in \mathcal{E}l$

Une entrée lexicale $el \in \mathcal{E}l$ est associée à un mot-vedette et correspond au mot dans une étymologie et dans une partie du discours particulières.
Une fonction $\mathtt{Pos} : \mathcal{E}l \to \mathcal{C}hs$ associe à une entrée lexicale une étiquette de partie du discours. Une fonction $\mathtt{Etim} : el \mapsto \mathbb{N}$ y associe un nombre identifiant son origine étymologique.
À une entrée lexicale el est associé un ensemble de sens de mots par la fonction[a] : $\mathtt{Sens} : \mathcal{E}l \to \mathcal{P}(\mathcal{S})$.

a. Voir **Définition** 3.4

Chaque entrée lexicale possède un certain nombre de sens (**Définition** 3.4), qui correspondent aux différents usages du mot en contexte[1].

Définition 3.4 **Sens (lexie)/Acception monolingue (formel) –** $s \in \mathcal{S}$

Un sens (lexie) $s \in \mathcal{S}$ est une unité atomique d'une RLS qui pointe vers l'information sémantique d'un mot utilisé dans une acception particulière dans le discours (signifié).
Le signifié est porté par une signature sémantique $sg \in \mathcal{S}g$, qui peut contenir différents types d'information (par exemple des définitions textuelles ou encore des exemples illustrant le ou les contextes d'usage)[a].
Nous pouvons ainsi définir la fonction $\mathtt{signf} : \mathcal{S} \to \mathcal{S}g$, qui associe à chaque sens sa signature sémantique.
L'ensemble $\mathcal{E}nt = \mathcal{E}l \cup \mathcal{S}$ se nomme l'ensemble des entités lexicales. Pour tout entité lexicale, la fonction $\mathtt{langue} : \mathcal{E}nt \mapsto L$ donne la langue L à laquelle correspond le vocable ou l'entrée lexicale englobant cette entité. Ainsi, nous définissons l'ensemble $\mathcal{E}nt_{\mathrm{L}} = \{ent \mid \mathtt{language}(ent) = \mathrm{L}\}$ de toutes les entités d'une langue L et par extension, les ensembles $\mathcal{E}l_{\mathrm{L}}$ et $\mathcal{S}_{\mathrm{L}}$ qui correspondent respectivement aux entrées lexicales et aux sens dans cette langue L.

a. Voir **Définition** 3.5

Chaque sens possède un ensemble d'attributs qui portent son information sémantique (composée de définitions, d'exemples d'usage, des "plongements sémantique"[2], etc. – **Définition** 3.5), et qui se nomment habituellement signatures sémantiques.

Il peut y avoir des relations entre les entrées lexicales et/ou les sens de mots (l'union des deux est l'ensemble des entités lexicales) ainsi que les vocables. Chaque relation porte un type propre (synonymie, antonymie, etc.), et est limitée aux entités appartenant à une même ressource/langue. Comme l'indique la **Définition** 3.6, chaque relation lexico-sémantique a des propriétés formelles très différentes, mais étant donné qu'elles n'interviennent pas directement dans la for-

1. Nous ne considère pas ici de notion d'homonymie car nous nous calquons sur le modèle de représentation de LMF pour cette formalisation.
2. *Semantic vector embedding* en anglais dans la littérature.

Définition 3.5 **Signature sémantique (formel) – $sg \in \mathcal{S}g$**

Soit $\mathcal{S}b = \{sb_i \mid sb_i \in \mathcal{C}hs\}$, l'ensemble des symboles sémantiques. Nous pouvons ainsi définir la signature sémantique d'un sens $s \in \mathcal{S}$ comme une union disjointe de sous-ensembles de symboles :

$$\exists sg_1, sg_2, \ldots, sg_n \in \mathcal{S}b^n : \mathtt{signf}(s) = \bigcup_{i=1}^{n} sg_i, \bigcap_{i=1}^{n} sg_i = \emptyset$$

Chaque sous-ensemble sg_i est associé à un type $tSg \in TypesSig$ par la fonction $TypesSig : E \subset \mathcal{S}g \mapsto \mathcal{S}b$, qui indique le type de la signature sémantique (définition, arcs dans un graphe, etc.).
Nous pouvons maintenant définir une fonction $\mathtt{signf}$ raffinée retournant le ou les sous-ensemble(s) de symboles de la signature sémantique d'un type particulier associé à un sens, où k est le nombre de sous-ensembles retournés :

$$\begin{array}{lll} \mathrm{signf}: & \mathcal{S} \times TypesSig & \rightarrow \mathcal{S}b^k \\ & (s, tSg) & \mapsto \mathrm{signf}(s, tSg) = \{sg_i \in \mathtt{sigf}(s) \mid TypesSig(sg_i) = tSg\} \end{array}$$

Cette vision de la signature sémantique correspond exactement à celle de M. T. Pilehvar et Roberto Navigli (2014), présentée dans la Figure 2.2 de la Section 2.1.2.3.

mation d'acceptions à partir de ressources alignées deux à deux, leur propriétés formelles ne seront pas étudiées ici.

Définition 3.6 **Relations lexico-sémantiques (formel) – $r_{ls} \in \mathcal{R}_{ls}$**

Une relation lexico-sémantique $r_{ls} : \mathcal{E}nt \mapsto \mathcal{E}nt$ met en relation deux entités lexicales. L'ensemble des relations lexico-sémantiques $\mathcal{R}_{ls}$ se compose de plusieurs types (ensemble ouvert) de relation qui ont chacune des propriétés formelles différentes. Les relations lexicales s'appliquent plutôt entre entrées lexicales alors que les relations strictement sémantiques ne s'appliquent qu'au niveau des sens.

3.1.2 Alignements, classes d'équivalences et relations de raffinement

Ainsi, la relation de traduction ou d'alignement (**Définition 3.7**) est dans le cadre général une relation de dépendance, qui n'est pas toujours transitive à cause des phénomènes contrastifs évoqués dans la Section 2.2.

Définition 3.7 **Relations de traduction/d'alignement (formel)–** $t \in \mathcal{T}r$

C'est une relation binaire $\mathcal{T}r : \mathcal{S}_{L_1} \times \mathcal{S}_{L_2}$ entre deux sens. La fonction qui pour chaque sens retourne l'ensemble de ces alignements/traductions est définie comme :

$$\begin{aligned} t : \mathcal{S} &\longrightarrow \mathcal{S} \\ s &\longmapsto \{s_i \in \mathcal{S} \mid s\mathcal{T}rs_i\} \end{aligned}$$

Nous définissons cette relation comme une relation de dépendance **réflexive, symétrique** et **non transitive**.

Si la relation d'alignement était toujours transitive, il serait toujours possible, à partir d'un ensemble d'alignements entre plusieurs paires de ressources de trouver un alignement interlingue en une classe d'équivalence unique (**Définition** 3.8).

Définition 3.8 **Relations et classes d'équivalence**

Une relation binaire $\sim$ est une relation d'équivalence sur un ensemble E si et seulement si elle est réflexive, symétrique et transitive.

Une classe d'équivalence sur un ensemble E est un ensemble $[a]$ contenant tous les éléments équivalents à l'élément a par une relation d'équivalence $\sim$ sur cet ensemble E : $[a] = \{x \in E \mid a \sim x\}$.

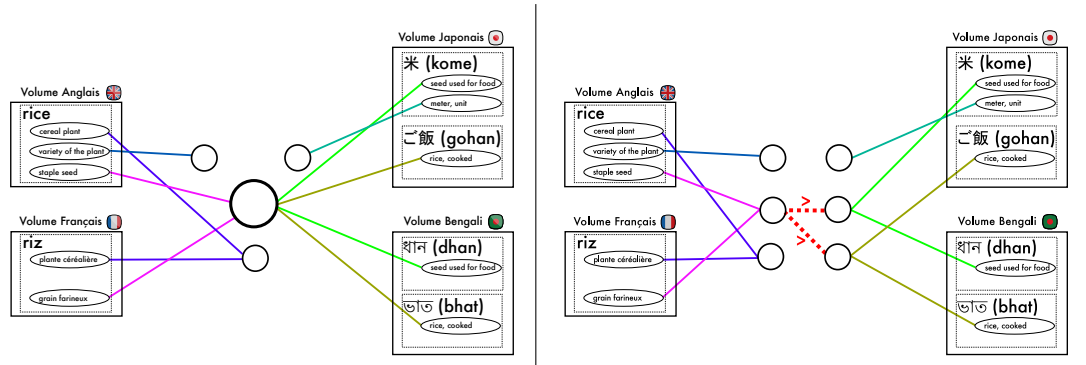
L'ensemble des classes d'équivalence forme une partition de E. Cet ensemble est appelé « quotient de E par R » et noté E/R.

Or, nous avons donné un exemple dans la Section 2.3 où cette hypothèse était fausse. La situation décrite par l'exemple est la motivation même d'utiliser des acceptions interlingues d'un point de vue linguistique pour pouvoir aligner les sens par une représentation interlingue non biaisée.

Reprenons donc les deux cas de figure de cette section, où d'une part tous les alignements sont considérés transitifs (Figure 3.2a) et où nous aurions une classe d'équivalence unique entre tous les sens mutuellement alignés deux à deux et où d'autre part on définit plusieurs classes d'équivalences reliées entre elles par une autre relation (dite de raffinement – Figure 3.2a) présentée dans la Section 2.3.

Avec cette représentation, on peut ainsi regrouper toutes les instances où la relation de traduction/d'alignement est transitive dans des classes d'équivalence et caractériser la non-transitivité due à un phénomène contrastif d'une manière explicite. Nous proposons donc une formalisation de la notion d'acception et de relation de raffinement que nous présentons dans la **Définition** 3.9 ci-contre.

Ainsi, d'après cette définition, on dit ainsi qu'une acception $ax_a \in \mathcal{A}x$ est plus générale qu'une $ax_b \in \mathcal{A}x$ si et seulement si $ax_a > ax_b$ et on sait qu'une acception $ax_a \in \mathcal{A}x$ est plus spécifique qu'une $ax_b \in \mathcal{A}x$ si et seulement si $ax_b > ax_a$.



(a) Relation d'alignement/traduction transitive

(b) Relation d'alignement/traduction non transitive avec raffinement

FIGURE 3.2 – Relation de traduction transitive menant à une classe d'équivalence unique contre relation de traduction partiellement transitive avec une relation de raffinement.

Définition 3.9 **Acceptions et raffinement (formel) –** $ax_a \in \mathcal{A}x > ax_b \in \mathcal{A}x$

La relation de raffinement est une relation binaire entre deux classes d'équivalence de sens. Si l'on note une classe d'équivalence de sens comme $ax = \{[s] \mid s \in \mathcal{S}\}$ et l'ensemble de toutes les classes d'équivalences de sens $\mathcal{A}x = \{[s_i] | s_i \in \mathcal{S}\}$ nous pouvons définir la relation de raffinement comme $> : \mathcal{A}x \times \mathcal{A}x$. Pour deux acceptions reliées par cette relation, le sens en partie gauche est plus général que le sens en partie droite, et celui de droite plus spécifique.

Nous noterons l'appartenance d'un sens à une acception par la relation

$$\begin{array}{lll} \in : & \mathcal{S} \longrightarrow & \mathcal{A}x \\ & s \longmapsto & ax = [s] \end{array}$$

La spécificité peut être à la fois de nature sémantique (lexicalisations divergentes) et de nature linguistique (divergence dans les catégories grammaticales) ; elle correspond au cas des contrastes identifiés dans la Section 2.2.

Intéressons-nous maintenant aux propriétés formelles de la relation d'une manière cohérente avec les descriptions informelles antérieures, par exemple celles Sérasset (1994).

Nous pouvons modéliser la relation de raffinement comme un ordre partiel strict, c'est-à-dire comme une relation asymétrique, antiréflexive et transitive. L'**Axiome 3.1** définit les propriétés formelles de la relation de raffinement et justifie les choix de chacune des propriétés.

Axiome 3.1 **Relation de raffinement – Ordre partiel strict**

La relation de raffinement est une relation d'ordre partiel strict (asymétrique, antiréflexive, transitive – (Davey et Priestley, 2002)) :

- *asymétrique.* Si une acception est strictement plus spécifique qu'une autre ($ax_g > ax_{s_i}$) alors cette acception ne peut être plus générale que ax_g qui est plus générale qu'elle : $ax_g > ax_{s_i} \Rightarrow \neg(ax_g > ax_{s_i}) \Rightarrow ax_{s_i} \not> ax_g$.
- *antiréflexive.* Une acception étant une classe d'équivalence, tous les éléments qu'elle contient sont équivalents, une acception ne peut donc porter une information sémantique plus générale ou plus spécifique qu'elle même : $\forall ax \in \mathcal{A}x : ax \not> ax$.
- *transitive.* Si une acception ax_a porte une information sémantique plus générale qu'une acception ax_b, qui elle-même est plus générale qu'une troisième acception ax_c, alors ax_a portera elle-même une information sémantique plus générale que ax_c. Ainsi, $\forall ax_a, ax_b, ax_c \in \mathcal{A}x : ax_a > ax_b \wedge ax_b > ax_c \Rightarrow ax_a > ax_c$.

Comme la relation de raffinement est un ordre partiel strict, un ensemble d'acceptions reliées les unes aux autres par des relations de raffinement formeront une hiérarchie sous la forme d'un treillis, comme nous le définissons dans la **Définition 3.10**.

Définition 3.10 **Hiérarchie d'acceptions – $ah \in \mathcal{AH}$**

Une hiérarchie d'acceptions $ah \in \mathcal{AH}$ est un ensemble d'acceptions $\mathcal{A}x$ muni de la relation de raffinement $>$, c'est-à-dire un ordre partiel strict $\langle \mathcal{A}x_a \subset \mathcal{A}x, > \rangle$ représentable par un treillis.

La structure de la hiérarchie d'acceptions est arborescente : toute acception de la hiérarchie sauf la racine admet un suprémum (acception plus générale). La hiérarchie est donc un sup-demi-treillis. Nous noterons $\top$ le sommet du treillis, qui sera le maximum global vis-à-vis de la relation $>$ tel que $\forall ax \in \mathcal{A}x_a, \top > ax$. Comme nous travaillons avec des ensembles finis, les treillis correspondants seront également finis, nous pourrons donc ajouter un minimum global $\bot$ tel que $\forall ax \in \mathcal{A}x_a, ax > \bot$.

À l'échelle d'une ressource, il y aura plusieurs hiérarchies d'acceptions disjointes que nous pouvons définir comme n'ayant pas d'acceptions en commun (**Axiome 3.2**).

Axiome 3.2 **Hiérarchies disjointes d'acceptions**

L'ensemble de toutes les hiérarchies d'acceptions $\mathcal{AH}$ est composé de sous ensembles disjoints vis-à-vis de la relation de raffinement, c'est-à-dire que deux hiérarchies sont disjointes quand elles ne contiennent pas deux acceptions qui sont reliées l'une à l'autre par une relation de raffinement ou qui sont égales.

$$ah_x \in \mathcal{AH}, ah_y \in ah : ah_x \neq ah_y \implies \nexists ax_x \in ah_x, \nexists ax_y \in ah_y : ax_x > ax_y \vee ax_y > ax_x \vee ax_x = ax_y.$$

Du fait de la structure d'une hiérarchie, nous pouvons également définir pour chaque hiérarchie disjointe une acception racine plus générale que toutes les autres acceptions de la hiérarchie et des acceptions feuilles, qui ne sont plus générales qu'aucune autre acception de la hiérarchie (**Définition 3.11**).

Définition 3.11 **Feuilles et racines d'une hiérarchie**

Chaque hiérarchie disjointe a une *racine* qui est l'acception la plus générale dans la hiérarchie, définie comme suit : $\forall ah \in \mathcal{AH}, \exists ax_\top \in ah : \forall ax_x \in \mathcal{A}x, ax_x \neq ax_\top, ax_\top > ax_x$.
Chaque hiérarchie contient des acceptions dites *feuille* qui ne peuvent pas être plus générales qu'une autre acception de la hiérarchie : $\forall ah \in \mathcal{AH}, \exists ax_\bot \in ah : \nexists ax \in ah : ax_\bot > ax$.

Nous définissons également des opérateurs d'appartenance à une hiérarchie pour une acception et un sens (**Définition** 3.12).

Définition 3.12 **Appartenance à une hiérarchie**

Nous définissions la relation d'appartenance d'une acception à une hiérarchie comme :

$$\begin{aligned} \in: \quad \mathcal{A}x &\longrightarrow & \mathcal{AH} &= \{\top\} \cup \{\bot\} \cup \mathcal{A}x \\ ax &\longmapsto & \exists \mathcal{A}x_x, \mathcal{A}x_y &\subseteq \mathcal{AH} : \\ & & \forall ax_{x_i} \in \mathcal{A}x_x, ax_{y_j} &\in \mathcal{A}x_j, \\ & & ax_{x_1} > \ldots > ax_{x_n} > ax > ax_{y_1} &> \ldots > ax_{y_n} \end{aligned}$$

Nous pouvons définir l'appartenance d'un sens à une hiérarchie comme :

$$\begin{aligned} \in: \quad \mathcal{S} &\longrightarrow & \mathcal{AH} \\ s &\longmapsto & \exists ax \in ah \in \mathcal{AH} : s \in ax \end{aligned}$$

La conjecture que nous avons faite dans ce chapitre est que nous travaillons sur des ressources déjà alignées deux à deux. Donc, à partir ce ces alignements, qu'il faudra *trouver* les acceptions, ce qui signifie que tous les sens appartenant à une même acception ou une même hiérarchie d'acceptions seront toujours liés les uns aux autres par des relations d'alignement.

D'après les définitions formelles de la relation de traduction et des acceptions, nous pouvons directement inférer la propriété PROPRIÉTÉ 3.1 qui dit que des sens faisant partie d'une même acception seront reliés par une relation de traduction.

PROPRIÉTÉ 3.1 $\forall s_1, s_2 \in \mathcal{S}, ax \in \mathcal{A}x : s_1 \in ax, s_2 \in ax \implies s_1 \sim s_2 \implies s_1 \mathcal{T} r s_2$.

Cette propriété implique la contraposée, qui stipule que deux sens qui ne sont pas reliés par une relation de traduction/alignement ne peuvent pas être reliés par une relation d'équivalence et donc ne peuvent pas faire partie de la même acception :

PROPRIÉTÉ 3.2 Contraposée de la PROPRIÉTÉ 3.1

$$\begin{aligned} & \neg(\forall s_1, s_2 \in \mathcal{S} : s_1 \sim s_2 \implies s_1 \mathcal{T} r s_2) \\ & \iff \\ & \exists s_1 \in \mathcal{S}, s_2 \in \mathcal{S} : s_1 \not{\mathcal{T}} r\, s_2 \implies s_1 \nsim s_2 \\ & \quad \implies \nexists ax \in \mathcal{A}x : s_1 \in ax \wedge s_2 \in ax \end{aligned}$$

(par la **Définition** 3.9)

Exprimons cela dans notre système axiomatique afin de le rattacher à la situation de départ que nous avons postulé.

Axiome 3.3 **Relation de traduction et hiérarchies d'acceptions**

Pour deux acceptions quelconques, si la première est plus générale que la deuxième, alors tous les sens équivalents reliés à la première acception, sont reliés par une relation de traduction/alignement avec tous les sens équivalents reliés à la deuxième acception.

$$\forall ax_1, ax_2 \in ah \in \mathcal{AH} \implies \forall s_i \in ax_1, \forall s_j \in ax_2 : s_i \mathcal{T} r s_j$$

3.1.3 Quelques propriétés et contraintes de validité

À partir des définitions et axiomes élémentaires portant sur les relations de traduction/alignement, les acceptions interlingues et les relations de raffinement, nous pouvons inférer des contraintes qui caractérisent des acceptions valides et qui seront utiles à la conception d'un algorithme permettant de construire les acceptions automatiquement.

Teeraparbseree (2005) et Sérasset (1994) décrivent de manière informelle une règle de validité pour les acceptions interlingues, qui consiste à dire qu'un sens donné ne peut appartenir à plus d'une acception. C'est un résultat que l'on peut prouver de manière formelle à partir des axiomes précédents en montrant que la relation d'appartenance d'un sens à une acception $\in : \mathcal{S} \to \mathcal{A}x$ est injective, ce qui constitue un théorème important (**Théorème 3.1**) pour le maintien de la cohérence dans une base d'acceptions.

Théorème 3.1 **Injectivité de la relation** $\in : \mathcal{S} \longrightarrow \mathcal{A}x$

Un sens ne peut faire partie de plus d'une acception à la fois.

Démonstration Supposons qu'un sens s soit lié à deux acceptions différentes : $\exists ax_a \, \exists ax_b : s_a \in ax_a, s_a \in ax_b, ax_a \neq ax_b$. Or, ax_a est une classe d'équivalence, donc $\forall s_i \in ax_a, s_i \sim s$, de même, ax_b est une classe d'équivalence et $\forall s_j \in ax_b, s_j \sim s$. Donc, $\forall s_i \in ax_a, \forall s_j \in ax_b, s_i \sim s_j \Leftrightarrow ax_a = ax_b$ par la transitivité de la relation d'équivalence, ce qui contredit le postulat de départ. Ainsi, $\nexists ax_a \, \nexists ax_b : ax_a \neq ax_b$ ■

D'une manière similaire, il est possible de montrer qu'une acception ne peut faire partie de plus d'une hiérarchies en prouvant que la relation $\in : \mathcal{A}x \longrightarrow \mathcal{AH}$ est surjective (**Théorème 3.2**).

Théorème 3.2 **Injectivité de la relation** $\in : \mathcal{A}x \longrightarrow \mathcal{AH}$

Une acception ne peut faire partie de plus d'une hiérarchie de sens.

Démonstration Par la **Définition 3.10**, deux hiérarchies différentes $ah_a, ah_b \in \mathcal{AH}$ sont disjointes, or si deux hiérarchies d'acceptions sont disjointes : $\nexists ax_a \in ah_a, \nexists ax_b \in ah_b : ax_a = ax_b$, ce qui implique qu'il n'existe pas d'acceptions égales dans deux hiérarchies différentes. ■

À partir du Théorème 3.1 et du Théorème 3.2 nous pourrons arriver à un théorème plus fort (Théorème 3.3) qui stipule qu'un sens donné ne peut faire partie de deux hiérarchies différentes d'acceptions.

Théorème 3.3 Un sens n'appartien qu'à une hiérarchie de sens

Un sens ne peut appartenir à deux hiérarchies de sens différentes. $\nexists ah_a, ah_b \in \mathcal{AH} : \exists s \in \mathcal{S}, s \in ah_a, s \in ah_b$

Démonstration Comme un sens ne peut pas faire partie de plus d'une acception (Théorème 3.1), si il existait $ah_a, ah_b \in \mathcal{AH}$, alors d'après la Définition 3.12 $\exists s \in \mathcal{S}, s \in ah_a, s \in ah_b \Leftrightarrow \exists ax : s \in ax \wedge ax \in ah_a \wedge ah_b$, ce qui contredirait le Théorème 3.2 qui stipule qu'une même acception ne peut faire partie de deux hiérarchies disjointes. ■

Ces trois théorèmes sont importants pour la construction d'acceptions, car une conséquence est qu'une hiérarchie d'acceptions donnée et toutes les relations de traductions qui la composent sont contenues dans un sous-graphe complètement déconnecté de tout autre sommet du super-graphe de la ressource (Définition 3.13) dans le graphe. Cela veut dire qu'il sera possible de travailler sur des sous-graphes de manière itérative en ayant comme garantie de ne pas influencer d'autres parties de la ressource (Voir Section 3.3).

Définition 3.13 Sous-graphe déconnecté

Un sous-graphe déconnecté dans un graphe est un sous-graphe dans lequel tous les sommets sont connectés entre eux par un chemin, mais ne sont reliés à aucun autre sommet dans le super-graphe (Hopcroft et Tarjan, 1973).

Une autre contrainte portant sur la validité de la structure des acceptions interlingues est le fait qu'une acception plus générale dans la hiérarchie sera reliée à au moins deux acceptions plus spécifiques du fait que la divergence sémantique ou linguistique, qui donne lieu à l'introduction de la relation de raffinement, impliquerait nécessairement qu'il y aurait au moins deux formes lexicales/linguistiques distinctes (Théorème 3.4).

Théorème 3.4 Facteur de branchement minimum d'une hiérarchie

Une acception en partie gauche d'une relation de raffinement est liée à au moins deux acceptions moins spécifiques (en partie droite)

$$|\{x_i \in \mathcal{A}x \mid ax \in \mathcal{A}x > x_i\}| \geqslant 2$$

Démonstration La relation de raffinement modélise les relations d'alignement non transitives (Définition 3.9).
Ainsi, si il existe un lien de raffinement entre deux acceptions, il modélise une divergence sémantique ou linguistique (Définition 2.1).
Cela implique qu'un sens (le plus général) dans une ressource correspond à deux sens différents (plus spécifiques mais distincts l'un de l'autre) dans une ou plusieurs des autres ressources alignées à la hiérarchie.
Donc, chacun des deux sens plus spécifiques appartiendra à une classe d'équivalence plus spécifique reliée à la classe d'équivalence à laquelle appartient le sens plus général. Ainsi une divergence se retranscrira toujours par au moins deux sous-acceptions. ■

Une dernière contrainte d'intégrité importante pour la cohérence d'une base d'acceptions est le fait que les sens d'une même entrée lexicale ne devraient jamais être alignés entre eux, puisqu'ils doivent normalement correspondre à des usages différents. Il s'agit d'une contrainte que nous considérerons comme un axiome pour la construction d'acceptions (**Axiome** 3.4).

Axiome 3.4 **Les sens d'une même entrée lexicale ne sont pas alignés**

Deux sens appartenant à la même entrée lexicale correspondent à des usages différents dont la distinction indique que les sens n'ont pas été considérés comme alignés ni traductions l'un de l'autre par le jugement humain.

$$el \in \mathcal{E}l, s_1, s_2 \in \mathcal{S}(el)^2 \implies s_1 \not{\mathcal{T}r} s_2 \implies s_1 \nsim s_2$$

Dans la pratique, dans une ressource avec une très petite granularité de distinction des sens, il se peut que l'algorithme de AAS utilisé, s'il ne maintient pas cette contrainte, aligne différents sens d'une même entrée lexicale ensemble. Nous considérons ici qu'il s'agit d'un problème en amont de la construction d'acceptions, qui doit être traité avant la construction des acceptions (par exemple en prenant la décision de fusionner de tels sens).

De cet axiome découle le **Théorème** 3.5 qui stipule que des sens d'une même entrée lexicale ne peuvent pas faire partie de la même acception ni de la même hiérarchie d'acceptions. Ainsi, les différents sens d'une même entrée feront partie de cliques différentes dans le graphe des alignements bilingues.

Théorème 3.5 **Disjonction des sens de la même entrée**

Deux sens de la même entrée lexicale ne peuvent appartenir à la même acception ($\star$) ou à la même hiérarchie d'acceptions ($\heartsuit$).

$$el \in \mathcal{E}l, s_1, s_2 \in \mathcal{S}(el), s_1 \neq s_2 \Rightarrow \nexists ax \in \mathcal{A}x : s_1 \in ax \wedge s_2 \in ax \qquad (\star)$$

Démonstration Par l'**Axiome** 3.4 $el \in \mathcal{E}l : s_1, s_2 \in \mathcal{S}(el), s_1 \neq s_2 \implies s_1 \nsim s_2$, ce qui implique qu'il n'y a pas de relation de traduction entre elles $s_1 \nsim s_2 \Rightarrow s_1 \not{\mathcal{T}r} s_2 \implies \nexists ax \in \mathcal{A}x : s_1 \in ax \wedge s_2 \in ax$ par définition d'une classe d'équivalence. ■

$$el \in \mathcal{E}l, s_1, s_2 \in \mathcal{S}(el), s_1 \neq s_2 \Rightarrow \nexists ax_a, ax_c, ax_k \in \mathcal{A}x : s_1 \in ax_a \wedge s_2 \in ax_c \wedge ax_a > \ldots > ax_k > \ldots > ax_c \qquad (\heartsuit)$$

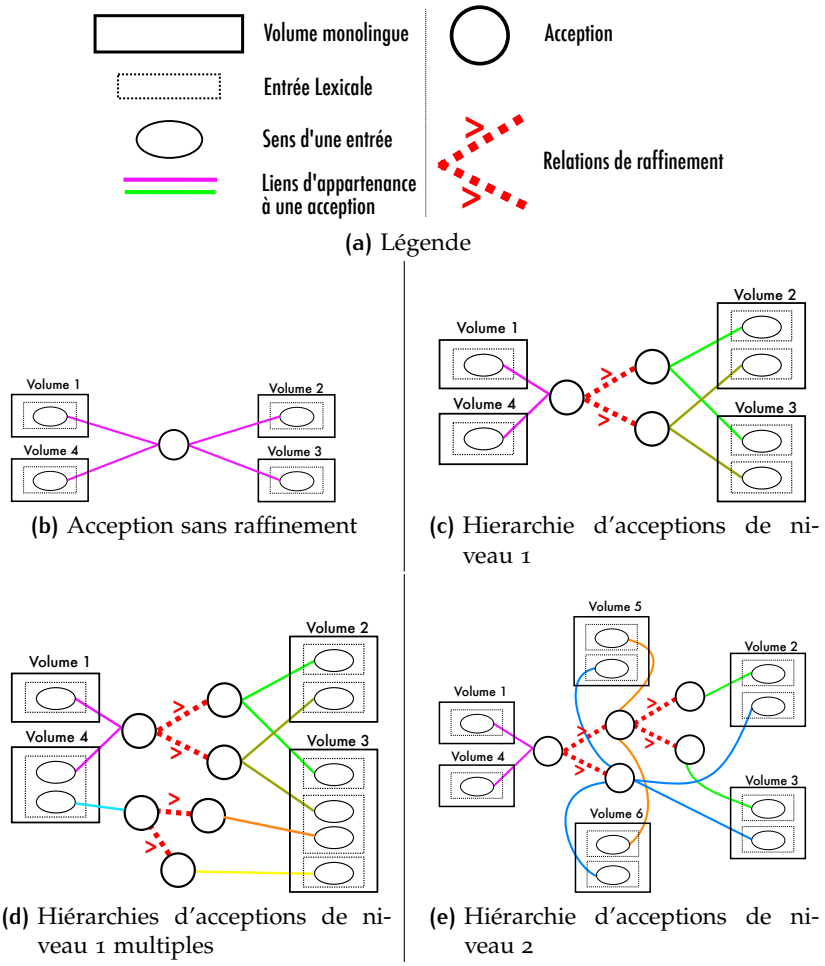
Démonstration Par l'**Axiome** 3.3, $s_1 \mathcal{T}r\, s_2$, or, comme par ailleurs $s_1, s_2 \in \mathcal{S}(el), s_1 \neq s_2$, ce qui d'après l'**Axiome** 3.4 implique que $s_1 \not{\mathcal{T}r} s_2$, ce qui est contradictoire. Ainsi $\heartsuit$ est vérifiée. ■

Maintenant que nous avons défini les contraintes de structuration et de validité formelle, nous allons explorer la relation entre la hiérarchie d'acceptions et les alignements bilingues initiaux, afin d'en déduire un algorithme de construction pour passer des alignements aux hiérarchies d'acceptions.

3.2 Autopsie d'acceptions anatomiquement correctes

Pour ce faire, nous allons considérer quelques exemples génériques de différents cas de figure dans la structuration des hiérarchies d'acceptions, et pour chacun des cas décrire le passage aux alignements bilingues (en faire l'autopsie, en quelque sorte) et faire différentes observations sur les éléments du graphe d'alignements bilingues qui caractérisent les acceptions et leur hiérarchie formés à partir de ce graphe.

Nous présenterons en premier lieu les cas de hiérarchies d'acceptions que nous prendrons comme exemple. Puis, nous les disséquerons et commenterons nos observations pour aller vers la conception d'un algorithme de construction.



(a) Légende

(b) Acception sans raffinement

(c) Hierarchie d'acceptions de niveau 1

(d) Hiérarchies d'acceptions de niveau 1 multiples

(e) Hiérarchie d'acceptions de niveau 2

FIGURE 3.3 – Acceptions anatomiquement correctes qui serviront de cobayes pour l'autopsie.

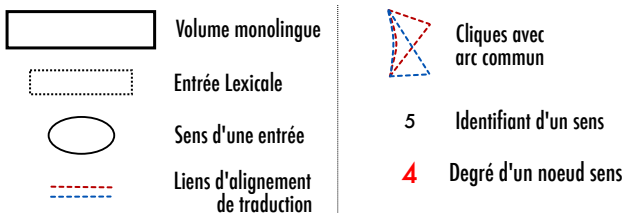
3.2.1 Profil des sujets

La Figure 3.3 présente l'ensemble des cas typiques que nous allons étudier. La Figure 3.3b présente une acception sans relation de raffinement (tous les sens sont équivalents). La Figure 3.3c présente une hiérarchie d'acceptions avec un niveau de raffinement. La Figure 3.3d présente deux hiérarchies d'acceptions avec un niveau de raffinement, elle illustre une conséquence du **Théorème** 3.5 qui énonce que deux sens de la même entrée lexicale ne peuvent pas faire partie de la même

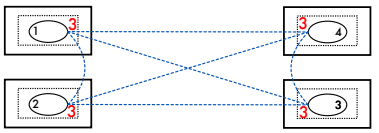
acception ou hiérarchie d'acceptions. La Figure 3.3e présente une hiérarchie d'acceptions avec deux niveaux de raffinement. Ici toutes les hiérarchies sont binaires, cependant il peut y avoir à chaque niveau, en théorie, un nombre arbitraire d'acceptions plus spécifiques.

3.2.2 Dissection des acceptions, observations et conjectures

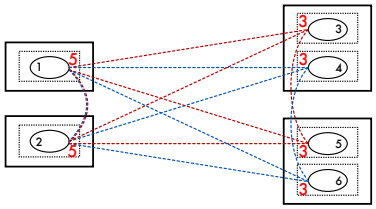
Nous allons maintenant «disséquer» les acceptions et revenir aux équivalences par paires auxquelles les acceptions correspondent, c'est-à-dire aux relations d'alignement bilingue entre les différents volumes. Nous pourrons ainsi déterminer si certaines structures et informations émergent, et si elles nous permettraient de retrouver les acceptions interlingues.



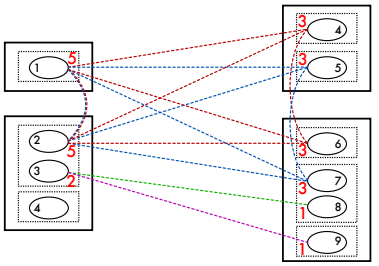
(a) Légende



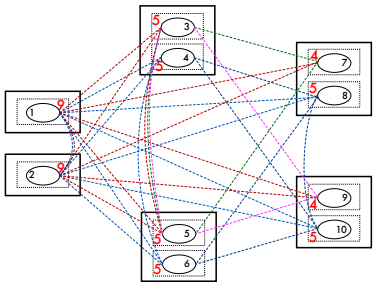
(b) Acception sans raffinement disséquée



(c) Hierarchie d'acceptions de niveau 1 disséquée



(d) Hiérarchies d'acceptions de niveau 1 multiples disséquées



(e) Hiérarchie d'acceptions de niveau 2 disséquée

FIGURE 3.4 – Acceptions anatomiquement correctes disséquées.

3.2.2.1 Hiérarchies élémentaires

De par les propriétés d'une classe d'équivalence, nous pouvons déjà déterminer que tous les sens faisant partie d'une classe d'équivalence doivent être alignés deux à deux, c'est-à-dire qu'ils doivent former un sous-graphe complet qui correspond à une clique (**Définition** 3.14).

SANS CONTRASTE. Nous pouvons observer un cas qui ne présente pas de contraste Figure 3.4b et dans lequel tous les sens sont équivalents car les relations d'aligne-

ment individuelles forment un graphe complet où tous les sens sont interconnectés (c'est une clique maximum).

Définition 3.14 Clique

Une **clique** est un sous-ensemble de sommets d'un graphe, qui forment un graphe dont tous les sommets sont adjacents deux à deux (un sous-graphe complet)[a].
Une **clique maximale** est une clique qui ne peut être étendue en lui ajoutant un sommet adjacent supplémentaire.
Une **clique maximum** est une clique de taille contenue dans le graphe.

a. (ERDÖS, 1966)

CONTRASTE DE PREMIER DEGRÉ. Dans la Figure 3.4c qui correspond à une hiérarchie d'acceptions de premier niveau, on peut constater qu'il y a deux cliques avec une partie commune. La partie commune est une sous-clique (ici un arc, 1–2, qui est une clique de deux sommets) qui regroupe les deux sens qui étaient dans l'acception générale. Chaque chemin dans la hiérarchie semble correspondre à chacune des cliques.

Ici, il est possible de reconstituer la hiérarchie des acceptions simplement en identifiant les deux cliques maximales. Les sommets de la partie commune constituent l'acception générale et tout le reste les acceptions plus spécifiques. Ces observations peuvent s'étendre à un nombre d'acceptions plus spécifiques supérieur à deux, mais nous n'en avons mis que deux dans la figure pour ne pas la surcharger. Pour un nombre arbitraire d'acceptions plus spécifiques au premier niveau, il y aura simplement autant de cliques divergentes d'un côté que de chemins au premier niveau de la hiérarchie d'acceptions.

Une autre observation que l'on peut faire est que le degré (**Définition 3.15**) des sens faisant partie de l'acception générale est le même et est strictement supérieur à celui des sens faisant partie des acceptions plus spécifiques, ce qui est dû au fait que les sens de l'acception générale sont en commun avec les deux cliques alors que les sens des acceptions plus spécifiques ne se trouvent que dans une seule clique à la fois.

Définition 3.15 Degré du nœud/sommet E d'un graphe – $deg(E)$

Le degré d'un nœud/sommet est le nombre d'autres sommets directement reliés à ce sommet par un arc[a].

a. (DIESTEL, 2005)

La présence de cliques différentes dans un graphe de traduction dans les cas de lexicalisations divergentes a été observée par MAUSAM et al. (2009), mais l'objectif de leur algorithme était uniquement de détecter les cas où la polysémie est problématique et nécessite de connaître les sens source d'où partent les relations de traduction, sans ce soucier de distinguer les sens des mots cibles des traductions, comme ici. Ainsi nous faisons nos observations à une granularité plus fine, d'une façon à pouvoir les appliquer aux acceptions.

PLUSIEURS SENS DE LA MÊME ENTRÉE. Si nous regardons maintenant du côté de la Figure 3.4d, nous voyons d'abord que les deux hiérarchies correspondent à des graphes complètement séparés. C'est une conséquence du **Théorème** 3.5, ce qui veut dire que chaque hiérarchie d'acceptions correspondra à un sous-graphe déconnecté différent.

La hiérarchie du bas est un cas particulier de celle du haut, puisque chaque acception est composée d'un sens unique. Ce que l'on observe est que chaque chemin de la hiérarchie est un seul arc entre deux sommets, le sommet composant le sens aligné à l'acception la plus générale et le sommet correspondant au sens de chaque acception plus spécifique. Le point commun est l'unique sommet correspondant au sens de l'acception générale. Les deux lignes correspondent tout de même à des cliques (de deux sommets) et le point commun entre les deux cliques est une clique de 1 sommet, ce qui ne déroge pas à l'observation pour la Figure 3.4c qui présente un cas plus général.

3.2.2.2 Hiérarchies généralisées

Examinons enfin la Figure 3.4e qui illustre une hiérarchie avec deux niveaux. Nous observons encore qu'il y a deux grandes cliques correspondant à la bifurcation du premier niveau de la hiérarchie, avec une partie commune qui correspond aux sens de l'acception la plus générale.

Pour l'acception plus spécifique du premier niveau en bas, tous les sens font partie d'une grande clique.

Pour l'acception du premier niveau en haut, qui est reliée à des acceptions plus spécifiques qu'elle au niveau deux, on constate que la partie commune des cliques couvre les niveaux zéro et un, mais qu'au niveau 2 il y a une divergence.

Il est toujours possible de constater que le degré des sens appartenant à des acceptions de plus en plus spécifiques est de plus en plus petit. Par ailleurs, les sens appartenant aux mêmes acceptions ont tous le même degré.

Comme une partie commune entre les cliques correspond aux acceptions plus générales communes entre les deux chemins de la hiérarchie, nous conjecturons qu'il est possible de résoudre le problème de la construction :

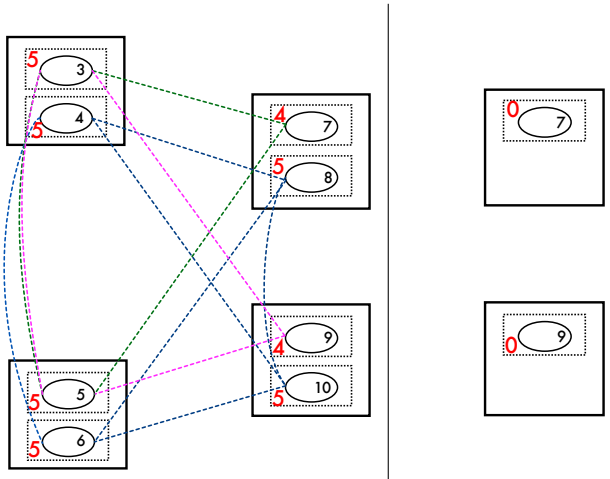
- en commençant à déterminer les points communs,
- puis en ôtant tous les points communs et en appliquant l'algorithme récursivement à chacun des sous-graphes obtenus, jusqu'à tomber sur une situation élémentaire (il ne reste plus de points communs entre les cliques).

DÉCOMPOSITION RÉCURSIVE. La Figure 3.4e montre ce qui se passe si l'on enlève les sens de l'acception la plus générale au niveau zéro dans la Figure 3.5a.

Nous obtenons deux graphes séparés qui correspondent chacun aux cliques avant la suppression des points communs. Le graphe bleu correspond exactement à la situation de la Figure 3.4c, c'est-à-dire à une clique unique où tous les degrés des sommets sont égaux.

Nous savons ainsi qu'il correspond à une acception unique qui ne sera liée à aucune autre acception plus spécifique par la relation de raffinement.

Le graphe composé des cliques verte et rose n'est pas un graphe complet et les sommets ont donc des degrés différents. Il faudra ainsi répéter la procédure qui a



(a) Hiérarchie d'acceptions de niveau deux où l'acception la plus générale et les sens qui en font partie ont été supprimés.

(b) Décomposition avec deux niveaux de récursion

FIGURE 3.5 – Décompositions récursives successives d'acceptions de la hiérarchie générale.

permis l'obtention de la Figure 3.4c en supprimant les sommets communs entre les cliques.

Nous obtiendrons ainsi de nouveau un graphe par clique, sauf qu'ici le graphe ne sera composé que d'un seul sommet (Figure 3.5b). Nous considérerons que chaque nœud est une clique d'un sommet et de degré zéro et correspond à une acception séparée.

Il est aisé de voir qu"on peut répéter ces étapes récursivement pour une profondeur de hiérarchie d'acceptions arbitraire, afin de retrouver les acceptions de départ dans toutes les configurations possibles d'alignements deux-à-deux.

3.3 Algorithme de construction initiale d'acceptions interlingues

À partir des intuitions et conjectures résultant de l'autopsie des acceptions, nous allons maintenant formuler des algorithmes pour la formation des acceptions interlingues, et dans un deuxième temps étudier la complexité dans le pire des cas de chacun des algorithmes.

Comme point de départ, nous avons un très grand graphe. La première question essentielle à se poser est : faut-il appliquer l'algorithme au graphe entier ? Ou alors faut-il plutôt travailler sur des sous-graphes de ce graphe ?

Une conséquence du **Théorème** 3.1, du **Théorème** 3.2 et du **Théorème** 3.3, comme nous l'avons explicité dans la Section 3.1.3 est que les sens qui feront partie d'une hiérarchie d'acceptions correspondront à un sous-graphe déconnecté du graphe d'alignement bilingue.

Plutôt que d'appliquer les algorithmes directement sur tout le graphe, nous pourrons donc les appliquer sur chacun des sous-graphes déconnectés. Il suffit donc de trouver tous les sous-graphes déconnectés, en utilisant des algorithmes standard d'exploration.

Nous pouvons exprimer cette approche de manière très simple d'un point de vue algorithmique (Algorithme 3.1). Un avantage d'avoir des sous-graphes déconnectés est qu'il sera relativement aisé d'effectuer le traitement d'une ressource de manière distribuée, avec la garantie qu'il n'y aura pas d'effets de bord.

Algorithme 3.1 – AcceptionsDistribuees

```
fonction AcceptionsDistribuees(G):
  composants = ComposantsConnectes(G)
  acceptions = ∅
  pour chaque composant dans composants:
    acceptions = acceptions ∪ ConstruireAcceptions(composant)
  retour acceptions
```

Sachant cependant que nous travaillons sur une ressource lexico-sémantique qui a une structuration supplémentaire en entrées lexicales, il est également possible de travailler entrée par entrée, en calculant pour chaque sens d'une entrée, par fermeture transitive, le sous-graphe déconnecté correspondant. La seule contrainte est de marquer les sens déjà couverts par un sous-graphe afin de ne pas les traiter plusieurs fois. C'est une segmentation bien plus logique, qui permet un meilleur suivi humain de ce qui est déjà fait et de ce qui reste à faire. L'Algorithme 3.2 donne le pseudo-code de cette approche.

Algorithme 3.2 – AcceptionsDistribueesParEntree

```
fonction AcceptionsDistribueesParEntree(RLS):
  acceptions = ∅
  senstraites = ∅
  pour chaque el dans ℰl ⊂RLS:
    pour chaque s ∉ sentraites dans 𝒮(el):
      composant = ParcoursProfondeurEnPremier(RLS, s)
      senstraites = senstraites ∪ composant
      acceptions = acceptions ∪ ConstruireAcceptions(composant)
  retour acceptions
```

3.3.1 Algorithme récursif

Algorithme 3.3 – AcceptionRec

```
fonction ConstruireAcceptions(G):
  retour  AcceptionsRec(G, ∅ , ∅ )

fonction AcceptionsRec(G, cliques , degres):
G = (V, E)
// V= sens
// E= alignements
  si cliques = ∅  et degres = ∅ :
    cliques = BronKerbosch( ∅ , ∅ ,V(G))
    degres = Degres(cliques[0])//Degres des sommets de la clique

  si |cliques| == 1: //Cas elementaire
      // Graphe avec une unique acception et sans arcs
      retour ({Acception(cliques[0])}, ∅ )
  sinon: // Recursion
    SC = SommetsCommuns(cliques)
    //Acception generale
```

```
18      ax_g = Acception(SC)
19      //Graphe de la hierarchie d'acceptions
20      G_AH = ({ax_g}, ∅)
21
22      pour chaque clique dans cliques:
23        si clique ≠ ∅:
24        //On enleve les sommets communs
25        clique = clique − clique ∩ SC
26        //Sous hierarchie d'acceptions (appel recursif)
27        G_AH_s = AcceptionsRec(SousGraphe(G, clique), cliques, degres)
28        //Ajout des acceptions (sommets) de la sous-hierarchie
29        V(G_AH) = V(G_AH) ∪ V(G_AH_s)
30        //Lien entre ax_g et le sommet de la sous-hierarchie
31        E(G_AH) = {(ax_g, V(G_AH_s)[0])} ∪ E(G_AH)
32        //Ajout du reste des relations
33        E(G_AH) = E(G_AH) ∪ E(G_AH_s)
34      retour G_AH
```

L'approche la plus simple (naïve) se base sur les observations de la Section 3.2.2, consistant à supprimer les points communs des cliques (acceptions plus générales) et à traiter les sous-graphes résultant récursivement, jusqu'à arriver à un cas élémentaire, puis à remonter en reconstruisant la hiérarchie d'acceptions.

3.3.1.1 L'algorithme

Le pseudocode de l'algorithme de construction récursive est donné dans l'Algorithme 3.3, que nous allons commenter de manière plus détaillée.

Comme nous l'avons illustré dans la section précédente, un cas élémentaire correspond à un sous-graphe complet, c'est-à-dire où tous les sens sont connectés les uns aux autres et où le degré de tous les sommets est le même, ce qui correspond au cas de figure de la Figure 3.4b et à la situation illustrée par la Figure 3.5. Ainsi, la première chose à faire sera de calculer l'ensemble de toutes les cliques (ligne 8) avec l'algorithme de Bron-Kerbosch (Algorithme 3.4) avec ordonnancement des arcs, qui est une approche standard pour calculer l'ensemble des cliques d'un graphe (Eppstein, Löffler et Strash, 2010).

Algorithme 3.4 – BronKerbosch

```
1
2   fonction BronKerbosch(G):
3     P=V(G)
4     R = X = ∅
5     cliques = ∅
6     pour chaque v dans V(OrdreDegresDecroissants(G)):
7       BronKerbosch2(R ∪ {v}, P ∩ N(v), X ∩ N(v), cliques)
8     retour cliques
9
10  fonction BronKerbosch2(R, X, V, cliques):
11    si |P| == 0 et |X| == 0:
12        cliques = cliques ∪ R
13    u = Pivot(P ∪ X)
14    pour chaque sommet v dans P \ N(u):
15        BronKerbosch2(R ∪ {v}, P ∩ N(v), X ∩ N(v), cliques)
16        P := P \ {v}
17        X := X ∪ {v}
```

Dans l'Algorithme 3.3, si le graphe G passé en argument correspond à un cas élémentaire, il ne contient qu'une clique unique où tous les degrés sont égaux (ligne 11). Dans cette situation, il nous suffit de retourner une hiérarchie d'acceptions contenant une unique acception correspondant aux sommets de la clique (ligne 13).

Dans le cas contraire, nous sommes dans un cas non élémentaire qu'il faudra traiter récursivement (ligne 14). Dans un premier temps, nous calculons les sommets en commun aux cliques (ligne 15) grâce à l'Algorithme 3.5. Les sommets communs formeront l'acception plus générale, car c'est une sous-partie du graphe dont tous les sommets sont reliés et ont le même degré.

Algorithme 3.5 – SommetsCommuns

```
1  fonction SommetsCommuns(G, cliques):
2    intersection = ∅
3
4    pour i dans 0..|cliques|−1:
5      clique = Tri(cliques[i]) // Tri par ordre naturel des
           sommets dans G
6      si i==0:
7        intersection = clique
8      sinon:
9        intersection = intersection ∩ cliques[i]
```

Ensuite (toujours dans l'Algorithme 3.3), pour chaque clique (ligne 21), nous enlevons les sommets en commun (ligne 23) pour pouvoir faire l'appel récursif (ligne 25). Comme l'algorithme nécessite un graphe en entrée et non seulement une liste de sommets, nous appelons AcceptionRec avec comme argument le sous-graphe correspondant à la clique.

L'appel récursif retourne un graphe de la sous-hiérarchie d'acceptions, donc les sommets sont ordonnés par degré décroissant, par construction, ce qui garantit que le premier sommet est le plus général, et qui facilite la liaison de l'acception générale ax_g correspondant aux sommets communs à la sous-hiérarchie.

Après le retour de l'appel récursif, nous ajoutons toutes les acceptions de la sous-hiérarchie $G_{\mathcal{AH}_s}$ à la hiérarchie résultat ($G_{\mathcal{AH}}$ (ligne 27), **après** ax_g (ce qui garantira l'ordre par généralité) et nous ajoutons ensuite un arc entre l'acception plus générale ax_g et la première acception de la sous-hiérarchie (ligne 29), qui sera aussi la plus générale de la sous-hiérarchie, tel que $ax_g > V(G_{\mathcal{AH}_s})[0]$. Nous finissons en ajoutant le reste des relations de la sous-hiérarchie (ligne 31). À la fin de l'itération, la hiérarchie est complète pour le niveau de récursion actuel et nous pouvons renvoyer la hiérarchie au niveau supérieur (ligne 33).

3.3.1.2 Complexité dans le pire des cas et performance à l'échelle

La complexité de l'algorithme a une grande importance, car il sera appliqué à des ressources avec potentiellement des millions d'entrées et des sous-graphes déconnectés potentiellement composés de centaines de sommets. Il est assez intuitif que cet algorithme récursif n'est sans doute pas optimal, car il effectue l'ensemble des opérations à chaque récursion. Ainsi, le nombre d'appels récursifs aura une grande importance dans le pire des cas.

Calculons maintenant ce nombre. Une acception doit au moins être constituée d'un sens pour exister. Si nous disposons de n sens de mots dans le sous-graphe, et si nous construisons une hiérarchie valide telle que chaque acception n'est associée qu'à un seul sens, alors les points communs ne seront qu'un unique sens,

et il faudra à chaque fois faire l'appel récursif avec un sous-graphe contenant un sens de moins. Ainsi, à chaque profondeur les appels récursifs travailleront avec $\frac{n-p}{k}$ nœuds où p est le nombre de sommets communs aux cliques et où k est le nombre de cliques maximales.

Il nous faut maintenant déterminer la complexité dans le pire des cas d'un appel récursif.

Tout d'abord les opérations d'intersection ensemblistes, que nous considérerons optimales ont une complexité de $O(n \cdot \log(n))$, car il faut d'abord trier les ensembles (Les implémentations des bibliothèques standard des langages de programmation majeurs incluent une implémentation d'un tri optimal).

Enfin. il faut comparer les éléments 1 à 1, ce qui dans le cas où les ensembles sont disjoints prendra $O(c)$, avec $c = \|S_1\| \times \|S_2\|$, le produit des cardinalités des ensembles.

D'autre part les opérations d'union implémentés avec duplication des données) auront besoin de parcourir tous les éléments des deux ensembles, soit $O(c)$.

Ensuite, détaillons la complexité de chaque sous-algorithme appelé dans l'Algorithme 3.3 :

- **Acception** : La fonction Acception crée un nouvel objet acception à partir d'un ensemble de sens, nous considérerons qu'elle opère en temps constant, soit en $O(1)$.
- **BronKerbosch** (Algorithme 3.4) : La variante de l'algorithme de Bron-Kerbosch utilisée ici (Eppstein, Löffler et Strash, 2010) a une complexité de $O(dn_p 3^{\frac{d}{3}})$, où d est le degré maximal dans le graphe.
- **Degres** : Le calcul des degrés d'un ensemble de sommets consiste, pour chaque sommet, à compter le nombre de voisins, dans le pire des cas, si il y a $\frac{n-p}{k}$ sommets dans l'ensemble, il y aura $\frac{n-p}{k} - 1$ voisins, et donc, la complexité de cet algorithme sera $O(\frac{n-p}{k} \cdot (\frac{n-p}{k} - 1)) = O(\frac{n-p}{k}^2)$.
- **SommetsCommuns** : (Algorithme 3.5) : SommetsCommuns est similaire à une intersection d'ensemble, successivement appliqué à toutes les cliques. En considérant que la somme des sommets de chaque clique est à peu près égale au nombre total de sommets, nous pouvons considérer que la complexité dans le pire des cas est $O(\frac{n-p}{k} \cdot \log(\frac{n-p}{k}))$.
- **SousGraphe** : SousGraphe retourne le sous graphe correspondant à une liste de sommets, ce qui consiste en deux opérations d'intersection, une sur les sommets et une sur les relations. Comme il y a au maximum $(\frac{n-p}{k} - 1)$ relations, alors la complexité sera $O(\frac{n-p}{k} + \frac{n-p}{k} - 1) = O(\frac{n-p}{k})$.

BronKerbosch n'est appelé qu'une fois au début de l'algorithme et ne comptera donc pas dans le calcul de la complexité de l'appel récursif. Ainsi, pour un appel récursif, la complexité dans le pire des cas nous donnera un total de $O(1 + \left(\frac{n-p}{k}\right)^2 + \frac{n-p}{k} \cdot \log(\frac{n-p}{k}) + \frac{n-p}{k}) = O(\left(\frac{n-p}{k}^2\right)$

Si on cherche à calculer la complexité totale, il faut prendre en compte le fait qu'à chaque niveau d'appel récursif il y a p nœuds en moins qu'il y a O(k) appels à chaque récursion (facteur de branchement k, un pour chaque clique). Dans le pire des cas les nombre de points communs et de 1 sommet, et donc à chaque niveau de récursion $p = p - 1$ avec $p_0 = n$:

$$O(\frac{k(n-1)^2}{k^2} \cdot \frac{k(n-2)^2}{k^2} \dots \frac{k(n-k)^2}{k^2})$$
$$= O(\frac{(n-1)(n-1) \cdot (n-2)(n-2) \dots (n-k)(n-k)}{k^k})$$
$$= O(\frac{(n-1)(n-2) \dots (n-k) \cdot (n-1)(n-2) \dots (n-k)}{k^k}) = O(\frac{\left(\frac{n!}{(n-k-1)!}\right)^2}{k^k})$$

Si le nombre de cliques k est proche de n et que $n!$ croit moins vite que n^n, la complexité serait sub-linéaire. Cependant si c'était le cas dans un graphe de sens, il n'y aurait que des nœuds déconnectés qui formeraient des cliques composées d'un seul nœud, ce qui par construction des graphes en question n'arriverait que si $n = 1$. Dans la plus part des cas k sera au moins de l'ordre de $\frac{n}{2}$, soit une complexité $O(\frac{\left(\frac{n!}{(n-\frac{n}{2}-1)!}\right)^2}{\frac{n}{2}^{\frac{n}{2}}})$ où le numérateur croit plus vite que le dénominateur (un tracé 2D de cette fonction indique une croissance super-exponentielle).

Les appels récursifs sont précédés de l'appel à BronKerbosch de complexité $O(dn \cdot 3^{\frac{d}{3}})$ soit au total $O(dn \cdot 3^{\frac{d}{3}} + (\frac{\left(\frac{n!}{(n-k-1)!}\right)^2}{k^k})$. Dans le cas typique discuté ci-dessus, la complexité du calcul récursif l'emporte largement sur celle du calcul de cliques.

3.3.2 Algorithme itératif

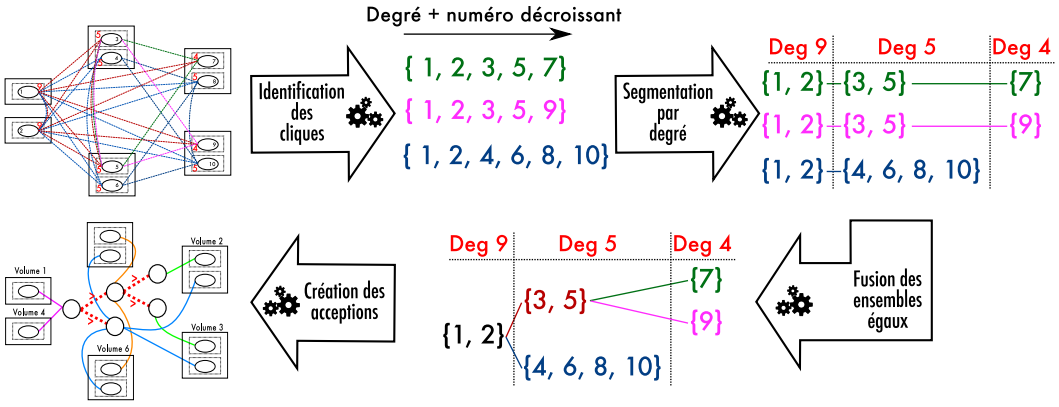


FIGURE 3.6 – Exemple d'application de l'approche itérative par décomposition dégénérescente.

Une propriété qui n'est pas explicitement exploitée dans l'algorithme récursif, est que les sens composant chaque acception au même niveau de la hiérarchie ont le même degré. Comme par ailleurs le premier appel vers BronKerbosch renvoie une clique par chemin dans la hiérarchie, il serait possible de segmenter chaque clique par degré, puis de regrouper les sous ensembles contenant les mêmes sens à degrés égaux. Par ce procédé nous pourrions obtenir les groupements des sens de chaque clique dans les bonnes acceptions, puis lier les acceptions en suivant les acceptions communes dans chaque chemin. Pour mieux visualiser le processus, nous proposons un exemple de l'application de cette idée sur une hiérarchie

généralisée de niveau deux (Figure 3.6), comme pour le reste des observations faites dans la Section 3.2.2.2.

3.3.2.1 *L'algorithme*

Initialement, l'algorithme (Algorithme 3.6) commence comme la version récursive, par le calcule des cliques (ligne 7) puis calcule les degrés sur l'ensemble des sommets du graphe (ligne 9).

Algorithme 3.6 – AcceptionsIter

```
1  fonction ConstruireAcceptions(G):
2    retour  AcceptionsIter(G)
3  fonction AcceptionsIter(G):
4    //V= sens, E= alignements
5    G = (V, E)
6
7    cliques = BronKerbosch(G)
8
9    degres = Degres(V(G))//Degres des sommets de la clique
10
11   TrierParDegre(degres,cliques) // Tri par degre decroissant
12
13   segmentation = SegmenterParDegre(cliques, degres)
14
15   //Graphe de la hierarchie d'acceptions
16   G_AH = ({ax_g}, ∅)
17
18   //Acceptions par degre
19   Ax_dg =HierarchieParCliqueEtDegre(G_AH, degres)
20
21   IdentifierDoublonsEtFusionner(Ax_dg,G_AH)
22
23   retour G_AH
```

Il suit ensuite les étapes décrites dans la Section 3.2.2.2. D'abord, les cliques sont triées par ordre de dégénérescence (ligne 11), puis sont segmentées par degré (ligne 13) par l'appel à l'Algorithme 3.7.

Algorithme 3.7 – SegmenterParDegre

```
1    fonction SegmenterParDegre(cliques, degres):
2      //clique: liste des sommets d'une clique, degres: degres de
         chaque sommet
3      segemntation = ∅×∅ //ensemble associatif clef/valeur
4      indicedg = 0
5      tant que indicedg < |degres|:
6        segment = ∅
7        si indicedg > 0 et degre[indicedg]!=degre[indicedg-1]:
8          segementation[degre[indicedg]] = {segment}
9          segment = ∅
10       segment = segment ∪ {clique[indicedg]}
11     retour segementation
```

Ensuite, nous créons une hiérarchie liant les ensembles successifs de chaque clique d'un degré à l'autre (lignes 19) par l'appel de l'Algorithme 3.8.

Algorithme 3.8 – HierarchieParCliqueEtDegre

```
1  HierarchieParCliqueEtDegre(G_AH, degres):
2    Ax_dg = ∅  //Acceptions par degre
3    pour chaque dgi dans 1..|degres|:
4        ax_prec = nil
5        pour chaque clq dans cliques:
6          //Une acception par clique et degre
7          ax = Acception(sgementation[degres[dgi]][clq])
8          Ax_dg[degres[dgi]] = Ax_dg[dg] ∪{ax}
9          V(G_AH) = V(G_AH) ∪{ax}
10         //Mise en relation d'acceptions degres successifs
11         si dgi > 0:
12           E(G_AH) = E(G_AH) ∪{(ax_prec,ax)}
13         ax_prec = ax
14   retour  Ax_dg
```

Ensuite nous identifions les sous ensembles (acceptions) par degrés qui sont égaux, pour finalement les fusionner (lignes 19) par l'Algorithme 3.9 et obtenir la hiérarchie finale d'acceptions (ligne 21).

Algorithme 3.9 – IdentifierDoublonsEtFusionner

```
1  IdentifierDoublonsEtFusionner(Ax_dg, G_AH):
2    eq = ∅×∅  //Un ensemble associatif clef/valeur
3    //clef: Acception, valeur: Ensemble d'acceptions identiques
4
5    //Identification des acceptions identiques par degre
6    pour chaque dg dans uniques(degres):
7      Ax = Ax_dg[dg]
8      pour i dans 1..|Ax_dg[dg]|:
9        pour j dans 1..|Ax_dg[dg]|:
10         si i≠j et Ax[i] == Ax[j]:
11           //Ajout avec contrainte d'unicite
12           eq[Ax[i]] = (eq[Ax[i]] ∪ {i, j}) \ (eq[Ax[i]] ∩ {i, j})
13   //Fusion des acceptions
14   pour chaque clef dans clefs(eq):
15     //Supprime les doublons et modifie les arcs pour pointer vers
           une unique acception
16     Fusionner(eq[clef], G_AH)
17   retour G_{\acceptionhierarchyset}
```

3.3.2.2 *Complexité dans le pire des cas et performance à l'échelle*

La raison principale d'exister de cet algorithme est de trouver une solution avec une complexité dans le pire des cas moins prohibitive que pour l'algorithme récursif. Ainsi, nous allons ici utiliser la même situation du "pire cas" décrite dans la section Section 3.3.1.2 portant sur la complexité dans le pire cas pour l'algorithme récursif.

Examinons maintenant la complexité des différentes parties de l'algorithme :

— **Acception.** La fonction Acception crée un nouvel objet acception à partir d'un ensemble de sens, nous considérerons qu'elle opère en temps constant, soit en $O(1)$.
— **BronKerbosch.** (Algorithme 3.4) : La variante de l'algorithme de BronKerbosch utilisée ici a une complexité de $O(dn3^{\frac{d}{3}})$, où d est le degré maximal dans le graphe.
— **Degres.** Le calcul des degrés d'un ensemble de sommets consiste, pour chaque sommet à compter le nombre de voisins, dans le pire des cas, il y aura $n-1$ voisins, et donc si il y a n sommets dans l'ensemble (dans notre cas les sommets correspondent aux sens), la complexité de cet algorithme sera $O(n \cdot (n-1)) = O(n^2)$.
— **TrierParDegre.** : La complexité est celle d'un tri considéré optimal, c'est-à-dire $O(n \cdot log(n))$ ainsi que celle de la permutation de la liste des cliques en fonction du tri par degré ce qui est constant pour chaque élément, soit $O(n)$. Au total la complexité est dominée par celle du tri en $O(n \cdot log(n))$.
— **SegmenterParDegre.** (Algorithme 3.7). La segmentation réalise uniquement des opérations en coût constant, en $O(1)$, dans une boucle parcourant les degrés de chacun des n nœuds (nombre de sens), soit une complexité en $O(n)$.
— **Création des liens entre segments par degrés des cliques** : Cette partie du code parcourt l'ensemble des degrés des arcs ($O(n)$), puis parcourt les cliques (dont le nombre est largement inférieur à n). Le reste des opérations est en $O(1)$, soit une complexité finale de $O(n)$.
— **Identification des acceptions égales.** Ici, nous parcourons l'ensemble des degrés uniques qui normalement est bien inférieur à n, ensuite il y a deux boucles imbriquées qui itèrent sur les acceptions par degré. Quand toutes les acceptions ont toutes un unique degré (pire des cas), ces itérations reviennent à $O(n^2)$ opérations de coût constant.
— **Fusion des acceptions.** La fusion n'itère que sur les acceptions égales, qui sont en nombre bien inférieur au nombre d'arcs du graphe initial, nous considérerons que c'est une opération de coût constant en $O(1)$.

Ainsi, au total nous aurons une complexité dans le pire des cas bornée par :

$$O(1 + dn3^{\frac{d}{3}} + n^2 + n \cdot log(n) + n + n + n^2 + 1) \in O(dn3^{\frac{d}{3}} + n^2)$$

La complexité dans le pire des cas est dominée par la recherche de cliques, soit $O(dn3^{\frac{d}{3}})$, ce qui est exponentiel par rapport au degré et qui en pratique pour les graphes de ressources lexicales, croit plus vite que le terme n^2. La complexité de cette algorithme dans le pire des cas est meilleure que l'algorithme récursif.

3.4 Stratégies de mise à jour

Nous avons ici proposé deux algorithmes pour une construction initiale d'acceptions interlingues sur la base de la formalisation axiomatique proposée. Cependant, la construction initiale n'est qu'une petite partie du cycle de vie d'une base lexico-sémantique à acceptions interlingues.

Des ressources lexico-sémantiques sont amenées à évoluer, comme nous l'avons illustré dans le Chapitre 1. Dans ce cas, il faut proposer des algorithmes capables de gérer les ajouts, suppressions et mises à jour d'éléments de la base d'acceptions, d'une manière garantissant la validité axiomatique du résultat à la suite de

la modification. Par ailleurs, il est fréquent qu'on veuille rajouter de nouveaux volumes dans de nouvelles langues, sans nécessairement devoir repasser par les alignements bilingues (c'était l'un des avantages évoqués à propos des architectures à base de pivot).

3.4.1 Ajout d'entités ou de relations d'alignement

L'ajout d'une entité est une opération essentielle d'une base d'acceptions, en particulier dans le cas où la source des données est de nature collaborative. Il s'agit ici plus particulièrement d'ajouter une nouvelle entrée lexicale ou un sens de mot à une entrée existante. L'ajout d'une entrée lexicale peut cependant se ramener à l'alignement de chacun de ces sens aux hiérarchies d'acceptions correspondantes. L'Algorithme 3.10 illustre les grandes étapes de l'algorithme d'alignement et sera commenté dans ce qui suit.

La première étape sera de trouver les hiérarchies d'acceptions candidates vers lesquelles nous pourrons aligner les sens. Il faudra donc disposer d'au moins un dictionnaire bilingue vers l'une des langues présentes dans les hiérarchies, afin de trouver la correspondance entre la nouvelle entrée et une entrée candidate pour l'alignement, c'est-à-dire l'ensemble des hiérarchies d'acceptions associées aux sens des traductions du mot correspondant à l'entrée lexicale. Dans le cas où l'on dispose d'un graphe de traduction (qui aurait par exemple servi à la construction de la base initiale d'acceptions), il est préférable de calculer la fermeture de traduction du mot dans le graphe (ligne 5, appel à `FermetureTransitiveDeTraduction`[3]) et de récupérer les hiérarchies d'acceptions uniques des sens de toutes les entrées présentes dans la fermeture (ligne 6, appel à `RecupererHierarchiesCandidates`), afin de s'assurer d'avoir au moins une acception correspondante.

Il faudra ensuite trouver, parmi toutes les hiérarchies candidates, celle qui est la plus proche (ligne 7, appel à `HierarchieLaPlusProche`), ce qui nécessitera une similarité ou une distance entre les sens de l'entrée lexicale et une acception. Si l'on suppose l'existence d'une mesure de similarité $\mathtt{sim} : \mathcal{S} \times \mathcal{S} \mapsto [0,1] \subset \mathbb{R}$ translingue entre des sens dans des langues différentes appartenant à la ressource, nous pourrons définir la mesure $\mathtt{sim} : \mathcal{S} \times \mathcal{A}x \mapsto [0,1] \subset \mathbb{R}$ qui opère entre les sens et les acceptions comme $\mathtt{sim}(s, ax) = \frac{\sum_{s_i \in \mathcal{S}(ax)} (\mathtt{sim}(s_i, s))}{|\mathcal{S}(ax)|}$, la moyenne des similarités entre le sens à aligner et les sens de l'acception. Par extension nous pouvons définir la similarité entre un sens et une hiérarchie d'acceptions ($\mathtt{sim} : \mathcal{AH} \times \mathcal{S} \mapsto [0,1] \subset \mathbb{R}$) en prenant la moyenne de la similarité entre le sens et chaque acception de la hiérarchie.

Nous pourrons ensuite utiliser la similarité entre une acception et un sens pour trouver l'acception la plus proche dans la hiérarchie (ligne 8). Cependant, il faut noter qu'il est fort probable que les similarités entre les sens et l'ensemble les acceptions soient très proches et qu'il soit difficile de faire une distinction statistiquement significative. Ainsi il conviendra de faire usage des motifs de divergence de lexicalisation (où d'une mesure la caractérisant) afin d'éliminer les acceptions candidates invalides vis-à-vis de l'axiomatisation des acceptions. Par ailleurs, nous supposons ici que nous évaluons l'ensemble des alignements bilingues (deux à deux) entre les sens de la nouvelle entrée et les hiérarchies d'acceptions candidates, ce qui fait perdre son intérêt à l'utilisation d'acceptions interlingues plutôt qu'une architecture par transfert. Avant de pouvoir passer à la dernière étape de

3. Parcours exhaustif du graphe de traduction jusqu'à ce qu'aucun nouveau sommet ne puisse être visité, par exemple en profondeur d'abord comme dans l'Algorithme 3.2.

Algorithme 3.10 – AjouterEntreeLexicale

```
1 AjouterEntreeLexicale(el, G)
2 // el l'entree lexicale a ajouter
3 // G le graphe de la ressource lexicale
4 pour chaque s_k dans S(el):
5   fermeture = FermetureTransitiveDeTraduction(s_k, G)
6   AH = RecupererHierarchiesCandidates(fermeture)
7   ah* = HierarchieLaPlusProche(s_k, AH)
8   ax* = AlignerDansHierarchie(s_k, ah*) //Trouver acception de la
        hierarchie a aligner
9   CreerLiensSensAcception(ax*, G) // Creation des liens bilingues
        avec les sens de l'acception
```

l'algorithme d'alignement (ligne 9, appel à `CreerLiensSensAcception`), il faudra ainsi proposer des solutions à ces problèmes afin de pouvoir définir le comportement exact de la fonction `AlignerDansHierarchie`. Dans les deux sous-sections suivantes nous allons étudier deux approches possibles à ce problème, où la deuxième solution sera théoriquement plus élégante, mais nécessitera l'emploi et la création de ressources sémantiques nouvelles avec des propriétés spécifiques.

3.4.1.1 Alignements par ensembles noyaux d'une hiérarchie d'acceptions

La hiérarchie d'acceptions doit contenir au moins 1 sens (dans un volume particulier) par acception pour pouvoir exister. Ainsi, s'il y a des sens équivalents dans 10 langues dans une acception donnée, il serait uniquement nécessaire de calculer l'alignement bilingue avec l'un des sens associés à l'acception.

Pour une hiérarchie d'acceptions, s'il y a des sens dans toutes les acceptions de la hiérarchie provenant de la même ressource, alors il suffira d'aligner le nouveau sens aux sens de cette ressource-là uniquement.

Il est cependant peu probable qu'une langue soit présente dans toute une hiérarchie systématiquement. En effet il peut y avoir dans une hiérarchie plusieurs divergences qui sont causées par des lexicalisations divergentes vers des langues différentes. Dans ce cas, il faudra choisir à chaque niveau de la hiérarchie un sens correspondant à une langue où la divergence de lexicalisation modélisée à ce niveau de la hiérarchie est présente.

Nous pourrons alors définir la notion de noyau vis-à-vis d'une hiérarchie d'acceptions (**Définition** 3.16), qui représentera chaque acception par un sens unique, afin que les contraintes des lexicalisations divergentes soient préservées.

Définition 3.16 Noyau d'une hiérarchie d'acceptions – $\mathcal{N}(ah)$

Le noyau $\mathcal{N}(ah)$ d'une hiérarchie d'acception ah est un arbre de sens avec la même topologie que la hiérarchie, dans lequel chaque classe d'équivalence est représentée par l'un des sens qui la composent.
Chaque nœud de l'arbre représente une acception représentée par un sens unique (l'un des sens composant la classe d'équivalence). Il faut cependant qu'à chaque embranchement, les sens fils proviennent de la même ressource et appartiennent à des entrées lexicales différentes (il doit y avoir une divergence de lexicalisation).

Ainsi, nous pourrons utiliser ce noyau pour procéder à un alignement bilingue entre le nouveau sens et la hiérarchie. Une fois l'alignement bilingue créé, du fait de la transitivité de la relation d'équivalence, nous pourrons créer des alignements bilingues vers tous les autres sens reliés à cette acception ou hiérarchie d'acceptions indépendamment de leur volume d'appartenance.

Algorithme 3.11 – GenererNoyauAcceptions

```
1  fonction GenererNoyauAcceptions(ah, D_L(s),parent, G_ah):
2  //ah une hierarchie d'acceptions
3  //D_L(s) un ensemble associant a chaque paire de langue avec comme
       point de depart la langue du sens a aligner, un score
       correspondant a la qualite des ressources disponibles, la
       valeur allant de 0 a 1.
4
5    si G_ah ≠ ∅ × ∅:
6      G_N(ah) = ∅ × ∅
7      racine = Racine(ah)
8      sens = Sens(racine)
9      pour chaque enfant dans Enfants(racine):
10       sens = sens ∩ Sens(enfant)
11     sens = TrierParScore(sens, D_L)
12
13     si |sens| > 0:
14       V(G_N(ah)) = V(G_N(ah)) ∪ {sens[0]}
15       si parent ≠ nil:
16         E(G_N(ah)) = E(G_N(ah)) ∪ {(parent, sens[0])}
17     sinon:
18       lancer Exception("Il n'existe pas de noyau")
19
20     pour chaque enfant dans Enfants(racine):
21       GenererNoyauAcceptions(enfant, ah, D_L, G_ah)
```

Un problème persiste cependant. Sachant qu'il y a un unique alignement, nous ne pouvons plus exploiter de façon combinatoire la topologie du graphe d'alignement bilingue pour déterminer exactement à quel endroit de la hiérarchie le sens devra être aligné (il faudrait faire l'alignement bilingue avec au moins trois langues différentes à chaque niveau de la hiérarchie).

En effet, le sens sera probablement proche de tous les sens de la hiérarchie avec un critère d'alignement de type AAS présenté dans la Section 2.1.2.

Nous pouvons cependant dans ce cas de figure utiliser des dictionnaires bilingues donnant les traductions mot à mot pour déterminer si oui ou non il y a une divergence de lexicalisation.

Ainsi, si il y a une divergence dans les traductions de l'entrée lexicale qui contiennent le nouveau sens et les deux branches, alors le sens ne sera pas aligné aux acceptions plus spécifiques, mais à l'acception plus générale.

Il faudra répéter ce processus de manière récursive pour déterminer la position exacte dans la hiérarchie.

Formalisons maintenant un peu le processus, tout d'abord en proposant un algorithme qui permet de générer le noyau d'une hiérarchie d'acceptions.

Soit une hiérarchie d'acceptions $ah \in \mathcal{AH}$. Il existe plusieurs noyaux possibles pour une acception, et le choix d'un noyau particulier dépends de la langue l_s

à laquelle correspond le nouveau sens à aligner et des ressources bilingues dont nous disposerons.

De plus, pour générer le noyau de sens dans une langue l_c, il faut qu'il existe une ressource bilingue depuis l_s.

Ainsi, si nous n'imposons pas de contraintes sur les langues, nous pouvons simplement choisir les sens dans la langue qui est la plus présente à tous les niveaux de la hiérarchie.

Pour des parties de la hiérarchie auxquelles aucun sens de cette langue majoritaire ne serait rattaché, nous choisirons les sens de la langue qui correspond à la majorité de sens présents dans cette sous-hiérarchie.

L'Algorithme 3.11 présente une version synthétique de l'algorithme en pseudo-code. C'est un algorithme récursif, avec un appel récursif pour chaque acception dans la hiérarchie, et avec une complexité à chaque appel de $k \cdot \log(k)$ où k est le facteur de branchement[4]. Au total la complexité dans le pire des cas sera en $O(k^2 \log(k))$

Une fois le noyau obtenu, il faut procéder à l'alignement en lui-même et définir la fonction `AlignerDansHierarchie` utilisée dans l'algorithme d'alignement général que nous redéfinissons ici pour utiliser les noyaux de hiérarchie (Algorithme 3.12).

Algorithme 3.12 – AlignerDansHierarchie – Par noyau

```
1  AlignerDansHierarchie(s, ah) :
2   // s sens a aligner
3   //ah la hierarchie d'acception avec laquelle aligner
4
5   N(ah) = GenererNoyauAcceptions(ah, D_L(s), nil, G_ah)
6   candidats = ∅
7   pour chaque s_h dans ParcoursProfondeur(N(ah)) :
8     tr = Tr(lemme(s_h), L(s))
9     si |tr| = 1 et lemme(s) ∈ tr :
10      candidats = candidats ∪{Ax(s_h)}
11    sinon si |tr| > 1 et
          ∄s_f ∈ Fils(s_h) : lemme(s) ∈ Tr(lemme(s_f), L(s)) ∧ |Tr(lemme(s_f), L(s))| = 1 :
12        ax_new = Acception({s})
13        AjouterFils(Ax(s_h), ax_new)
14        candidats = {s_h}
15        arret //break
16  //Candidat qui maximise la similarite
17  retour arg max_{s_c ∈ canddidats}(sim(s_c, s))
```

L'algorithme est relativement simple et utilise un graphe de traduction pour vérifier, pour chaque acception de la hiérarchie, si le sens à aligner respecte les contraintes de divergence lexico-linguistique, ce qui constitue la base de la construction d'une hiérarchie d'acceptions.

— Nous commençons par générer le noyau de la hiérarchie (ligne 5) et par initialiser une liste d'acceptions candidates avec lesquelles un alignement est valide.

4. Le facteur de branchement est le nombre de fils pour chaque nœud de la hiérarchie. Ici pour la complexité dans le pire des cas, k correspondra au nombre de fils maximum parmi tous les nœuds.

— L'étape suivante consiste à parcourir les sens composant le noyau, en profondeur d'abord par exemple (ligne 7).
— Pour chaque sens, il nous faut récupérer les traductions du lemme du sens actuel dans le noyau vers la langue du sens actuel à aligner (ligne 8).
— S'il n'y a qu'un seule traduction et si cette traduction correspond au lemme du sens à aligner, nous sommes dans une situation où il n'y a pas de divergence de lexicalisation, et nous pouvons ajouter l'acception à la liste des acceptions candidates.

Dans le cas où il y a plus d'une traduction, il y a une divergence. Dans certains cas, il faudra créer une nouvelle acception dans la hiérarchie pour modéliser cette divergence s'il n'y a pas d'acception existante la modélisant, c'est-à-dire s'il **n'existe pas de fils de cette acception tel que son sens noyau (s_f)** respecte les conditions suivantes (ligne 11).

— Il y a une seule traduction du lemme du sens fils (s_f) dans la langue du sens à aligner.
— Le lemme du sens à aligner correspond à cette unique traduction.

Dans le cas où un tel sens fils n'existe pas, il faudra procéder à la création de l'acception composée uniquement du sens à aligner (ligne 12) et au rattachement de cette acception à la hiérarchie (ligne 13) et qui sera le seul candidat d'alignement (ligne 14).

Ainsi, à condition d'avoir au moins une ressource bilingue par niveau de chaque hiérarchie d'acceptions, il sera possible d'aligner de nouvelles acceptions à la ressource sans qu'il soit nécessaire de calculer tous les alignements bilingues.

Il faut noter ici, l'utilisation d'un critère de similarité sémantique pour permettre de choisir l'acception d'alignement, il conviens, tout comme en AAS pour des ressources deux à deux d'établir de manière empirique des seuils de similarité pour les différentes paires de langues qui permette de ne rien choisir si la similarité est trop basse où si il n'y a pas de différence significative de similarité entre les différentes valeurs pour les acceptions candidates. Un autre problème à considérer est qu'il faut de préférence garantir l'homogénéité des distributions des valeurs de similarité translingue pour les différentes paires de langues.

Cependant, nous pourrions souhaiter pouvoir comparer les sens avec les acceptions d'une hiérarchie d'une manière qui encode intrinsèquement les divergences. C'est ce que nous allons étudier dans ce qui suit.

3.4.1.2 Alignement direct de sens aux hiérarchies d'acceptions

Nous avons vu dans la section précédente comment réduire le nombre d'alignements bilingues uniquement à ceux qui sont essentiels. Cependant, ne serait-il pas possible d'aligner directement de nouveaux sens aux classes d'équivalence sans devoir passer par les autres langues, mais en comparant directement l'information sémantique portée par la signature sémantique du sens ?

Il pour que cette approche soit possible, il faut alors attribuer une représentation sémantique propre à la hiérarchie d'acceptions indépendante d'une langue particulière.

Par rapport aux techniques présentées dans le Chapitre 1 et le Chapitre 2, ce type de représentation sémantique pourrait être fondée sur une représentation telle qu'UNL ou encore sur une représentation vectorielle multilingue projetée directement sur les hiérarchies d'acceptions.

Utiliser UNL nécessiterait d'avoir des dictionnaires spécifiques UNL-langue (existent pour plus de 7 langues avec plus d'un million entrées), alors que dans le cas d'espaces vectoriels multilingues, il faudrait avoir à disposition des dictionnaires bilingues ou alors des corpus parallèles (AMMAR et al., 2016), plus aisés à obtenir. Les deux approches nécessiteraient une recherche plus poussée sur le moyen ou voire même sur le long terme. Cependant, même si nous ne pouvons pas encore construire une telle représentation sémantique, nous pouvons caractériser les propriétés théoriques d'une mesure de proximité sémantique que nous pourrions utiliser pour aligner un sens nouveau aux acceptions.

La première contrainte est le respect des propriétés de la relation d'ordre partiel strict qu'est la relation de raffinement. En prenant deux acceptions $ax_a, ax_b \in \mathcal{A}x$: $ax_a > ax_b$, il faut que la similarité entre les deux sens puisse retranscrire les contraintes. Une solution pour encoder cette contrainte est de ne plus utiliser une similarité, mais une mesure de divergence **Définition** 3.17). L'asymétrie de la mesure représenterait la relation d'ordre partiel : si une acception ax_a est plus générale qu'une autre ax_b, alors la divergence entre ax_a et ax_b doit être plus grande que la divergence entre ax_b et ax_a(**Axiome** 3.5).

Définition 3.17 **Divergence –** $D(a \parallel b)$

Une divergence est une fonction $D(\bullet \parallel \bullet) : S \times S \to \mathbb{R}^+$, où S est une variété statistique (variété servant de support à une distribution statistique) similaire à une distance, mais ne nécessitant pas de respecter la symétrie et l'inégalité triangulaire.
Nous étendons ici la définition pour opérer sur l'ensemble $S = \mathcal{A}x \cup \mathcal{S}$ que l'on suppose être d'une variété statistique.

Une divergence D doit satisfaire les deux conditions suivantes :
- Non-négativité : $D(p \parallel q) \geqslant 0$
- $D(p \parallel q) = 0$ ssi $p = q$

Nous noterons le dual de la divergence comme :

$$D^*(a \parallel b) = D(b \parallel a)$$

Axiome 3.5 **Isomorphisme entre ordre de raffinement et de divergence**

Soit deux acceptions $ax_a, ax_b \in A \subset \mathcal{A}x$ et deux ordres partiels, l'un induit par l'opérateur de raffinement sur l'ensemble A et l'autre induit par l'opérateur de comparaison sur les divergences entre les éléments de A : $(A, <_{D(\cdot \parallel \cdot)})$, $(A, >)$. La topologie définie par les opérateurs pour les deux ordres partiels est la même, c'est-à-dire que :
- $D(ax_a \parallel ax_b) > D^*(ax_a \parallel ax_b) \Leftrightarrow ax_a > ax_b$
- $D(ax_a \parallel ax_b) < D^*(ax_a \parallel ax_b) \Leftrightarrow ax_a < ax_b$
- $D(ax_a \parallel ax_b) = D^*(ax_a \parallel ax_b) \Leftrightarrow ax_a = ax_b$

Ainsi, à partir de l'**Axiome** 3.5, il est possible de définir une notion de proximité vis-à-vis de la position d'une acception dans une hiérarchie d'acceptions, en définissant une mesure de l'asymétrie de la divergence entre deux acceptions. C'est ce que nous appellerons un indice d'asymétrie (**Définition** 3.18).

Définition 3.18 **Indice d'asymétrie**

L'indice d'asymétrie est la différence entre la divergence et son dual.

$$\Delta_D(s_n, ax_i) = D(s_n \parallel ax_i) - D^*(s_n \parallel ax_i)$$

D'après l'**Axiome 3.5**, une asymétrie positive signifiera que le premier argument est plus général que le deuxième et une asymétrie négative signifiera que le premier argument est plus spécifique que le deuxième. Ainsi, il sera possible de trouver le niveau adéquat dans la hiérarchie en essayant de trouver l'indice d'asymétrie le plus proche de zéro. Il restera ensuite à retrouver la bonne acception sur un niveau donné de la hiérarchie, ce que nous pourrons faire en cherchant l'acception la plus similaire (avec la divergence la plus petite) au sens à aligner. Il convient d'établir des seuils de divergence de manière empirique afin de permettre de détecter des cas où il faut créer une nouvelle acception et non pas simplement aligner le nouveau sens à une acception existante dans la hiérarchie.

Ainsi, l'asymétrie ayant joué son rôle, nous pourrons utiliser une version symétrique de la divergence pour réaliser l'alignement vers l'acception précise. La méthode la plus simple est de faire la somme de la divergence et de son dual, cependant cela ne préserve pas l'intervalle de valeurs de départ. Un meilleur moyen de réaliser cette normalisation est calculer une moyenne entre la divergence et son dual (**Définition 3.19**).

Définition 3.19 **Divergence symétrique normalisée**

À l'instar du score F1 qui combine précision et rappel de manière équilibrée, nous pouvons équilibrer la divergence et son dual par le calcul de la moyenne. Si $D(a \parallel b)$ et $D^*(a \parallel b)$ sont dans le même intervalle de valeurs, nous pouvons utiliser la moyenne arithmétique :

$$D_s(a,\ b) = \frac{D(a \parallel b) + D^*(a \parallel b)}{2}$$

Si la divergence est un rapport, alors il sera préférable d'utiliser la moyenne harmonique :

$$D_s(a,\ b) = \frac{2 \cdot D(a \parallel b) \cdot D^*(a \parallel b)}{D(a \parallel b) + D^*(a \parallel b)}$$

En conséquence l'alignement avec une hiérarchie devient un problème de minimisation où l'acception sélectionnée pour l'alignement est celle pour laquelle l'indice d'asymétrie Δ_D ainsi que la divergence symétrisée sont minimaux suivant la fonction d'objectif de la **Définition 3.20**.

Cette approche permet cependant uniquement de localiser l'acception au niveau de laquelle l'insertion doit avoir lieu. Si l'acception candidate est une feuille et si nous nous trouvons dans une situation de contraste, il faudra utiliser une ressource bilingue pour réaliser l'alignement de la même manière que dans la Section 3.4.1.1.

Le fait de devoir utiliser quand même les traductions et potentiellement les noyaux d'hiérarchies semble enlever tout intérêt à cet algorithme d'alignement à base de divergences. Cependant l'intérêt principal du calcul d'une divergence respectant les contraintes liées à la hiérarchie est applicatif, car une telle divergence serait un outil précieux pour exploiter une base d'acceptions de manière pratique.

Définition 3.20 **Fonction objectif pour l'alignement d'acceptions**

La fonction d'objectif cherche à minimiser l'asymétrie d'abord, puis ensuite la divergence symétrisée :

$$\begin{aligned} \mathcal{A}x_{\mathrm{obj}} : \mathcal{S} \times \mathcal{A}x &\longrightarrow \mathbb{R} \\ (s, ax) &\longmapsto \alpha \cdot \Delta_{\mathrm{D}}(s, ax) + \beta \cdot \mathrm{D}_s(s,\ ax) \end{aligned}$$

Comme l'objectif prioritaire est d'avoir l'asymétrie la plus petite possible, la divergence servant surtout à départager les acceptions à un même niveau d'asymétrie, il faudra choisir $\alpha >> \beta$.

Algorithme 3.13 – AlignerDansHierarchie – Par divergence

```
1  AlignerDansHierarchie(s, ah) :
2   // s sens a aligner
3   //ah la hierarchie d'acception avec laquelle aligner
4    N(ah) = GenererNoyauAcceptions(ah, D_L(s), nil, G_ah)
5   candidat = arg min_{s_h ∈ ParcoursProfondeur(N(ah))} (Ax_obj(s_h, s))
6
7   si estFeuille(candidat, ah) :
8     tr = Tr(lemme(candidat), L(s))
9     si |tr| > 1 et ∄s_f ∈ Fils(candidat) : lemme(s) ∈
          Tr(lemme(s_f), L(s)) ∧ |Tr(lemme(s_f), L(s))| = 1 :
10      ax_new = Acception({s})
11      AjouterFils(Ax(s_h), ax_new)
12      candidat = {ax_new}
13
14  retour candidat
```

Cela est particulièrement vrai du fait de la popularité et du succès des approches distributionnelles à la modélisation des langues et à la production d'espaces distributionnels multilingues. Ainsi, la hiérarchie d'acceptions servirait de structure topologique pour le plongement, afin de doter les espaces sémantiques vectoriels d'une topologie compatible avec le pivot interlingue[5]. La construction contrainte de ces projections est un verrou important.

Quant à l'alignement en lui-même, nous pouvons donc redéfinir la fonction d'alignement d'un sens à une hiérarchie en utilisant la méthode d'alignement par divergence dans l'Algorithme 3.13

3.4.2 Suppression d'entités ou de relations d'alignement

Maintenant que nous avons défini quelques algorithmes pour l'addition d'éléments dans des hiérarchies d'acceptions, une opération tout aussi importante sera la suppression d'éléments.

Contrairement à l'ajout, la suppression est relativement simple, dans le sens où il suffit d'enlever l'élément en s'assurant que la validité axiomatique est préser-

5. À l'heure actuelle la construction des espaces sémantiques vectoriels multilingues avec plus de deux langues s'apparente à l'utilisation de l'espace vectoriel d'une langue en particulier comme pivot pour l'ensemble des projection dans l'espace commun, ce qui induit les mêmes phénomènes de contraste qu'avec les RLS

vée après la suppression et que la hiérarchie reste dans un état consistant. Nous pouvons recenser plusieurs cas de figure de suppression.

1. Suppression d'un sens appartenant à une acception. Si le sens est le seul constituant de l'acception qui le contient, alors il conviendra de supprimer cette acception. L'algorithme de suppression est alors trivial (Algorithme 3.14).

Algorithme 3.14 – SupprimerSens

```
1  SupprimerSens(s, ah) :
2    ax = Ax(s) \ {s}
3    si |ax| = 0:
4      SupprimerAcception(ax, ah)
```

2. Suppression d'une acception appartenant à une hiérarchie. Cela arrivera uniquement si l'acception devient vide à la suite d'une suppression de sens (Algorithme 3.15 illustré par la Figure 3.7, la Figure 3.8 et la Figure 3.9) :

Algorithme 3.15 – SupprimerAcception

```
1  SupprimerAcception(ax, ah) :
2    //Unique acception, on la supprime. Cas a).
3    si |Enfants(ax_p, ah)| = 0:
4      ah = ah \ {({ax}, {(ax, y)|∀y : (ax, y) ∈ E(ah)})}
5
6    //Cas b).
7    si estRacine(ax, ah) :
8      //Suppression de la racine et de tous les arcs du
           graphe de la hierarchie partant de la racine
9      ah = ah \ {({ax}, {(ax, y)|∀y : (ax, y) ∈ E(ah)})}
10     pour chaque enfant dans enfant(ax, ah):
11       //Une nouvelle hierarchie par enfant
12       ah_e = SousHierarchie(enfant, ah)
13
14   //Cas c).
15   si estFeuille(ax, ah) :
16     ax_p = parent(ax)
17     //Suppression de la feuille
18     ah = ah \ {({ax}, {(ax, y)|∀y : (ax, y) ∈ E(ah)})}
19     si |Enfants(ax_p, ah)| = 1:
20       //Ajout des sens de l'unique enfant restant dans l'
             acception mere
21       ax_e = Enfants(ax_p, ah)[0]
22       Sens(ax_p) = Sens(ax_p) ∪ Sens(ax_e)
23       //Supression de l'unique feuille apres fusion avec le
             parent
24       ah = ah \ {({ax_e}, {(ax_e, y)|∀y : (ax_e, y) ∈ E(ah)})}
25
26   //Cas d).
27   sinon:
28     ax_p = parent(ax)
29     enfants = Enfants(ax, ah)
30     //Supression de l'acception
31     ah = ah \ {({ax}, {(ax, y)|∀y : (ax, y) ∈ E(ah)})}
32     //On lie les enfants avec le parent
33     pour chaque ax_e dans enfants:
34       E(ah) = E(ah) ∪ {(ax_p, ax_e)}
```

a) Si la hiérarchie n'est composée que de cette unique acception, la hiérarchie tout entière est supprimée.

b) Si nous supprimons la racine de l'acception, alors chacun des enfants de cette acception dans la hiérarchie formera sa propre hiérarchie (Figure 3.7). En effet, les sens des acceptions filles n'auront plus aucun alignement en commun si la racine est supprimée.

c) Si nous supprimons une acception feuille de la hiérarchie et si l'acception parent n'a qu'un enfant restant, alors il n'y a plus de divergence de traduction et l'acception enfant restante est fusionnée avec son parent (Figure 3.8). En effet, selon le **Théorème** 3.4, le facteur de branchement dans une hiérarchie est de deux acceptions. Ainsi, vis-à-vis de l'algorithme de construction des acceptions, si il ne reste qu'une acception fille, c'est qu'il n'y a plus au niveau des alignement bilingues deux cliques séparées. Ainsi l'unique acception fille fera partie de la même clique que l'acception de niveau supérieur, et elle devra être fusionnée avec l'acception parent.

d) Si nous supprimons une acception qui a à la fois un parent et des enfants, alors les enfants de l'acception supprimée sont adoptés par son parent (Figure 3.9). Cela découle de la transitivité de la relation de raffinement (**Axiome** 3.1). En effet si l'on considère l'acception parent $ax_{\mathrm{p}} \in \mathcal{A}x$, l'acception supprimée $ax_{\mathrm{spp}} \in \mathcal{A}x$ et ses acceptions filles $ax_{\mathrm{f,i}} \in \mathcal{A}x$, nous aurons que $\forall ax_{\mathrm{f,i}}, ax_{\mathrm{p}} > ax_{\mathrm{spp}} > ax_{\mathrm{f,i}} \implies ax_{\mathrm{p}} > ax_{\mathrm{f,i}}$. Ainsi, après suppression de ax_{spp}, la relation $ax_{\mathrm{p}} > ax_{\mathrm{f,i}}$ implique que les acceptions filles soient adoptées par l'acception parent.

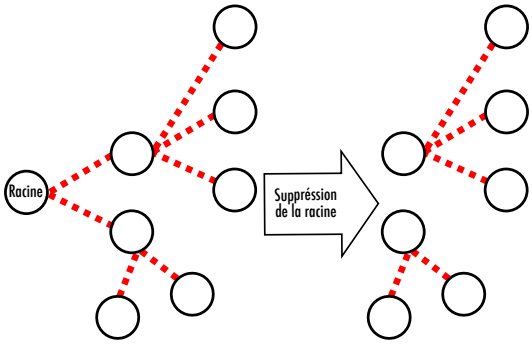


FIGURE 3.7 – Exemple de suppression de l'acception racine d'une hiérarchie.

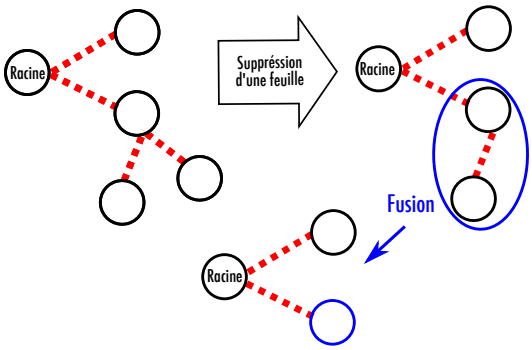


FIGURE 3.8 – Exemple de suppression d'une acception feuille d'une hiérarchie.

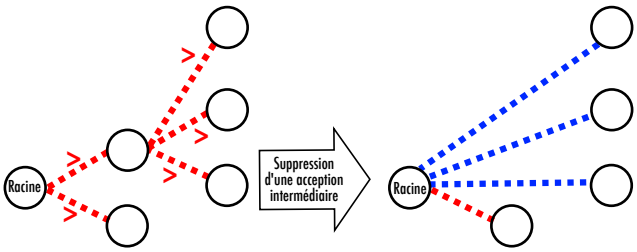


FIGURE 3.9 – Exemple de suppression d'une acception au centre d'une hiérarchie.

3.4.3 Validité axiomatique des acceptions produites

Les algorithmes de construction initiale des hiérarchies d'acceptions ainsi que les algorithmes de mise à jour sont construits sur la base des axiomes et théorèmes régissant les acceptions interlingues et les relations de raffinement. Ainsi, toutes ces opérations garantissent que les hiérarchies existantes restent axiomatiquement correctes. Par contre, le processus de construction initiale se base sur des alignements automatiques de sens (deux à deux) existants qui eux peuvent contenir des erreurs. Par ailleurs, les algorithmes d'ajout peuvent aussi contenir des erreurs d'alignement, notamment dans la sélection de la hiérarchie la plus proche, qui utilise des mesures de similarité heuristiques. Cependant, ces erreurs ne se manifesteront pas sous la forme de violations axiomatiques et devront être évaluées empiriquement, ce qui sera abordé principalement dans la Section 4.3 du Chapitre 4.

Conclusions

Dans ce chapitre, nous avons formalisé la notion d'acception interlingue et la relation de raffinement, des notions existantes mais jamais formalisées, ce qui nous a permis de proposer un ensemble d'algorithmes, d'une part pour la création initiale d'une base d'acceptions, et d'autre part pour la mise à jour et pour l'alignement direct. Il est ainsi en théorie possible de construire une base d'acceptions à partir d'un ensemble de paires de ressources, puis d'aligner d'autres ressources directement par l'approche noyau, à condition d'avoir les ressources bilingues nécessaires pour distinguer toutes les divergences de lexicalisations. Il conviens de noter le manque d'une approche permettant la fusion de deux bases d'acceptions différentes, qu'il faudra formaliser et développer ultérieurement.

Nous pouvons cependant identifier plusieurs limitations. La première limitation est le fait de devoir passer par l'ensemble des alignements bilingues pour construire une hiérarchie d'acceptions initiale. Cependant, la méthode d'alignement direct sur hiérarchie existante permettrait de construire une hiérarchie initiale minimale à partir de langues où il est plus aisé de produire les alignements bilingues, puis d'utiliser l'algorithme d'ajout direct pour construire le reste de la ressource.

Une autre limitation est que les algorithmes proposés permettent la construction à partir de ressources monolingues et bilingues uniquement, malgré la présence de ressources à base de pivot naturels (par. ex. BabelNet, EuroWordNet) que l'on pourrait vouloir intégrer et convertir du fait de leur taille et de leur couverture importante.

Il conviendra de développer un algorithme de conversion d'une ressource à base de synsets pour permettre de corriger les contrastes artificiels en construisant une hiérarchie d'acceptions directement à partir des synsets.

Par ailleurs, l'approche d'alignement direct par divergence ébauchée est loin d'être complète, étant donné que la difficulté principale est la création des projections vectorielles sur la hiérarchie d'acceptions. L'apparition de plongements sémantiques multilingues projetés dans plusieures langues à la fois est cependant une piste très prometteuse pour arriver à une projection sur une hiérarchie d'acceptions.

Enfin, il conviendra de terminer cette conclusion sur une note pratique, car l'alignement de ressources pose de nombreuses contraintes liées à la nature de la ressource et à la qualité des données. Ainsi, dans le prochain chapitre, nous étudierons le cas de DBNary et verrons quelles étapes sont nécessaires avant de pouvoir construire des acceptions interlingues à partir de la ressource.

Chapitre 4

DBNary
–
Une ressource massive par acceptions interlingues

Sommaire

Introduction

Nous avons vu dans le précédent chapitre la formalisation axiomatique des acceptions interlingues, les algorithmes de construction initiale et de mise à jour. Il y a cependant une condition préalable à l'application des algorithmes proposés : il faut déjà avoir l'ensemble des alignements bilingues entre toutes les paires de ressources.

En pratique, peu de ressources vont remplir toutes les conditions dès le départ et il faudra d'abord effectuer des traitements préalables et les évaluer pour pouvoir passer à l'étape suivante qui est la création d'acceptions interlingues. Dans ce chapitre, le cas d'étude sera *DBNary*, qui comme présenté dans le Chapitre 1 est une ressource extraite de Wiktionary, et dans laquelle il y a des relations de traduction, mais uniquement attachées aux vocables et non directement aux sens.

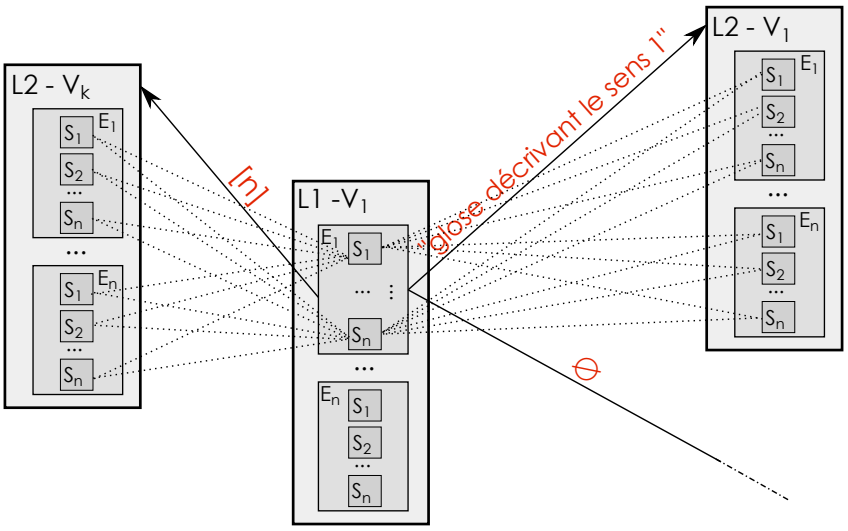


FIGURE 4.1 – Les données de *DBNary* post-extraction sans pré-traitement des relations de traduction.

Il s'agira principalement de se ramener à un graphe de traductions où toutes les relations bilingues partent et arrivent sur des sens de mots/acceptions monolingues, ce qui est un peu plus aisé que de faire de l'AAS directement. En effet, AAS ne présuppose aucun lien existant entre les deux ressources, alors que *DBNary* est déjà un graphe de traductions bilingues, même si il est incomplet. Dans un premier temps nous exploiterons les informations présentes dans *DBNary* et non explicitement représentées dans la structure de la ressource afin d'inférer les sens source de certaines relations de traduction. Dans un deuxième temps, nous proposerons des approches pour la détermination des sens source non traités dans l'étape précédente ainsi que des cibles des relations de traduction.

La Figure 4.1 présente la situation initiale dans *DBNary*, avant l'application de tout pré-traitement. Les traits pleins correspondent aux informations effectivement disponibles alors que les traits en pointillé correspondent à toutes les combinaisons possibles qu'il faudrait évaluer pour trouver toutes les correspondances avec les bons sens. Entre deux entrées lexicales, ayant respectivement n et m sens, il y a en tout $n \times m$ combinaisons possibles.

Il faudra ainsi, en utilisant au mieux les informations présentes, identifier le ou les sens source et cible des liens de traduction.

N.B. Cette section se base sur un article de séminaire, publié en 2014 avec Gilles Sérasset comme second auteur (TCHECHMEDJIEV et al., 2014).

4.1 Attachement des traductions aux sens source

Dans *DBNary*, tel qu'extrait de Wiktionary, les relations de traduction sont associées aux entrées lexicales et correspondent à la forme écrite de la traduction dans la langue cible. Dans les éditions de Wiktionary plus volumineuses, les traductions sont classées par sens. Une glose est fournie et permet d'évoquer à un utilisateur humain le sens correspondant. Dans de rares cas, les numéros des sens sont également fournis (une partie des traductions du français, de l'allemand, de l'anglais, du portugais, du japonais).

Ainsi, lorsque le numéro de sens est présent, il est directement possible d'aligner la relation de traduction au sens source et lorsqu'il y a une glose textuelle,

Langue	Entrées	Entrées Lexicales	Traductions	Gloses	Texte	N. Sens	Txt.+N.S.
anglais	544,338	438,669	1,317,545	1,288,667	1,288,667	515	515
finnois	49,620	58,172	121,278	120,728	120,329	115,949	115,550
français	291,365	379,224	504,061	136,319	135,612	28,821	28,114
allemand	205,977	100,433	388,630	388,553	3,101	385,452	0
grec	242,349	108,283	56,638	8,368	8,368	12	12
italien	33,705	47,102	62,546	0	0	0	0
japonais	24,804	28,763	85,606	22,322	20,686	4,148	2,512
portugais	45,109	81,023	267,048	74,901	72,339	71,734	69,172
russe	129,555	106,374	360,016	151,100	150,985	115	0
turc	64,678	91,071	66,290	53,348	585	52,901	138

TABLE 4.1 – Statistiques sur *DBNary* en septembre 2014 pour les différentes langues disponible à ce moment-là : nombre d'entrées, nombre de sens, nombre de relations de traduction, nombre de traductions avec une glose associée, nombre de traductions où la glose est du texte, nombre de traductions où la glose est un numéro de sens, nombre de traductions où il y a les deux.

il est possible d'établir une correspondance avec le sens en comparant la sémantique portée par la glose à celle de la définition des sens afin de déterminer celui ou ceux qui sont les plus proches. Dans les cas où les deux informations sont disponibles, il sera possible de construire un jeu de données d'évaluation étalon de référence afin de permettre d'évaluer la qualité de l'alignement.

Dans l'état de l'art, en ce qui concerne l'attachement des traductions aux sens source, les travaux les plus similaires aux nôtres sont ceux de Christian M MEYER et GUREVYCH (2012a). Ils ont le même objectif que nous ; cependant, ils se limitent aux éditions de langue de Wiktionary présentes dans Uby, c'est-à-dire aux versions allemande et anglaise. Leur jeu de données d'évaluation a été créé manuellement et est significativement plus petit que notre jeu de données d'évaluation endogène généré à partir des informations présentes dans Wiktionary.

De plus, contrairement à nous, ils utilisent l'assignement au premier sens quand une correspondance n'a pu être déterminée par leur algorithme (fondé sur la structure de la ressource et sur des mesures de similarité). Une autre différence est que leur mesure de similarité prend également en compte des informations telles que le domaine, le registre, etc., alors notre similarité ne le fait pas pour la raison que l'extracteur *DBNary* ne permet pas encore de récupérer ces informations depuis Wiktionary.

Nous arrivons à atteindre une précision et un rappel d'alignement du même ordre qu'eux pour l'allemand et l'anglais, mais aussi pour l'ensemble des éditions des langues traitées ici, y compris des langues telles que le finnois avec une morphologie complexe, malgré le fait que nous utilisons uniquement de mesures de similarité se basant sur une comparaison de chaînes et des formes de mot.

Dans les travaux de Christian M MEYER et GUREVYCH (2012a), une similarité à base de traits est utilisée (recouvrement de définitions ou gloses). Lors de travaux précédents, ils utilisèrent une mesure de similarité textuelle fondée sur un modèle sémantique vectoriel calculé à partir d'un corpus (Analyse Sémantique Ex-

plicite ou dans l'anglais original *Explicit Semantic Analysis*) (Christian M Meyer et Gurevych, 2010).

Ici nous proposons l'utilisation d'une mesure de similarité à base de glose (Indice de Tversky (1977b)) où nous remplaçons la correspondance exacte des formes de la gloses par une mesure de distance de chaîne (distance de Levenshtein) qui permet d'identifier une correspondance partielle entre les formes de surface.

L'utilisation d'une distance de chaîne approchée est un cas particulier de la notion de "cardinalité douce"proposée par Jimenez, Becerra et Gelbukh, 2012 ; Jimenez, Gonzalez et Gelbukh, 2010, où la cardinalité est la somme des correspondances de chaîne partielles (valeur entre 0 et 1) entre toutes les paires de mots des définitions au lieu d'une correspondance exacte (valeurs uniques, 1 pour une correspondance exacte, 0 sinon).

Définition 4.1 N-gramme

«Un n-gramme d'ordre n est une suite de n éléments successifs construits à partir d'une séquence d'éléments.»[a] Ce type de modèle permet de construire des modèles de langue où pour chaque n-gramme de lexèmes successifs, la vraisemblance que les n mots apparaissent ensemble dans un corpus est calculée sur la base de la fréquence d'apparition des n-grammes de lexèmes (C. Manning et Schütze, 1999).

a. https://fr.wikipedia.org/wiki/N-gramme

Alors que la cardinalité douce de Jimenez, Becerra et Gelbukh (2012), utilise un modèle de langue dit q-gramme (avec des n-grammes (**Définition 4.1**) de caractères correspondant aux sous chaînes possibles de chaque mot) combiné à une pondération en information mutuelle point par point [1], notre mesure n'utilise pas d'information autre que celles contenue dans les définitions/gloses. En effet, la construction d'un modèle q-gramme pour l'ensemble des 10 langues traitées (en septembre 2014, en octobre 2016 sur 20 éditions de langues) serait très difficile du fait de la grande variabilité de la disponibilité de ressources langagières pour chacune des langues.

Cette approche a été exploitée extensivement dans les tâches de similarité textuelle dans la campagne d'évaluation SemEval, que ce soit spécifiquement par le système de Jimenez, Becerra et Gelbukh (2012), ou par d'autres systèmes tels que (S. Wu et al., 2013) qui utilisent une distance basée sur la sous-chaîne commune la plus longue.

4.1.1 Relations de traduction

DBNary utilise le formalisme Ontolex pour sa représentation, mais ce modèle ne permet pas directement de représenter des relations de traduction.

Ainsi, ces dernières sont représentées de manière *ad hoc*. *DBNary* définit la classe `dbnary:Translation`, qui permet de relier deux entités du modèle :

$$\mathrm{LexicalEntry} \cup \mathrm{LexicalSense} \rightarrow \mathrm{Vocable} \cup \mathrm{LexicalEntry} \cup \mathrm{LexicalSense}$$

.

1. *Point-Wise Mutual Information* (PWMI).

L'**Exemple 4.1** illustre les données d'une relation de traduction partant de l'entrée lexicale anglaise *frog*.

Exemple 4.1 **Instance d'une traduction dans *DBNary***

Une instance de traduction[a] présente une référence à l'entité à laquelle elle correspond en source, la langue cible (représentée grâce à l'ontologie `lexvo.org` MELO et WEIKUM, 2008), l'information d'usage telle qu'extraite de Wiktionary (par exemple le genre), la forme écrite de la traduction dans la langue cible et optionnellement une glose décrivant le sens source auquel correspond la traduction.

```
eng:__tr_fra_1_frog__Noun__1    a dbnary:Translation ;
       dbnary:gloss  "amphibian"@en ;
       dbnary:isTranslationOf eng:frog__Noun__1 ;
       dbnary:targetLanguage lexvo:fra ;
       dbnary:usage  "f" ;
       dbnary:writtenForm  "grenouille"@fr .
```

a. Syntaxe turtle de RDF.

Comme mentionné dans l'**Exemple 4.1**, il y a une chaine optionnelle qui décrit le sens auquel se réfère la traduction. Cette chaine contient parfois un indice textuel court (glose) d'au plus quelques mots du sens, qui pour un humain est suffisant pour identifier le sens. Parfois la chaine est numérique et contient le numéro de sens correspondant à la traduction. Parfois elle contient les deux.

Ainsi dans cet exemple la glose donne comme indice *amphibian*, qui doit normalement faire référence à l'un des sens de *frog*.

Si nous examinons les définitions des sens de l'entrée lexicale pour le nom *frog* (**Exemple 4.2**), nous pouvons voir que le mot a huit sens.

Il y a en tout quatre traductions de ce mot vers le français (« grenouille », « archet », « fourchette », « traverse (poutre de) »), qui sont respectivement annotées des gloses : « *amphibian* », « *end of a string instrument's bow* », « *organ in a horse's foot* », « *part of a railway* ».

Ainsi, en tant qu'humains, nous pouvons immédiatement voir que la première traduction correspond au premier sens, la deuxième traduction au deuxième sens, la troisième traduction au cinquième sens et la dernière traduction au sixième sens, en comparant les gloses et les définitions.

Lorsque la glose contient uniquement le numéro de sens, elle peut par exemple prendre une forme ressemblant à «5» pour indiquer que la traduction correspond au sens 5.

Nous pouvons également trouver les deux à la fois. Par exemple la glose de la quatrième traduction pourrait être de la forme «*6. part of a railway*». Il n'y a toutefois pas de norme imposée ou recommandée dans la plupart des éditions de Wiktionary, ainsi la glose peut prendre l'une des nombreuses formes suivantes : «*6.*», «*[6]*», «*6. part of a railway*» ou encore *[6] part of a railway*.

Comme illustré par le Tableau 4.1, la quantité de gloses textuelles ou numériques, ou même la présence de la glose varie grandement selon les éditions. Par exemple, dans l'édition de l'italien, il n'y a jamais de gloses. L'édition anglaise ne fournit souvent que des gloses textuelles alors que l'édition allemande contient uniquement des gloses numériques. Dans le cas des éditions finnoise, française et

Exemple 4.2 **Les sens du mot anglais *frog***

La `LexicalEntry` anglaise *frog* a 8 sens, définis comme suit :

1. A small tailless amphibian of the order Anura that typically hops.
2. The part of a violin bow (or that of other similar string instruments such as the viola, cello and contrabass) located at the end held by the player, to which the horsehair is attached.
3. (Cockney rhyming slang) Road. Shorter, more common form of frog and toad.
4. The depression in the upper face of a pressed or handmade clay brick.
5. An organ on the bottom of a horse's hoof that assists in the circulation of blood.
6. The part of a railway switch or turnout where the running-rails cross (from the resemblance to the frog in a horse's hoof).
7. An oblong cloak button, covered with netted thread, and fastening into a loop instead of a button hole.
8. The loop of the scabbard of a bayonet or sword.

portugaise, beaucoup de relations de traductions fournissent à la fois des gloses numériques et textuelles.

Ainsi, dans les cas où les deux informations sont présentes, nous pouvons obtenir un jeu de données étalon de référence, où nous avons à la fois la glose textuelle que nous pourrons essayer de faire correspondre à un sens, mais aussi la bonne réponse (le numéro), ce qui permettra d'évaluer la technique proposée dans cette section.

4.1.2 Création d'un étalon de référence

Parmi l'ensemble des gloses associées aux traductions, certaines sont invalides, ou alors nécessitent un pré-traitement avant la génération de l'étalon de référence comme décrit ci-dessous.

— suppression des gloses vides, ou contenant des informations sans aucun lien avec les traductions, par exemple des notes de type «À faire», ou «traduction à vérifier».

— Normalisation du format des gloses pour pouvoir faire abstraction des différences de format telles que «*textual gloss (1)*» ou «*1. textual gloss*».

L'étalon de référence a été créé au format de la campagne d'évaluation `trec_eval`[2], car nous voulions réutiliser le programme d'évaluation, qui offre de nombreuses mesures. Le format consiste en réalité en un simple fichier texte associant à l'URL (provenant du graphe RDF de DBNary), l'indice du sens correspondant. Même si `trec_eval` évalue la correspondance entre une requête et le rang des réponses attendues, si l'on remplace la requête par l'URL de la traduction et les rangs par les indices de correspondance, le script `trec_eval` permet d'évaluer notre tâche de manière équivalente.

2. http://trec.nist.gov/trec_eval/index.html

4.1.3 Algorithme de rattachement

Le rattachement des relations de traduction au niveau des sens source quand des gloses sont disponibles se fait :

— en associant directement la traduction au bon numéro de sens quand une glose numérique est présente,
— ou en comparant la glose avec les définitions des différents sens source, jusqu'à trouver celui ou ceux qui correspondent le mieux.

Nous présenterons d'abord l'algorithme de correspondance de manière formelle, puis nous en viendrons à la mesure de similarité utilisée pour établir les correspondances entre les gloses et les définitions des sens.

4.1.3.1 Formalisation

Afin de formaliser l'algorithme, nous utiliserons autant que possible les notations définies dans le Chapitre 3.

Soit $\mathcal{T}r$ l'ensemble des relations de traduction, et $\mathcal{E}l$ l'ensemble de toutes les entrées lexicales dans une version de *DBNary* donnée.

Soit la fonction $\mathtt{Glose} : \mathcal{T}r \rightarrow \mathcal{S}b$, associant une relation de traduction à un ensemble de symboles sémantiques (**Définition** 3.5) correspondant à la glose de la relation de traduction.

Soit la fonction $\mathtt{Source} : \mathcal{T}r \rightarrow \mathcal{E}l$ qui retourne l'entrée lexicale source d'une relation de traduction, et l'association $\mathtt{Source}^* : \mathcal{T}r \mapsto \mathcal{S}$ qui retourne la ou les sens source de la relation de traduction. Nous utiliserons la fonction $\mathcal{S}$ (**Définition** 3.3) qui retourne l'ensemble des sens s_i d'une entrée lexicale $el \in \mathcal{E}l$.

Soit la fonction $\mathtt{def}(s) = \mathtt{signf}(s, '\mathtt{def}')$, qui retourne la signature sémantique d'un sens $s \in \mathcal{S}$, qui correspond à la définition du sens (**Définition** 3.5).

Soit la fonction $\mathtt{score} : \mathcal{S}b \times \mathcal{S}b \rightarrow [0,1] \in \mathbb{R}^+$, qui renvoie un score positif représentant la proximité sémantique entre deux signatures sémantiques de même type.

Nous pouvons ainsi, en partant d'une entrée lexicale source el_s définir le processus de rattachement des traductions aux sens source ainsi :

$$\forall t_i \in \mathcal{T}r(el_s), \mathtt{Source}^*(t_i) \leftarrow \underset{s_i \in \mathcal{S}(el_s)}{\arg\max}\{\mathtt{score}(\mathtt{glose}(t_i), \mathtt{def}(s_i))\}$$

Cette approche assez élémentaire, a le désavantage de n'assigner la traduction qu'à une unique source, alors que dans beaucoup de cas, la traduction peut correspondre à plusieurs sens (par exemple, entre l'anglais et le français les mots *night* et nuit, la plupart des sens se traduisent de la même manière).

La solution adoptée par Christian M. Meyer et Gurevych (2012b) est l'utilisation d'une valeur seuil k comme décrit, dans la Section 2.1.2.1. Cependant, comme ici nous choisissons d'utiliser une ou des mesures de similarité normalisées entre 0 et 1, nous adoptons une approche un peu différente qui consiste à utiliser un intervalle de scores de $+/-\delta$.

Ainsi tous les sens dont les scores avec la glose de la traduction sont à $+/-\delta$ du score maximal seront sélectionnées comme étant une source de la relation de traduction (la relation sera démultipliée pour chaque source différente). Cette approche a cependant comme désavantage de toujours assigner un sens à une

traduction même si tous les scores sont proches de zéro, ce qui serait indicatif d'une traduction incorrectement assignée dans Wiktionary. Cependant nous ne cherchons pas ici à traiter ce problème, qui pourra être remédié par un filtrage a *posteriori*.

Ainsi, nous pouvons intégrer l'utilisation du delta, en rajoutant une condition au calcul du arg max :

$$\begin{aligned} M_s = \max_{s_i \in \mathcal{S}(el_s)} \{score(glose(t_i), def(s_i))\} : \\ \arg\max_{s_i \in \mathcal{S}(el_s)} \{score(glose(t_i), def(s_i))\} \\ = \{s_i | M_s > score(glose(t_i), def(s_i)) > M_s - \delta\} \end{aligned}$$

4.1.3.2 Mesure de similarité

Afin de calculer un score entre les gloses, nous devons construire une mesure de similarité sémantique qui puisse comparer les gloses aux définitions des sens.

La mesure de Lesk (1986) est l'une des plus employées (recouvrement de gloses) quand il s'agit de comparer des définitions de sens pour calculer une similarité sémantique entre des définitions.

Cependant cette mesure présente certaines limites, qui nécessitent une adaptation de la mesure (Tchechmedjiev, 2012).

- Si les tailles des gloses ou définitions comparées diffèrent, comparer des définitions plus longues mènera mécaniquement à des valeurs de similarité plus grandes.
- Les mots exacts présents dans les définitions sont importants, il suffit qu'un mot critique manque pour perdre toute similarité alors que les sens sont en fait pratiquement identiques. Cela serait le cas si deux définitions sont des paraphrases l'une de l'autre et qu'il n'y a aucun mot en commun.
- La mesure de Lesk n'est pas une similarité métrique mais une mesure se rapprochant plus d'une vraisemblance. La normalisation de la mesure n'est pas trivial, et un choix de normalisation doit être fait en fonction des applications.

Il existe d'autre mesures similaires à Lesk qui elles sont normalisées d'une manière ou d'une autre, par exemple le coefficient de Dice, les indices de Tatimono ou de Jaccard. Toutes ces mesures sont en fait des cas particuliers de l'indice de Tversky (1977b). Cette mesure prend en compte à la fois les similarités, mais aussi les dissimilarités entre les définitions. Dans sa formulation de base elle s'écrit, pour deux sens s_1, s_2 dont les définitions sont $d_1 = def(s_1), d_2 = def(s_2)$:

$$Score(s_1, s_2) = |d_1 \cap d_2| + \alpha|d_1 - d_2| + \beta|d_2 - d_1|$$

Cette notation n'est cependant pas normalisée et ne correspond pas à un cas général des mesures citées ci-dessus. Ainsi, Pirrò et Euzenat, 2010 proposent de l'exprimer sous forme de quotient, mais aussi remplacent l'opérateur de cardinalité d'ensemble par une fonction F quelconque :

$$Score(s_1, s_2) = \frac{F(d_1 \cap d_2)}{F(d_1 \cap d_2) + \alpha F(d_1 - d_2) + \beta F(d_2 - d_1)}$$

Selon les valeurs de α et de β, l'indice de Tverski ressemble à différentes mesures utilisées en recherche d'information. Avec $(\alpha = \beta = 0.5)$ il est équivalent à

l'indice de Dice , et pour ($\alpha = \beta = 1$) à l'indice de Tatimono. Lesk lui corresponds à $\alpha = \beta = 0$ dans l'expression originale (pas le quotient).

De manière générale, α et β correspondent à l'importance accordée à la différence d'un ensemble par rapport à l'autre, et vice versa. Du fait de l'asymétrie inhérente à l'indice de Tverski, même sous sa forme de quotient, elle n'est pas une métrique au sens mathématique du terme (car non symétrique et ne satisfaisant pas l'inégalité triangulaire).

Il existe cependant des variantes symétriques (Jimenez, Gonzalez et Gelbukh, 2010). Ici, du fait de l'asymétrie de longueur entre les gloses des relations de traduction et les définitions, nous optons pour l'utilisation de la version non symétrique. Comme travaillons sur de nombreuses langues, cette approche par recouvrement est pertinente, car si nous devions utiliser des ressources supplémentaires pour le calcul des similarités, il faudrait avoir une ressource dans chacune des langues traitées.

4.1.3.3 Extension par correspondance douce

Un problème qui se posera cependant est que les définitions et les gloses de traduction ne sont pas lémmatisées, ainsi une comparaison exacte des mots ne fonctionnera pas bien, en particulier pour certaines langues : Les langues agglutinantes comme l'allemand et le finnois, les langues avec une morphologie complexe (russie, finnois, latin, etc.), les langues sans vraie segmentation en mots (japonais, chinois). Ainsi il faudra un moyen de comparer des mots de manière partielle, même si ils ont accompagnés de suffixes ou de préfixes différents, à défaut de pouvoir construire des racinisateurs et des analyseurs de surface pour l'ensemble des langues qui seront traitées.

Lors du calcul de Lesk, lorsque deux mots des définitions sont identiques, le score de similarité est incrémenté de 1, si ils ne le sont pas (cardinalité de l'intersection des deux ensembles), le score est incrémenté de 0. Ainsi, au lieu d'ajouter 1 ou 0, nous pourrons ajouter une valeur décimale entre 0 et 1 qui caractériserait la proximité des deux mots, par exemple par l'usage d'une technique de correspondance de chaîne. Cette technique se base en fait sur le principe de cardinalité douce proposé par Jimenez, Becerra et Gelbukh, 2012. Ainsi dans la formule de Tverski-quotient présentée plus tôt, il faut remplacer F par la cardinalité douce, où ρ est un paramètre indiquant le degré de *douceur* et où strsim est une fonction de similarité de chaine :

$$F(A) = \sum_{A_i \in A} \left(\sum_{A_j \in A} \mathrm{strsim}(A_i, A_j)^{\rho} \right)^{-1}$$

Il existe de nombreuses mesures de similarité de chaîne, et il est difficile d'en choisir une *a priori*. Dans la partie expérimentale qui suit, nous estimerons à la fois la meilleure similarité de chaine à utiliser ainsi que les paramètres optimaux pour la mesure de Tverski mais aussi pour l'algorithme d'alignement.

4.1.4 Adaptation expérimentale et validation

Dans cette section, nous allons appliquer l'algorithme de rattachement de manière pratique dans DBNary. Pour ce faire nous allons d'abord sélectionner les

éditions de langues sur lesquelles nous allons travailler pour l'évaluation, puis estimer les paramètres optimaux et enfin passer à l'évaluation du rattachement en lui même grâce à l'étalon de référence.

Nous nous intéressons ainsi uniquement aux éditions de langues où un étalon de référence peut être généré, c'est-à-dire là où il y a à la fois une glose textuelle et un numéro de sens. Nous ajoutons une contrainte supplémentaire sur la taille de l'étalon de référence, qui doit faire plus de dix mille traductions avec une glose textuelle et numérique à la fois, afin que la comparaison soit suffisamment significative.

Ainsi, les éditions répondant aux critères sont les éditions française, portugaise, finnoise (voir Tableau 4.1).

4.1.4.1 Mesures d'évaluation

Pour l'évaluation, nous utilisons les mesures de correspondance standards utilisées en WSD et en recherche d'information, c'est-à-dire le rappel, la précision, et le score F_1, où, $P = \frac{|\{Corrects\} \cap \{Annotés\}|}{|\{Annotés\}|}$, $R = \frac{|\{Correct\} \cap \{Annotés\}|}{|\{Attendus\}|}$, et $F1 = \frac{2 \cdot P \cdot R}{P+R}$, la moyenne harmonique de R et de P. Dans l'expérience, ce qui nous intéresse est d'optimiser à la fois la précision et le rappel, ainsi, nous présenterons uniquement le score F_1.

4.1.4.2 Réglage des paramètres de la mesure de similarité

Comme il a été mentionné précédemment, il y a un certain nombre de paramètres de la mesure de similarité à estimer avant de réaliser l'alignement en lui même : le choix de la mesure de similarité de chaîne, le δ pour l'intervalle de correspondance, α et β pour l'importance relative des différences entre l'une ou l'autre des gloses/définitions.

Nous faisons l'hypothèse que δ est indépendant du reste des paramètres. Nous commençons donc en fixant $\delta = 0$, ce qui signifie que l'algorithme assignera un unique sens par traduction, afin d'estimer les autres paramètres d'abord. Par ailleurs, la similarité de chaîne est utilisée uniquement pour le calcul du recouvrement entre les ensembles, c'est-à-dire en amont du calcul global de l'indice de Tverski. Ainsi, sont choix est indépendant des paramètres α et β. Ainsi nous estimerons d'abord la mesure qui capture de manière optimale la similarité entre deux gloses/définitions avec $\alpha = \beta = 0.5$ (ce qui correspond au coefficient de Dice).

Pour trouver la similarité de chaîne optimale, nous allons évaluer les meilleures mesure parmi celles recensées par William W. Cohen, Ravikumar et Stephen E. Fienberg (2003).

Nous utiliserons : Jaro-Winkler, Monge-Elkan, la distance de Levershtein normalisée et la mesure de la sous-chaine commune la plus longue. Nous utiliserons comme référence l'indice de Tverski avec une cardinalité «dure».

Nous utiliserons les notations suivantes pour se référer aux différentes mesures :

- Indice de Tverski, – Ts,
- Jaro-Winkler – JW,
- Monge-Elkan – ME,
- Levenshtein Normalisé – Ls,
- Sous-chaine commune la plus longue – Lcss,

Système	français	portugais	finnois
FTiJW	78,53%	80,79%	94,79%
FTiLcss	77,78%	76,97%	94,95%
FTiLs	**78,61%**	**81,76%**	**95,36%**
FTiME	76,84%	76,83%	94,95%
Ti	70,88%	71,71%	88,06%

TABLE 4.2 – Résultats comparés du score F1 pour le français, finnois et portugais (meilleur résultat en gras).

— F – Flou/doux.

Par exemple, l'indice de Tverski avec une cardinalité «dure» sera nommé "Ti", alors que l'indice de Tverski «doux» qui utilise la mesure de distance de chaine de Monge-Elkan sera nommée "FTiME".

Tous les paramètres sont estimés sur un sous ensemble de 15 000 alignements de référence, soit 50% de l'alignement de référence pour la langue avec l'étalon de référence le plus petit, soit le français avec 28 000 alignements de référence. Avec un jeu de données si important, même des différences infimes d'un dixième de pourcent sont significatives.

La Tableau 4.2 présente les résultats pour chaque mesure de distance de chaîne pour chacune des langues utilisées pour l'évaluation (français (Fr), finnois (Fi), portugais (Pt)). La mesure qui offre les meilleurs résultats (score F_1) de manière consistante sur les trois langues est la mesure de Levenshtein normalisée, avec un score plus haut de $+1\%$ à $+1.96\%$.

Maintenant que nous avons déterminé la mesure de similarité de chaîne optimale, nous allons estimer les valeurs d'α et de β. Ici, nous fixons les valeurs pour qu'elles soient complémentaires, afin de garantir que le résultat de l'indice de Tverski demeure dans l'intervalle $[0, 1]$.

Du fait que les gloses sont courtes (souvent un mot unique) et que les définitions sont nettement plus longues, nous pouvons faire l'hypothèse que la valeur optimale se situe aux alentours de $\alpha = 1 - \beta \approxeq 0.1$. En effet cette valeur permettrait d'atténuer le biais de la différence de longueur. Pour cette expérience, nous avons fait varier la valeur de $\alpha = 1 - \beta$ par pas de 0.1. La Figure 4.2 présente le résultat de l'estimation du paramètre : le score F_1 pour chaque valeur de $\alpha = 1 - \beta$ pour toutes les trois langues.

Nous observons ainsi que la meilleure valeur du paramètre correspond à $\alpha = 1 - \beta = 0.1$ pour laquelle le score F_1 est meilleur de $+0.15\%$ à $+0.43\%$ par rapport à la deuxième meilleure valeur (selon les langues).

Maintenant, nous pouvons fixer la valeur de α et β à la valeur optimale afin de trouver la valeur optimale de δ. Nous faisons varier δ par palier de 0.05 entre 0 et 0.3. Le choix de la borne supérieure se fait sur l'hypothèse que la valeur optimale est probablement plus proche de zéro, car une valeur trop grande signifierait que les traductions seraient assignées à tous les sens, ce qui irait à l'encontre de l'objectif même de l'algorithme.

La Figure 4.3 présente le score F_1 pour chaque valeur de $delta$ et pour chaque langue. Pour chaque langue, la valeur de delta influence peu le score, même si pour les trois langues, $\delta = 0.1$ semble très légèrement meilleur. Cela laisse à penser que dans la plupart des cas, la mesure de similarité capture un sens unique bien

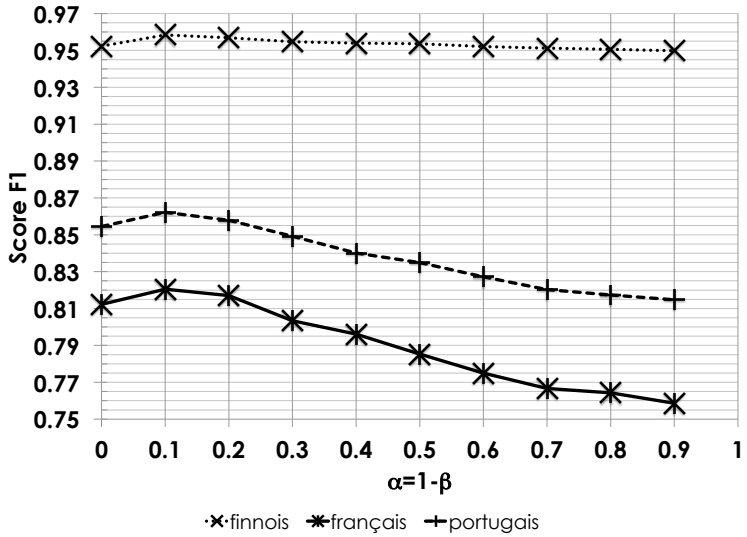


FIGURE 4.2 – Score F1 pour le français, portugais et finnois pour l'estimation de α et β.

distinct des autres, et que à ce niveau le delta n'a d'influence réelle que sur un très petit nombre de cas.

4.1.4.3 Évaluation de la mesure

Les paramètres optimaux étant estimés, nous pouvons maintenant présenter les résultats finals (Tableau 4.3), et les comparer aux résultats de référence, c'est-à-dire, l'affectation des traductions aux sens les plus fréquents, et l'affectation aléatoire. Ces résultats ont été calculés sur 15 000 autres instances issues des données d'évaluation strictement distinctes de celles utilisées lors de l'estimation des paramètres.

La première chose que l'on peut remarquer est que les résultats pour le finnois sont bien meilleurs que pour les deux autres langues de l'ordre de $\approx 10\%$. Cependant, si nous regardons les références du premier sens et de l'affectation aléatoire, nous pouvons nous apercevoir qu'elles sont plus hautes que pour les deux autres langues de l'ordre de $\approx 50\%$. Ainsi, nous pouvons en conclure que les entrées lexicales de l'édition finnoise sont très peu polysémiques et que le premier sens est souvent la traduction correcte.

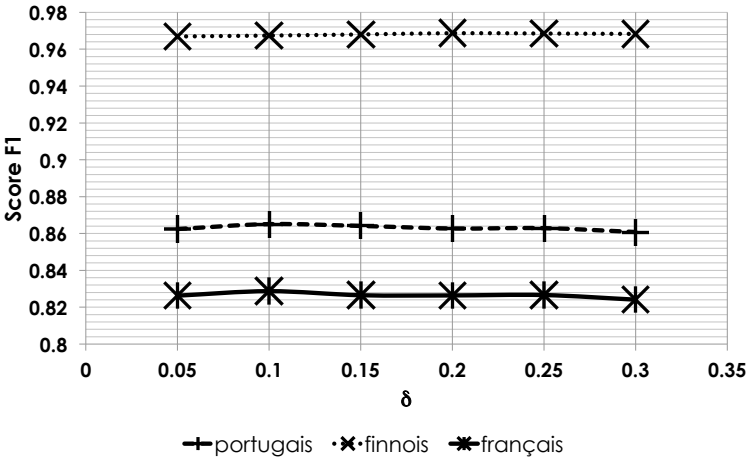


FIGURE 4.3 – Résultats pour l'estimation de δ.

Langue	P	R	F1	Pr. Sens. F1	Aléatoire F1
Portugais	85,72%	88,14%	86,51%	23,97%	31,03%
Finnois	96,42%	97,77%	96,87%	72,18%	79,62%
Français	82,67%	83,13%	82,63%	35,42%	37,67%

TABLE 4.3 – Résultats finaux avec les paramètres optimaux : Précision, Rappel, Score F1 pour les trois langues comparé aux références d'affectation au premier sens et aléatoire.

Par contre pour les autres langues, nous pouvons observer deux choses intéressantes. Premièrement, la référence par affectation aléatoire est très basse (de l'ordre de $\approx$ 30%), ce qui suggère qu'il y a en moyenne beaucoup de sens pour chaque entrée lexicale. Par contre on observe aussi que la référence de l'affectation aléatoire est supérieure à la référence d'affectation au premier sens, ce qui suggère que les sens ne sont pas ordonnés d'une manière arbitraire indépendamment de leur fréquence d'usage.

Par rapport à l'état de l'art, les scores F_1 obtenus lors de l'alignement sont plutôt bons, notamment par rapport à Christian M. MEYER et GUREVYCH (2012b), qui obtiennent entre 84% et 85% de précision pour une référence d'alignement au premier sens de 79% (indication sur le degré de polysémie) pour l'édition allemande. Ainsi d'après le score de référence au premier sens, on peut faire l'hypothèse que le degré de polysémie est à peu près le même que dans notre expérience avec le finnois, où nous avons obtenu un score F_1 meilleur de 5% à 6%.

En plus de cette évaluation quantitative, nous pouvons aisément identifier les erreurs en comparant les scores obtenus pour le recouvrement avec les gloses. Nous présentons ainsi les cas d'erreurs typiques dans l'alignement produit et donnons quelques raisons possibles pour ces erreurs.

4.1.4.4 Analyse des erreurs typiques

Nous n'avons pas fait d'analyse exhaustive des cas d'erreurs, mais nous avons parcouru les erreurs en notant manuellement quels types d'erreurs revenaient souvent. Ainsi, nous avons recensés les cas suivants.

1. Aucun recouvrement entre la glose et les définitions, ce qui a mené à un choix arbitraire de l'algorithme. Ce cas de figure se présente dans des cas où la glose de la traduction est une paraphrase ou un indice métaphorique, qui n'ont donc aucun mot en commun avec les définitions correspondantes.
2. Parfois la glose est un nom de domaine, qui identifie le sens unique correspondant à ce domaine. Or, l'extracteur DBNary ne permet pas à l'heure actuelle d'extraire les étiquettes de domaine, ce qui mène également à une absence de recouvrement.
3. Il arrive également que des nouveaux sens aient été ajoutés ou supprimés pour l'entrée lexicale en question, et que le numéro de sens ne corresponde plus avec le vrai sens correspondant, ce qui induit, cette fois une erreur dans le jeu de données d'évaluation.

Une amélioration de l'extracteur de DBNary permettrait de supprimer le deuxième type d'erreur et les erreurs du troisième type pourraient être corrigées manuellement dans les différentes éditions, car elle ne représentent au plus qu'une

une centaine d'entrées par édition. Le seul type d'erreur qui peut être mitigé est le premier type. Cette mitigation passe par un enrichissement des gloses et des définitions avec des mots proches sémantiquement et pourrait aider à outrepasser cette limitation en augmentant les chances qu'il y ait une correspondance pertinente.

4.1.4.5 Conclusion

Même si l'alignement produit n'est pas parfait, nous obtenons une qualité d'alignement très comparable à l'état de l'art. Par ailleurs, ce résultat est améliorable par des modifications simples du processus d'extraction, indépendamment de l'algorithme d'alignement. À l'issue de l'expérience, l'alignement a été calculé sur l'ensemble de la ressource, partout où il y avait une glose textuelle. Pour les gloses contenant des numéros de sens, les traductions ont été alignées directement aux bons sens. Les données issues de l'expérience ont été mises à disposition sur le site de *DBNary* comme des jeux de données distincts pour chaque langue et distincts des jeux de données principaux de DBNary contenant le résultat de l'extraction.

Bien entendu, cette extraction n'est pas parfaite, mais elle se rapproche tout de même de ce que des annotateurs humains produiraient comme le montrent Christian M. Meyer et Gurevych (2012b) expérimentalement. Par ailleurs, cet alignement se limite aux traductions possédant une glose. Pour l'anglais et le finnois, l'allemand, le japonais, le turc la majorité des traductions ont une glose et auront donc un alignement. Cependant le français, le grec, le portugais et beaucoup des autres éditions de langues extraites n'ont des gloses que pour une partie des traductions ($\approx$ 25% pour le français, $\approx$ 8% pour le grec, $\approx$ 40% pour le portugais, 0% pour l'italien).

Ainsi, le reste des relations de traductions relèveront de l'utilisation d'algorithmes d'AAS pour leur alignement de manière générale que ce soit au sens cible ou au sens source. Cependant l'alignement partiel obtenu ici a l'avantage d'être produit avec une précision supérieure à ce que produiraient des algorithmes d'AAS. L'état de la ressource après cette étape est montré par la Figure 4.4, où les sources des relations de traduction, quand c'est possible, sont rattachées aux sens. Nous pouvons voir que le nombre de combinaisons possibles pour l'alignement complet est fortement réduit là où les sources sont rattachées aux sens. Plus précisément, entre deux entrées lexicales, ayant respectivement n et m sens, on passe de $n \times m$ combinaisons possibles à m combinaisons possibles.

4.2 Alignements sur les sens cibles et le sens source sans glose de traduction

De manière générale l'alignement au niveau des cibles des relations traductions nécessite l'utilisation d'approches d'AAS et donc de mesures de similarité translingues. Cependant un sous-ensemble pourra être aligné en exploitant les sous-éditions de langue de Wiktionary. Nous présenterons d'abord l'alignement au travers de l'utilisation de ces sous éditions, puis nous nous intéresserons aux mesures de similarité translingues applicables à *DBNary*, puis leur utilisation dans le cadre d'un processus d'AAS général. Enfin, nous présenterons les manières possibles d'évaluer les alignements par acceptions, que ce soit les manières intrinsèques ou *in vitro* et les techniques extrinsèques ou *in vivo*.

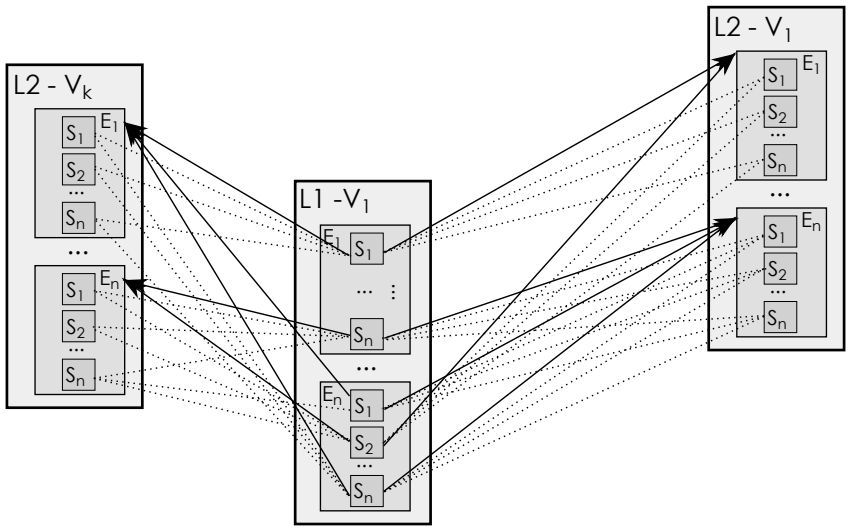


FIGURE 4.4 – Les données de *DBNary* avec rattachement des sources de relation de traduction. Les traits en pointillés représentent les alignements cibles possibles.

4.2.1 Utilisation des sous-éditions de Wiktionary

Dans une édition de Wiktionary dans une langue donnée *l* (par exemple l'anglais), il y aura également des entrées lexicales dans d'autre langues dont motforme est le même que le vocable, qui donnent potentiellement les distinctions de sens de la langue étrangère avec des description dans la langue *l* de l'édition.

Par exemple, sur la page anglaise *chat* (discuter) il y a également des informations sur le mot français chat, avec une liste abrégée de sens et des gloses anglaises indiquant un ensemble de mots anglais utilisés dans le même sens ou quelques indications supplémentaires qui précisent le sens (Figure 4.5).

FIGURE 4.5 – Un exemple de données des sous éditions de langues de l'édition anglaise de Wiktionary. Les distinctions de sens avec les gloses dans la langue de l'édition sont entourées en jaune doré.

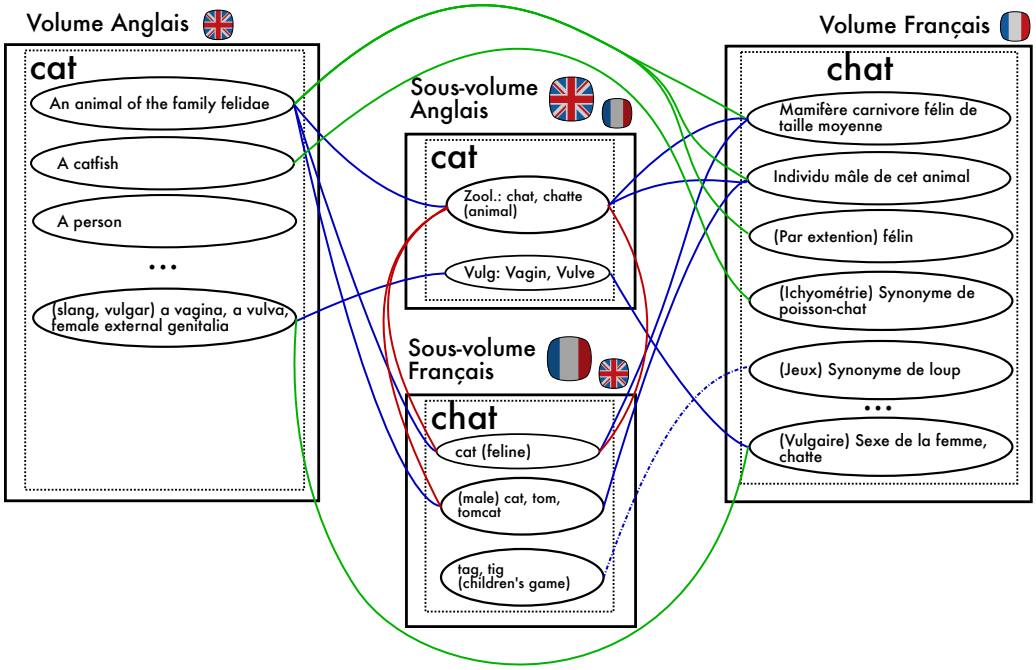


FIGURE 4.6 – Exemple de données de sous-éditions de langue, pour le mot français *chat* et l'entrée *chat* de la sous-édition française de l'édition anglaise et pour le mot *cat* de l'édition anglaise et l'entrée *cat* de la sous édition anglaise de l'édition française, ainsi que les alignements potentiels.

Les sous éditions sont extraites depuis peu par l'extracteur *DBNary*. Ainsi, lorsque ces données sont présentes, nous pouvons restreindre les sens cible ou source candidats à ceux qui se trouvent dans les sous éditions respectives, afin de rendre l'alignement plus aisé d'un point de vue combinatoire. Par ailleurs, les gloses de la sous édition en langue cible dans l'édition en langue source correspondent mieux à la formulation des définitions dans l'édition cible par rapport aux définitions des sens source (et vice versa).

Observons maintenant sur un exemple précis (Figure 4.6) le type d'informations disponible afin de déterminer de manière exacte les informations qui pourraient être utilisées pour l'alignement des éditions de DBNary.

Ainsi, nous pouvons voir que les sous éditions de langues ne sont pas exhaustives, et que souvent, uniquement les sens les plus courants sont présents. Par ailleurs, malgré la présence d'un sens partagé entre le français et l'anglais et la présence du dit sens dans la sous-édition anglaise du français ne garantit pas la présence du même sens dans la sous édition française de l'anglais.

Une autre observation possible, est que les distinctions de sens on des granularités différentes entre les éditions principales et le sous-éditions. C'est un aspect intéressant dans le sens où c'est une information supplémentaire pour conforter la validité ou non d'une acception particulière, en vérifiant que l'on retrouve la même chose au niveau des sous éditions.

De plus, les gloses des sous éditions contiennent des hyperliens vers d'autres mots, par exemple pour le dernier sens dans la sous édition anglaise de l'édition française, tag et tig, font référence aux mots anglais correspondants, ce qui s'apparente à une relation de traduction. Cela pourrait sembler redondant avec les liens de traductions des éditions principales, cependant dans ce même exemple, l'édi-

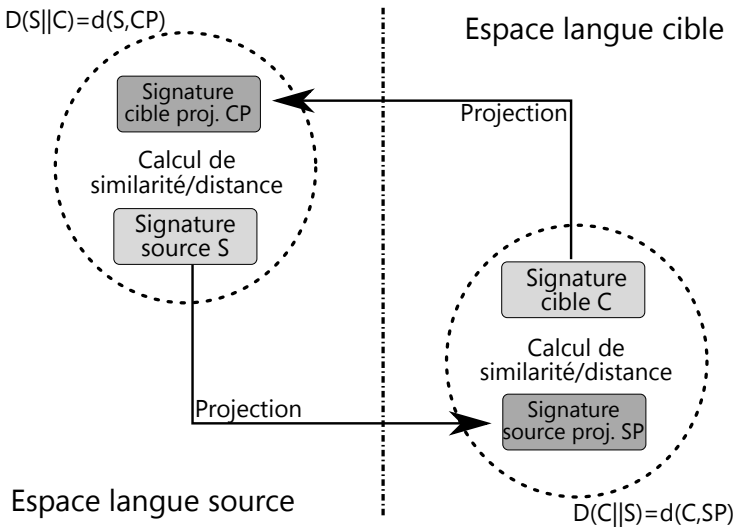


FIGURE 4.7 – Procédure de calcul d'une similarité/distance translingue.

tion française ne contient pas de traduction associée à chat au sens du jeu. Ainsi il serait possible d'utiliser ces gloses pour augmenter la couverture des relations de traduction.

En contrepartie, les gloses des sous éditions ne semblent pas toujours suffisantes pour établir une correspondance avec les définitions des éditions principales, sans recourir à une étape préalable d'enrichissement.

4.2.2 Mesures de similarité translingues

Nous avons présenté dans la Section 2.1 en même temps que l'AAS un certain nombre d'approches pour calculer la similarité sémantique translingue, qui en théorie pourraient toutes être utilisées avec DBNary, afin d'aligner les différentes éditions. Cependant, les spécificités de Wiktionary font que certaines approches seront plus pertinentes que d'autres.

Par exemple, dans DBNary, le nombre et la qualité des relations sémantiques est très variable, sans compter que comme les relations de traduction, les relations sémantiques relient les entrées lexicales à des vocables et non directement à des sens, ce qui est assez problématique pour l'application d'approches utilisant la topologie du graphe.

De manière générale, même en exploitant les informations des sous éditions, il est nécessaire de comparer des signatures sémantiques exprimées avec des représentations non comparables directement, c'est-à-dire typiquement des définitions ou des gloses dans des langues différentes potentiellement très éloignées sémantiquement. Il existe de nombreuses méthodes de mesure de similarité translingue, nous en retenons cependant trois familles qu'il est possible d'utiliser dans le cadre de *DBNary*.

Plus particulièrement, il s'agit de projeter les symboles de la signature sémantique dans un espace où ils sont comparables malgré que les langues soient différentes. Parmi les trois approches mentionnées, il n'y en a pas une qui soit formellement meilleure que l'autre. Tout dépend des ressources disponibles pour chaque paire de langues ; une combinaison de différentes approches pouvant potentiellement mener à de meilleurs résultats (en fonction des caractéristiques de chaque ressource).

La Figure 4.7 résume de manière générale le processus, dans le cas où les espaces de le langue cible et source sont distincts, c'est-à-dire qu'ils ne sont pas directement comparables. L'approche présentée ici projette l'une des signatures soit dans la langue source, soit dans la langue cible, puis calcule une similarité entre la signature d'un des sens et la projection de la signature de l'autre sens dans la langue du premier. Lorsque les signatures sémantiques sont générées à partir de définitions, le processus est asymétrique de manière inhérente, car les signatures des deux côtés étant différentes, leurs projections le sont également.

Ainsi, en reprenant les notations utilisées dans la Section 3.4.1.2, il est possible de voir le calcul de la similarité ou de la distance comme une mesure de similarité asymétrique, ou comme une divergence. Ainsi si nous notons sg_S la signature source, sg_C la signature cible sg_{SC} la signature source projetée dans la langue cible et sg_{CS} la signature cible projetée dans la langue source, nous pouvons exprimer la divergence/similarité asymétrique comme $D(sg_S \parallel sg_C) = d(sg_S, sg_{CS})$ et $D(C \parallel S) = d(sg_{SC}, sg_C)$, où d est une mesure de distance. Par extension nous pourrions noter une similarité asymétrique avec la même notation : $S(\cdot \parallel \cdot)$.

4.2.2.1 À base de traduction automatique

Une solution simple pour produire des définitions comparables entre plusieures langues est d'utiliser un système de traduction automatique qui viendra traduire la définition du sens source en langue cible ou alors la définition du sens cible dans la langue source. Une fois que les définitions sont traduites dans la même langue, nous pouvons utiliser une mesure de similarité monolingue à base de recouvrement. C'est une technique qui a été utilisée en alignement d'ontologies avec des précisions d'alignement supérieures comparé à une approche basée sur des ressources lexico-sémantiques (Fu, Brennan et O'Sullivan, 2009 ; Lesnikova, David et Euzenat, 2015). C'est également l'approche utilisée pour la construction de BabelNet pour produire les définitions multilingues des BabelSynsets pour des langues où il n'est pas possible de les obtenir autrement.

La qualité du résultat de la traduction automatique n'est bien sûr pas parfaite, et certaines erreurs produites par la traduction automatique peuvent impacter la qualité de la mesure de similarité finale.

Le plus problématique sont les erreurs de choix lexical lors de la phase de décodage du système de traduction automatique. Une partie de ces erreurs de choix lexical viennent du fait que le sens de la phrase de départ a été mal identifié et que la traduction obtenue du mot en question est incorrecte dans la langue cible du fait d'une lexicalisation divergente.

Un autre type de problème de choix lexical vient non du fait que le sens au niveau de la source à mal été identifié, mais qu'un autre mot synonyme est utilisé par rapport à l'autre définition.

Alors que le second problème peut être largement résolu en enrichissant les symboles de synonymes, le premier type d'erreur ne peut être corrigé si facilement car même étendre la définition n'aidera pas à trouver un recouvrement puisque les synonymes du mot-forme incorrectement lexicalisé n'ont pas plus de raisons de correspondre.

L'utilisation de la traduction automatique dans ce sens est une utilisation indirecte du modèle de traduction de mot combinée à la désambiguïsation implicite de la phase de décodage des systèmes de traduction automatique statistique. Vis-à-vis de la mesure de similarité, si elle est calculée uniquement sur le recouvrement des symboles des signatures sémantiques, il n'est pas essentiel d'utiliser un

système de traduction complet, en particulier si l'ordre des mots n'est pas pris en compte.

4.2.2.2 À base de ressources lexicales

Ainsi, il est possible d'utiliser une ressource lexicale, et plus particulièrement un graphe de traduction pour produire la projection, du moment où nous disposons d'un système de désambiguïsation lexicale et que nous savons à quel sens correspondent les sources des traductions, comme c'est le cas dans DBNary après application de l'expérience de la Section 4.1. Cette approche est très proche de la manière dont les techniques de désambiguïsation lexicale sont appliquées aux tâches de Désambiguïsation lexicale dite translingues (*cross-lingual* WSD), qui sont en réalité des tâche de traduction d'un mot particulier dans une autre langue en connaissant le contexte source (Lefever et Hoste, 2010).

Ainsi, en partant de la définition, il faut d'abord appliquer un processus de désambiguïsation lexicale des définitions des sens source avec les sens de la ressource contenant le graphe de traduction, puis pour chaque mot annoté, le remplacer par la traduction correspondant à ce sens là si elle existe. Si jamais, il y a plusieurs traductions possibles pour un lexème de la définition, les traductions sont toutes ajoutées[3].

Cette approche a cependant certaines limitations. Tout d'abord qu'il faut disposer d'un bon système de désambiguïsation pour la langue source, ce qui implique également d'avoir un lemmatiseur et un annotateur en parties du discours pour la langue source, ce qui limite l'application de cette technique à des langues suffisamment dotées.

Ainsi, le processus de «traduction» se déroule comme suit. (Figure 4.8) :

1. Lemmatisation et annotation en parties du discours de la signature du sens source (définition) S.
2. Désambiguïsation lexicale de S.
3. Récupération dans un graphe de traduction des traductions dans la langue cible pour le sens annotés dans S.
4. Obtention de la signature source S projetée dans la langue cible SC.

Le processus sera exactement le même dans l'autre sens, c'est-à-dire partir de C pour obtenir CS.

Sur l'exemple même présenté dans la Figure 4.8, la limitation majeure déjà évoquée pour la projection par traduction automatique, est apparente ici.

Les définitions sont équivalentes, mais la manière dont elles sont formulées et les mots utilisés sont majoritairement différents. Ce qui ne permettra pas de calculer une similarité par recouvrement. Ainsi il y a un besoin réel d'enrichir les définitions, que ce soit au niveau de la cible ou de la source.

4.2.2.3 À base d'espaces vectoriels sémantiques multilingues

Il est également possible de calculer des similarités translingues en utilisant un modèle de langue à base de plongements multilingues : par la construction d'un espace vectoriel sémantique où les vecteurs sont construits à partir de données bilingues ou multilingues. Ces approches sont devenues extrêmement populaires depuis 2014, si l'on en croit les actes des conférences principales du domaine

3. Cette situation arrive pour des mots pour lesquels les sources des relations de traduction ne sont pas rattachées au sens

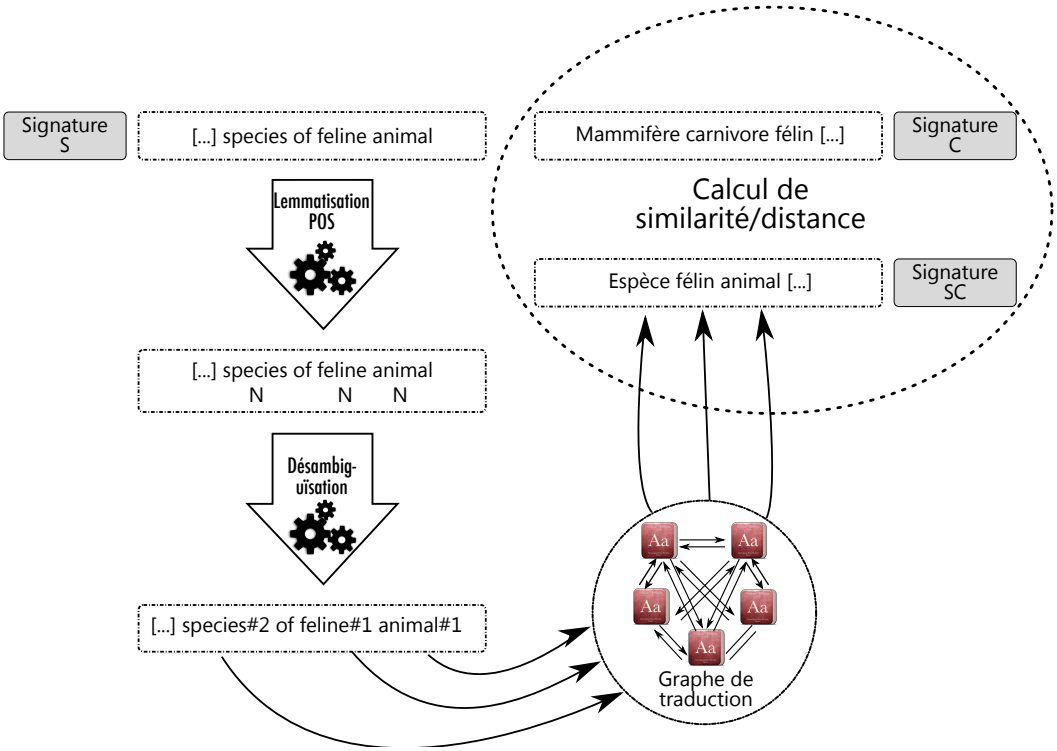


FIGURE 4.8 – Procédure de calcul d'une similarité/distance translingue par projection par une ressource lexicale multilingue.

mais aussi les dernières additions dans arXiV que ce soit à base de corpus parallèles/comparables (AMMAR et al., 2016 ; BÉRARD et al., 2016 ; CHANDAR A P et al., 2014 ; GOUWS, BENGIO et CORRADO, 2015 ; GOUWS et SØGAARD, 2015 ; ŠUSTER, TITOV et NOORD, 2016 ; VULIĆ et MOENS, 2015a) ou à base de RLS multilingues existantes telles que BabelNet (CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI, 2015 ; IACOBACCI, T. M. PILEHVAR et Roberto NAVIGLI, 2015).

La plupart des approches vectorielles produisent ce que l'on appelle des plongements de mots (*word embeddings*), c'est-à-dire qu'il y a une unique vecteur par mot du lexique, sans distinction particulière entre les sens de mots, comme celles citées ci-dessus. Quand aux plongements basées sur une RLS particulière, il s'agit souvent de plongements de sens, où l'on calcule un vecteur par sens de la RLS. Il est possible soit de construire des plongements pour chaque sens directement où alors adapter des plongements de mots existants pour qu'ils deviennent des plongements de sens pour cette ressource. Un autre usage des plongements de sens (notamment calculés par **D**écomposition en **V**aleurs **P**ropres (DVP) ou par **F**actorisation de **M**atrices **N**on **N**égatives (FMNN)) à partir de corpus est d'identifier les sens de mots de manière non supervisée en réalisant un regroupement des vecteurs de co-occurrence de mots dans chaque contexte d'apparition, c'est comme mentionné dans l'état de l'art l'induction de sens de mots (WSI). Nous ne traiterons cependant pas de ces approches ici.

Une approche simple pour produire des plongements de sens pour une RLS est d'utiliser un corpus annoté en sens comme corpus d'entrée pour l'entrainement de plongements sémantiques où les mots sont remplacés par leurs annotations sémantiques. Comme la disponibilité de ce type de corpus est limitée à quelques petits corpus et en anglais uniquement, une meilleure approche est d'utiliser un

bon système de WSD afin d'annoter un grand corpus avec la ressource afin d'en suite entraîner les plongements de sens, comme le proposent IACOBACCI, T. M. PILEHVAR et Roberto NAVIGLI (2015). Cette approche ne produit pas de plongements multilingues par elle-même, mais quand appliquée sur une ressource interlingue alignée par pivots, les vecteurs correspondant aux synsets sont communs à toutes les langues alignées dessus, comme c'est le cas avec BabelNet. L'espace des plongements de sens multilingues produit souffre bien entendu des mêmes problèmes de contrastes artificiels que la ressource sur laquelle l'espace est projeté.

Un autre approche serait de construire un corpus spécifique pour chaque sens/-concept dans la ressource en question, tel que le corpus corresponde aux usage du mot dans ce sens particulier-là. C'est l'approche retenue par CAMACHO-COLLADOS, T. M. PILEHVAR et Roberto NAVIGLI (2015) au travers de l'utilisation de la spécificité lexicale et du renforcement des vecteurs de sens a posteriori avec les relations lexico-sémantiques du réseau.

La première approche pourrait être appliquée à DBNary, mais l'inconsistance de la ressource et la brièveté des définitions poseraient problème pour obtenir un système de WSD performant. Par ailleurs le fait qu'il n'y a pas de pivot déjà existant signifie que les espaces produits ne seront pas multilingues. Aligner DBNary avec BabelNet pourrait permettre d'exploiter la représentation multilingue qui lui est associée, ce qui impliquerait que les erreurs de l'alignement avec BabelNet se cumuleraient avec les erreurs d'alignement entre les différentes éditions de langue de manière multiplicative.

Il est ainsi probablement préférable d'utiliser un ensemble de plongements de mots bilingues (comme ceux cités ci-dessus), puis de les contextualiser avec les signatures sémantiques rattachées aux définitions, à l'instar de ce qui a déjà été proposé par TEERAPARBSEREE (2005) pour la création d'acceptions avec l'utilisation de vecteurs sémantiques, sauf qu'ici nous travaillerons sur sur des vecteurs appartenant tous au même espace.

L'approche que nous pouvons utiliser est celle de la contextualisation faible que nous avons déjà décrite dans la Section 2.4.1.2. Rappelons-le, le principe de la contextualisation faible est d'amplifier les composantes des communes des vecteurs tous en atténuant les divergences. Ainsi, si nous supposons que la signature sémantique d'un sens, et plus particulièrement sa définition, est représentative de l'information sémantique portée par le sens, nous pouvons calculer un vecteur de sens par contextualisation faible du mot auquel appartient le sens par les mots de la définition du sens. Ainsi, nous obtenons un vecteur dont les composantes communes avec celles des vecteurs de la définition et donc caractérisant le sens seront renforcées, alors que les composantes divergent correspondant potentiellement aux autres sens du mots sont atténuées.

En partant du principe que nous utilisons des plongements sémantiques de mots se trouvant dans un espace bilingue aux vecteurs comparables, si nous appliquons cette procédure à des sens dans différentes langues, nous obtiendrons des vecteurs contextualisés pour chaque sens que nous pourrons comparer pour obtenir une similarité translingue.

Nous pouvons obtenir le vecteur correspondant à chaque sens en appliquant pour chaque mot de la définition la contextualisation faible (voir **Axiome** 2.4 et **Axiome** 2.3) entre le vecteur ce mot et le vecteur du lemme associé au sens en question déjà contextualisé par les mots précédents de la définition, comme présenté dans la **Définition** 4.2.

Définition 4.2 **Vecteur de sens par contextualisation faible**

Nous noterons l'espace vectoriel du plongement de mots bilingue $\mathcal{PB}_{l_1,l_2}$ et l'application du même nom $\mathcal{PB}_{l_1,l_2} : \mathcal{Chs} \rightarrow V_v$, qui retourne le vecteur correspondant à une forme de mot (forme de surface du vocable).

Soit la fonction $\mathtt{def}(s) = \mathrm{signf}(s,'\,\mathtt{def}')$, qui retourne la signature sémantique d'un sens $\mathcal{S}$, qui correspond à la définition du sens (**Définition 3.5**).

Nous pouvons obtenir le vecteur de sens en appliquant la contextualisation faible récursivement, c'est-à-dire partant du vecteur du mot duquel provient le sens que l'on contextualise successivement par les vecteurs des mots de la définition :

$$V_s = \mathcal{PB}_{l_1,l_2}(\mathtt{lemme}(\mathcal{E}l(s))) \tag{4.1}$$

$$\forall \mathtt{mot}_i \in \mathtt{def}(s) : V_s = \Gamma(V_s, \mathcal{PB}_{l_1,l_2}(\mathtt{mot}_i)) \tag{4.2}$$

Ainsi, une fois obtenus, les vecteurs $V_{s_{l_1}}$ et $V_{s_{l_2}}$, nous pourrons calculer la similarité entre le deux sens en calculant la dissimilarité entre les deux vecteurs. Pour calculer cette dissimilarité nous pouvons utiliser la mesure cosinus habituelle (**Axiome 2.1**), ou alors d'autres distances, étant donné que M. T. Pilehvar et Roberto Navigli (2015b) ont montré que la distance vectorielle optimale dépend grandement du type de vecteurs (word2vec, tfidf, specificité, etc.) et des ressources auxquelles elles sont appliquées. Par exemple *Weighted Overlap*, *Jensen-Shannon*, *Rank-biased overlap*, etc.

Il faut noter, que dans le cas où nous ne disposerions pas de plongements de mots multilingues pour générer les plongements multilingues, nous pouvons utiliser un système de traduction automatique comme dans la Section 4.2.2.1 afin de produire des définitions dans la même langue que nous pourrons utiliser pour calculer des plongements sémantiques comparables qui seront dans la même langue, tout en contournant implicitement, comme nous le verrons dans la Section 4.2.3.1, le problème de l'enrichissement des définitions mis en avant dans la section Section 4.2.2.1.

4.2.3 Alignement des sens restants

Les mesures de similarité translingues sont l'élément central lorsque l'on souhaite réaliser des alignements bilingues entre deux RLS. Il convient cependant de décrire d'une part comment enrichir les signatures pour éviter la sensibilité des mesures par recouvrement à la formulation précise des définitions, puis de décrire les différentes contraintes entrant en jeu au niveau du critère de décision pour l'alignement.

4.2.3.1 Enrichissement des signatures

L'enrichissement de signatures consiste à rajouter des symboles ou des mots à la signature sémantique afin de palier les problèmes de formulation exacte. De nombreux travaux ont été menés sur l'enrichissement de signature autour des mesures de similarité sémantique par recouvrement. Il existe deux familles d'approches.

ENRICHISSEMENT PAR LA STRUCTURE. La première idée est d'enrichir la définition ou le calcul de la similarité par l'utilisation de la structure de la ressource lexicale. Par exemple, dans une ressource telle que WordNet, cela consisterait à étendre la signature en incluant les définitions des *synsets* reliés. C'est par exemple ce qui se fait dans le contexte de la mesure de *Lesk étendue* (BANERJEE et PEDERSEN, 2002).

ENRICHISSEMENT PAR THÉSAURUS. Une deuxième manière de procéder est d'enrichir les définitions par des termes proches de chacun des mots, en utilisant un thésaurus listant les mots proches ; le thésaurus pouvant soit être une ressource construite manuellement par des experts, soit être généré de manière automatique.

L'approche automatique a pris de l'ampleur ces dernières années avec les plongements de mots, mais c'est une approche utilisée depuis des années en WSD. Nous pouvons en particulier citer les thésaurus distributionnels de D. LIN (1998b), qui génère un espace vectoriel mot-mot par ASL à partir d'un corpus annoté en dépendances reprenant les idées de PEREIRA, TISHBY et LEE (1993).

Notons aussi l'application plus récente de ces techniques pour l'enrichissement de définitions en WSD. T. MILLER, BIEMANN et al. (2012) montre une amélioration nette de la qualité de la désambiguïsation lors de l'enrichissement. Le même type de technique peut être appliqué avec des espaces vectoriels de mots tels que les plongements de mots. J'ai moi-même participé à des travaux tournant autour de l'enrichissement de définitions dans un contexte de WSD à partir d'un thésaurus construit à partir des contextes d'usage des sens issus de corpus annotés en sens, ce qui mène à des résultats dépassant l'état de l'art des systèmes supervisés (VIAL, TCHECHMEDJIEV et SCHWAB, 2016).

SIMILARITÉ PAR PLONGEMENT ET ENRICHISSEMENT IMPLICITE. Lors de l'usage d'un espace vectoriel de plongement de mots pour l'extension, tout comme l'utilisation du thésaurus distributionnel aussi fondée sur des plongements, un moyen de procéder, celui par exemple retenu dans les travaux de T. MILLER, BIEMANN et al. (2012), est de simplement ajouter les N mots les plus similaires à chaque mot de la définition.

Une fois que les définitions sont étendues, on peut calculer une similarité par recouvrement afin d'obtenir une similarité que l'on espère meilleure. Cependant, sachant que les N premiers mots sont souvent calculés en choisissant les N vecteurs les plus similaires par la similarité cosinus, comme c'est le cas par exemple lors de l'utilisation de l'outil `word2vec`.

Il peut être beaucoup plus judicieux d'utiliser directement les vecteurs des mots proches pour calculer le recouvrement et ainsi la similarité. On en revient ainsi dans ce cas de figure particulier (qui par ailleurs correspond au cas le plus typique à l'heure actuelle dans l'état de l'art) décrit dans la Section 4.2.2.3. Cette manière de procéder est dans l'esprit très similaire de l'approche générique d'alignement de ressources proposée par M. T. PILEHVAR et Roberto NAVIGLI (2014).

Du fait du manque dans DBNary d'une structure cohérente et consistante à travers l'ensemble des éditions de langues qui doivent être alignées, il est préférable d'utiliser cette dernière approche pour calculer les similarités multilingues entre les sens sur la base de plongements de sens multilingues, car cela permet de calculer la similarité tout en intégrant implicitement un enrichissement des définitions.

4.2.3.2 *Critère de décision de l'alignement*

Nous voulons ici appliquer des techniques d'AAS, afin d'obtenir un graphe de traduction de sens nécessaire à la création initiale des acceptions interlingues. C'est la situation présentée par la Figure 4.9, où toutes les relations de traduction vont de sens à sens.

Nous avons présenté un certain nombre de techniques qui permettent le calcul d'une similarité interlingue sur la base des définitions, et sans recourir à la structure de la ressource, qui dans le cas de DBNary est très incomplète, mises à part les trois éditions les mieux développées.

Étant donné qu'il n'y a pas de critère particulier pour préférer une technique par rapport à une autre et pas de manière simple d'évaluer la performance relative de chaque mesure sur les différentes paires de langues, il est tout à fait possible d'adopter la même approche que Matuschek (2015), qui est décrite dans la Section 2.1.2.1 :

$$(sim_1(ss, sc) > \theta_1) \wedge (sim_2(ss, sc) > \theta_2) \wedge \ldots \wedge (sim_n(ss, sc) > \theta_n) \quad (4.3)$$

Le désavantage évident est que chaque mesure est associée à une valeur seuil pour la décision d'alignement, et qu'il faut apprendre automatiquement ou le déterminer de manière experte. Ce que Matuschek (2015) propose est de faire une conjonction des décisions prises par chaque mesure, ce qui signifie qu'ils favorisent la précision en admettant une augmentation des faux négatifs. C'est en réalité une fusion *a posteriori*, c'est-à-dire un vote de type *tout ou rien* qui est réalisé.

Il est tout à fait envisageable de relaxer le critère de conjonction en admettant plus de faux positifs en passant à une stratégie de vote majoritaire, dans une situation où il y aurait au moins trois mesures différentes utilisées pour arriver à la décision globale. Il s'agit d'accepter la décision majoritaire de la combinaison des mesures de similarité utilisées par un algorithme de fusion assez simple (Algorithme 4.1).

Algorithme 4.1 – DecisionVoteMajoritaire

```
fonction DecisionVoteMajoritaire(sims = [sim_1,...,sim_n], seuils = [
    θ_1,...,θ_n]):
  countAccept = 0
  countReject = 0

  pour chaque (sim dans sims, θ dans seuils):
    si sim > θ:
      countAccept +=1
    sinon:
      countReject +=1

  retour countAccept > countReject
```

Une autre approche serait de combiner toutes les mesures de similarité en une seule en en faisant une moyenne pondérée. Ce qui ne nécessiterait d'utiliser qu'un unique seuil au lieu d'un seuil par mesure. Les mesures de similarité pouvant individuellement être évaluées sur des jeux de données bilingues, il est bien plus aisé de choisir des poids pour chaque mesure, exprimant leurs fiabilités respectives.

Même *a priori*, nous pouvons faire l'hypothèse que l'approche de similarité par traduction mot à mot au travers de la ressource à aligner elle-même sera d'une fiabilité moindre, car seul un petit sous-ensemble des relations de traduction est reliées aux sens sources, ce qui veut dire que dans le reste des cas toutes les traductions possibles seront utilisées, ce qui créera du bruit.

Par contre, il est bien plus difficile de départager l'approche par traduction automatique et l'approche par plongement sémantique, car cela dépend énormément de la qualité du système de traduction et du plongement multilingue qui en eux-mêmes dépendent directement de la qualité et de la représentativité du corpus vis-à-vis des usages des sens présents dans la ressource lexico-sémantique. Une hypothèse initiale cohérente pour rechercher des valeurs exactes serait par exemple affecter une confiance de 20% à la première mesure que l'on sait de moins bonne qualité et partager les 80% restants entre les deux autres mesures que nous ne savons pas départager *a priori*, soit 40% chacune.

Le problème final qui se pose est la modalité de combinaison en elle-même. Il est probable que les intervalles des différentes mesures ne seront pas les mêmes, en particulier les mesures par recouvrement. Ainsi, une moyenne arithmétique pondérée des différentes similarités ne permettrait pas de capturer fidèlement la contribution de chaque mesure. Une solution non supervisée et non paramétrique pour les combiner serait de calculer la moyenne géométrique des similarités, en supposant que les similarités ont des distributions à peu près uniformes dans les valeurs. C'est une technique de combinaison peu rencontrée dans la littérature, mais qui a cependant été utilisée avec succès, notamment BUSCALDI et al. (2012) dans le contexte de Semeval 2012.

$$\sqrt[n]{\omega_1 sim_1 \cdot \omega_2 sim_2 \cdot \ldots \cdot \omega_n sim_n} > \theta \tag{4.4}$$

L'estimation de θ nécessite tout comme dans MATUSCHEK (2015) une estimation supervisée avec un alignement existant. Cependant, pour l'estimation des poids associés à chaque mesure, il est possible d'utiliser des corpus parallèles comme données d'entraînement pour évaluer la qualité des mesures et pour apprendre une similarité combinée de la moyenne géométrique, ou alors utiliser des jeux de données de similarité bilingue. Une manière d'apprendre une telle mesure serait de suivre la procédure décrite dans l'article récent de (ZADEH, HOSSEINI et SRA, 2016).

Même si ici nous nous sommes focalisé sur des mesures translingues spécifiques pour les RLS, la combinaison de mesures est un élément indépendant. Ainsi, dans une étude comparative sur les mesures de similarité de texte translingue de l'état de l'art, FERRERO et al. (2016) étudient également les combinaisons de mesures de similarité. Ils trouvent que la meilleure manière de combiner est d'apprendre un arbre de classification qui permet de choisir la mesure la plus adaptée à un texte particulier, ce qui constitue une mesure de similarité adaptative.

Dans notre cas, bien entendu, le but de l'utilisation de plusieurs similarités est plutôt de permettre d'augmenter la précision de l'alignement en garantissant moins de bruit, et il n'est pas certain qu'apprendre une mesure dynamique soit le meilleur choix, mais c'est une solution qui mérite d'être explorée.

En théorie, il faut estimer un θ pour chaque paire de langues, ce qui est très contraignant, car cela suppose de produire des petits jeux de données étalons de référence pour chacune des paires de langues.

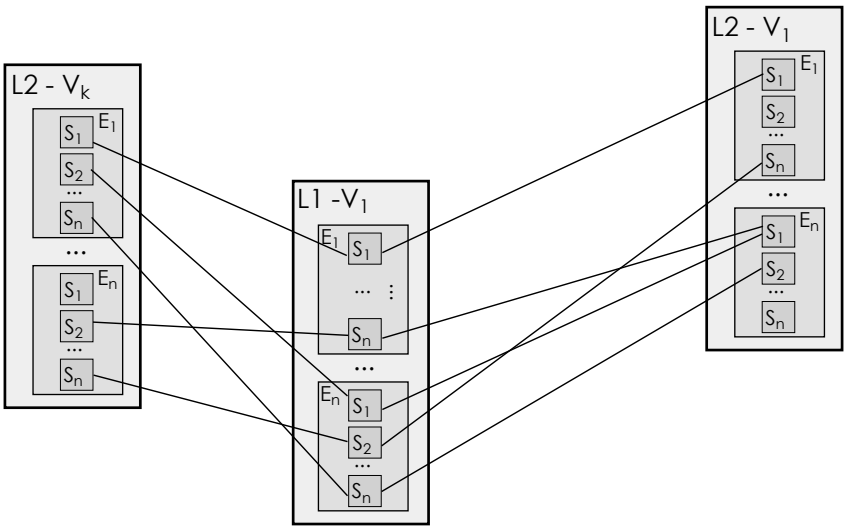


Figure 4.9 – Les données de *DBNary* comme graphe de traduction de sens après un alignement de sens de mots deux à deux.

Une solution alternative pour ne pas avoir à générer un tel jeu de données serait d'extraire tous les cas où il y a un unique candidat d'alignement, c'est-à-dire d'aligner deux entrées qui ont chacune un sens et qu'on sait déjà traduction l'une de l'autre. Dans ce cas, nous savons que ces sens devront être alignés et que c'est donc la décision correcte. On peut ainsi calculer la similarité entre ces deux sens et utiliser cette similarité comme caractéristique pour apprendre θ. L'incertitude associée à cette idée est bien entendu qu'il n'est pas certain que cette situation soit suffisamment courante pour produire assez de données d'entraînement. Ainsi, une étape préalable à l'alignement serait une étude quantitative répertoriant le nombre d'occurrences, dans les paires d'éditions de DBNary, dans le cas où l'alignement est évident combinatoirement, sans l'utilisation d'une mesure de similarité.

4.3 Évaluation du graphe de traduction

4.3.1 Étalon de référence intrinsèque (in vitro)

Comme nous l'avons décrit précédemment, la construction manuelle d'acceptions interlingues pose problème, car trouver des contributeurs volontaires est difficile. Cependant, à l'issue du projet Papillon (Boitet, Mangeot et Sérasset, 2002), un certain nombre d'acceptions avaient été produites, et ces dernières pourraient être exploitées pour produire un étalon de référence d'alignement afin d'évaluer, même d'une manière sommaire, les algorithmes de production d'acceptions.

L'obstacle majeur à ce mode opératoire est que d'une part les langues présentes dans Papillon-NADIA ne correspondent pas forcément à des éditons de *DBNary* avec des quantités de données exploitables, et d'autre part, Papillon utilise ces propres volumes monolingues, dont les distinctions en sens sont différentes de celles présentes dans DBNary. Ainsi, pour utiliser les données de Papillon comme étalon de référence, il faudrait d'abord aligner les sens des volumes monolingues de Papillon avec les éditions de langue correspondantes dans

DBNary , ce qui nécessiterait l'utilisation d'un algorithme d'AAS ou alors un alignement manuel par des experts pour les N volumes interlingues.

L'emploi de AAS risque de produire un étalon de référence erroné, alors que la construction manuelle de l'alignement prendrait beaucoup de ressources et de temps, qui pourraient au total être consacrés à produire directement des acceptions manuellement sur les données de *DBNary*.

Par contre, étant donné les propriétés axiomatiques et les algorithmes de construction proposés dans le Chapitre 3, il serait en fait possible de partir d'alignements bilingues parfaits (produits manuellement), c'est-à-dire d'étalons de référence d'AAS bilingues, à partir desquels nous pourrions générer des acceptions de référence valides. Ainsi, produire des étalons de référence bilingues sur un petit sous-ensemble de sens (disons autour de 500-1000) serait très probablement moins coûteux en temps et en argent que d'essayer de produire des acceptions interlingues de manière experte.

Pour la construction de l'étalon de référence, s'il y a une base d'acceptions existante correcte, il est possible de partir des acceptions contenues dans celle-ci et de suivre ensuite le cheminement inverse de l'algorithme de construction.

Nous pouvons générer la liste des alignements bilingues qui génèrent l'acception et qui peuvent donc servir d'étalon de référence pour les algorithmes d'AAS bilingue employés lors de la construction. Sinon, il faudra sélectionner un certain nombre d'entrées, qui sont traductions les unes des autres, afin de générer le graphe des alignements candidats afin de proposer des alignements à réaliser aux experts.

Le fait de partir d'une fermeture transitive de traduction au niveau des sens, permet de s'assurer que les alignements de sens correspondront à ceux qui sont nécessaires à la génération d'acceptions, car leur construction se base justement sur le graphe de la fermeture transitive de traduction (même si c'est au niveau du sens et non de l'entrée).

Nous pouvons ainsi ici utiliser les mesures de qualité par rapport à une référence de Teeraparbseree (2005), qui montre comment définir les vrais positifs et négatifs et les faux positifs et négatifs de la base d'acceptions à partir d'alignements individuels, ce qui est exactement ce qui est requis ici.

4.3.2 Évaluation extrinsèque (in vivo)

L'évaluation extrinsèque d'une RLS consiste à utiliser cette RLS comme source de connaissances dans une tâche particulière et ensuite à évaluer la performance du système sur cette tâche par rapport à un système de base. Dans le cas présent, on voudrait comparer la ressource alignée avec des acceptions interlingues et comparer la performance sur la tâche avec la même ressource où on utilise un pivot naturel tel que l'anglais. Ainsi, nous serions en mesure de quantifier de manière concrète l'apport de la construction des acceptions interlingues.

Une condition *sine qua non* pour que ce type d'évaluation fonctionne est d'appliquer la ressource à une tâche où les divergences de lexicalisation sont un élément essentiel de la réussite. Il y a en réalité une seule méthode permettant de réellement capturer cet apport pleinement, la traduction automatique. Il n'est bien sûr pas évident de construire un système de traduction automatique complet puis d'évaluer la performance de ce système dans l'absolu (l'évaluation correcte en traduction automatique étant un obstacle majeur dans le domaine).

Cependant nous pouvons utiliser une sous-tâche de la traduction automatique, à savoir la traduction d'un mot en contexte de la langue source vers une langue cible. Cette tâche est habituellement appelée *substitution lexicale translingue* et a fait l'objet de tâches d'évaluation dans plusieurs campagnes SemEval.

4.3.2.1 Tâche d'évaluation

La tâche de substitution lexicale translingue organisée dans le contexte de SemEval 2010 par Mihalcea, Sinha et McCarthy (2010) se base sur le principe de la tâche de substitution lexicale `LEXSUB` de l'anglais organisée lors de SemEval 2007 (McCarthy et Roberto Navigli, 2007), mais dans un contexte multilingue. Dans la tâche originale, chaque instance était une phrase avec un mot-cible pris dans cette phrase et le but était de produire des mots équivalents en prenant en compte le contexte en anglais.

Dans la version translingue, l'objectif est de faire la même chose, mais dans une autre langue, c'est-à-dire de produire des traductions contextualisées différentes.

La limite de cette tâche est que les données d'évaluation ne sont fournies que pour l'espagnol et que c'est donc la seule langue qu'il est possible d'utiliser pour l'évaluation de la ressource, ce qui est une limite, mais pas nécessairement un problème, du moment que les termes cibles anglais en question peuvent être traduits de différentes manières en fonction du contexte. Étant donné que c'est justement le but de la tâche, l'ensemble des instances devrait correspondre à cette situation.

Le principe de la tâche est de fournir une phrase de contexte, et de cibler un mot dans ce contexte pour lequel il faut fournir une ou plusieurs traductions en espagnol qui correspondent au contexte d'usage fourni.

La tâche fournit une longue liste d'entrées et de réponses attendues (étalon de référence) ainsi qu'un script d'évaluation permettant de calculer la meilleure précision, le meilleur rappel, le rappel «parmi dix» (*out-of-ten recall*), et la précision «parmi dix» (*out-of-ten precision*).

Entrée : Perhaps the effect of West Nile Virus is sufficient to extinguish endemic birds already **-> severely <-** stressed by habitat losses.

Réponse attendue : `{fuertemente, severamente, duramente, exageradamente}`

Figure 4.10 – Exemple de données de test de la tâche de substitution lexicale translingue et des réponses attendues.

L'ensemble des instances est noté I, une instance est notée $i \in I$, l'ensemble des traductions étalons de référence attendues est noté T_i, et l'ensemble des réponses du système sont notées S_i.

Le meilleur score associé à chaque instance est calculé comme indiqué dans l'Équation 4.5 ci-dessous :

$$\text{meilleurscore}(i) = \frac{\sum_{s \in S_i} \text{frequence}(s \in T_i)}{|S_i| \cdot |T_i|} \tag{4.5}$$

À partir de ce meilleur score individuel, on calcule la meilleure précision (Équation 4.6) et le meilleur rappel (Équation 4.7).

$$\text{meilleureprecision} = \frac{\sum_i \text{meilleurscore}(i)}{|\, i \in I : S_i \neq \emptyset \,|} \tag{4.6}$$

$$\text{meilleurrappel} = \frac{\sum_i \text{meilleurscore}(i)}{|\, I \,|} \tag{4.7}$$

De même, à partir du score individuel parmi dix (Équation 4.8), sont calculés le rappel «parmi dix» (Équation 4.9) et la précision «parmi dix» (Équation 4.10).

$$\text{scoreparmidix}(i) = \frac{\sum_{s \in S_i} \text{frequence}(s \in T_i)}{|\, T_i \,|} \tag{4.8}$$

$$\text{precisionparmidix} = \frac{\sum_i \text{scoreparmidix}(i)}{|\, i \in I : S_i \neq \emptyset \,|} \tag{4.9}$$

$$\text{meilleurscore}(i) = \frac{\sum_i \text{scoreparmidix}(i)}{|\, I \,|} \tag{4.10}$$

Il faut noter que cette tâche est très similaire à la tâche de désambiguïsation lexicale translinque (*Cross-lingual Word Sense Disambiguation*), également présente dans SemEval 2010 (Lefever et Hoste, 2010), à la différence que, dans cette dernière, il y a une restriction sur l'utilisation de sources de données. Il n'est en effet imposé d'utiliser uniquement le corpus EuroParl pour l'entraînement et la création du système, les réponses possibles pour chaque entrée étant définies par les contextes d'usage définis dans le corpus.

Cependant, les données sont fondamentalement similaires. Chaque instance est une phrase dont un mot particulier est la cible. Le but de la tâche est de fournir une unique traduction qui correspond exactement à la lexicalisation telle que définie dans la phrase alignée dans EuroParl. Concrètement, rien ne nous empêche d'ignorer les distinctions de sens imposées par EuroParl et d'utiliser la même évaluation que pour la tâche de substitution translingue. Il est en effet très facile de convertir les données d'évaluation d'un format à l'autre.

Cela signifie que nous pourrons évaluer le système avec le jeu de données anglais->espagnol de la première tâche et avec les jeux de données anglais->français, anglais->allemand, anglais->néerlandais et italien de la deuxième tâche. Les résultats sur la deuxième tâche seront par contre bien moindres du fait qu'il n'y a qu'une unique solution possible.

4.3.2.2 Construction d'une ressource de référence par pivot naturel

Il y a une chose qui est essentielle si l'on veut vraiment calculer l'impact des acceptions interlingues. Lors de la construction de la ressource de référence par pivot naturel, il faut que le pivot soit différent de la langue source de la tâche d'évaluation. En effet, dans le cas où le pivot serait l'anglais et ou la source pour l'évaluation serait aussi en anglais, il ne pourrait jamais y avoir de problème de perte de contraste. Pour cette évaluation, il faudra donc choisir un pivot autre que la langue source. Un bonne idée est de prendre le français comme pivot, car

il s'agit de la deuxième édition de Wiktionary en terme de taille. Il faudra par contre exclure la paire anglais->français lors de l'évaluation.

Pour construire la ressource de référence en partant du graphe de traduction de sens contenant l'ensemble des alignements bilingues utilisé pour la construction des acceptions interlingues, l'algorithme est relativement simple.

Nous partons sur la base de l'Algorithme 3.1 en itérant sur l'ensemble des composants connectés du graphe (Algorithme 4.2) puis en regroupant dans un pivot tous les sens alignés avec un sens Français (Algorithme 4.3) :

Algorithme 4.2 – PivotNaturelDistribue

```
fonction PivotNaturelDistribue(G):
  composants = ComposantsConnectes(G)
  acceptions = ∅
  pour chaque composant dans composants:
    pivots = pivots ∪ ConstruirePivotNaturel(composant)
  retour acceptions
```

Algorithme 4.3 – PivotNaturelDistribue

```
fonction ConstruirePivotNaturel(G):
G = (V, E)
// V= sens
// E= alignements
    pivots = ∅
  pour chaque v_i ∈ {v_i ∈ V ⊂ S | langue(v_i) ="fr"}:
    pivots = pivots∪{v_t ∈ V | (v_i, v_t) ∈ E}
  retour pivots
```

4.3.2.3 Système de désambiguïsation translingue pour l'évaluation

L'algorithme est très proche de l'algorithme de calcul de similarité translingue par traduction au travers d'une ressource lexicale (Figure 4.8) dans le déroulement général (Figure 4.11 ci-dessous).

1. Lemmatisation et annotation en catégories grammaticales.
2. Désambiguïsation lexicale de l'ensemble du contexte fourni pour l'instance courante dans la langue source.
3. Consultation des traductions possibles dans la ressource lexicale (Référence avec pivot naturel ou pivot par acceptions interlingues).

L'élément central qui différencie cet algorithme de celui de la projection lors du calcul de similarité, est que l'on ne fait la consultation que pour le mot cible. De plus, on ne va pas consulter un graphe d'alignements deux à deux, mais la ressource à pivot.

Afin de consulter la ressource par pivot, il faudra pour le sens sélectionné récupérer le pivot auquel il est lié, et, au travers du pivot retrouver l'ensemble des sens de la langue cible l_c et de récupérer les lemmes associés (Algorithme 4.4). On considère l'existence d'une fonction $Pivot(s \in S)$ qui pour un sens retourne son pivot :

— pour la ressource par acceptions elle retourne la classe d'équivalence à laquelle appartient directement le sens dans la hiérarchie des classes d'équivalences correspondante,

— et pour le pivot naturel la fonction retourne l'unique pivot auquel appartient le sens.

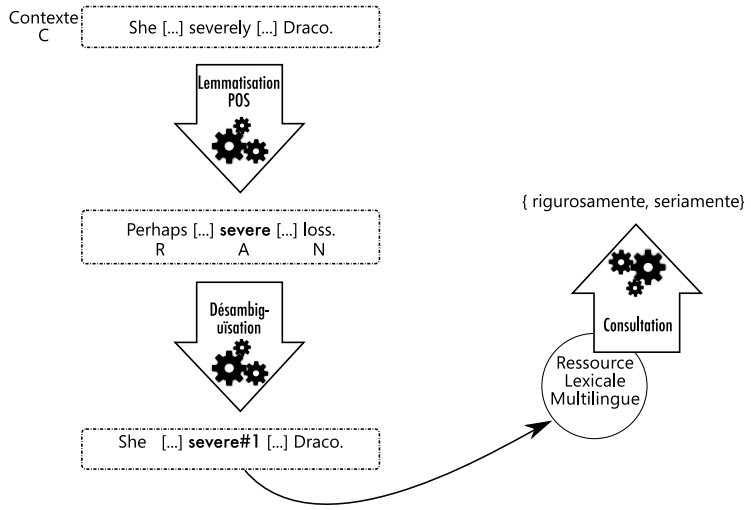


FIGURE 4.11 – Système de substitution translingue par désambiguïsation lexicale.

Algorithme 4.4 – ConsulterTraductions

```
1 fonction ConsulterTraductions(s_s, l_c) :
2   retour lemme{sense_c | s_c ∈ Pivot(s_s) ∧ langue(s_c) = l_c}
```

4.3.2.4 Proposition d'un protocole expérimental d'évaluation

Maintenant que nous avons présenté les éléments du système, nous définissons un protocole expérimental pour l'évaluation d'une ressource à base d'acceptions interlingues :

1. Génération des ressources lexicales à partir des alignements bilingues.
   REFPN. Référence avec un pivot naturel dont la langue est différente de la source ou la cible de la tâche d'évaluation.
   RACIN. Ressource alignée par acceptions interlingues.
2. Exécution de l'algorithme de désambiguïsation translingue sur les données d'évaluation de la tâche en utilisant respectivement les ressources REFPN et RACIN.
3. Génération des résultats et analyse.

Nous faisons l'hypothèse (**H1**) que le système basé sur la ressource REFPN obtiendra des résultats inférieurs à ceux du système se basant sur la ressource RACIN, particulièrement sur les entrées où il y a une divergence de lexicalisation.

Une autre hypothèse plausible (**H2**) est que la différence entre les résultats obtenus entre le système REFPN et RACIN pour des paires de langues plus proches sera plus petite que pour des langues plus éloignées (familles de langues, mais aussi différence grammaticales). En effet les langues qui ont évolué dans un environnent culturel et social commun ont plus de chances de partager les mêmes concepts et la même granularité de lexicalisation.

4.4 Implémentation des outils et algorithmes – LexSemA

Une partie importante de la thèse s'est concentrée sur l'implémentation et la mise en place d'un écosystème logiciel qui contient :

— l'accès à des ressources en données lexicales liées et en particulier DBNary ;
— l'ensemble des outils nécessaires à la mise en place des algorithmes de désambiguïsation lexicale (monolingue et translingue),
— des mesures de similarité translingue et tous les composants logiciels pour permettre de générer les modèles ;
— des modules d'enrichissement de signatures sémantiques ;
— des composants d'interface avec des système de traduction en ligne ou Moses ;
— des outils pour l'apprentissage automatique et l'optimisation ;
— des modules pour l'alignement de sens (clustering, structure, similarité, etc.) ;
— des algorithmes de construction et de mise à jour pour les acceptions interlingues.

C'est, en effet, une précondition importante à toute expérience d'alignement et de construction d'acceptions interlingues, car de nombreux outils et d'algorithmes hétérogènes sont nécessaires. Il est particulièrement important de ne pas se contenter d'assembler les outils en pêle-mêle avec des scripts, mais de construire des outils et une architecture pérennes, mis à disposition du public, afin que chacun puisse construire ces propres ressources et reproduire les expériences.

Il existe bien entendu des plateformes logicielles couvrant partiellement nos besoins, on pense tout particulièrement à la suite d'outils de l'université de Darmsdtadt :

— dkpro-core (Gurevych, Mühlhäuser et al., 2007) ;
— dkpro-similarity (Bär, Zesch et Gurevych, 2013) ;
— dkpro-lsr/dkpro-uby (Garoufi, Zesch et Gurevych, 2008 ; Gurevych, Eckle-Kohler, Hartmann, Matuschek, Christian M. Meyer et al., 2012) ;
— dkpro-wsd (T. Miller, Erbs et al., 2013) ; dkpro-labs (R. E. d. Castilho et Gurevych, 2011).

Les outils que nous développons vont au delà de ces composants, notamment au niveau de l'alignement et de la création d'acceptions, et n'ont pas vocation à refaire ce qui existe déjà dans dkpro par exemple. Une architecture modulaire à base de `maven`[4] a été spécifiquement mise en place pour permettre une compatibilité et une intégration avec les systèmes existants. Par exemple, pour la segmentation, la lemmatisation, l'annotation en parties du discours, ce sont les chaînes de traitement UIMA-fit de dkpro qui sont utilisées.

Un autre exemple est le module d'alignement de sens, qui implémente des algorithmes de clustering spécifiques à l'alignement des sens, mais qui permet également d'intégrer des outils d'alignement existants, comme par exemple ADW (T. M. Pilehvar, Jurgens et Roberto Navigli, 2013).

Tout comme dkpro, le projet lexsema se base sur une architecture maven multimodule qui permet de gérer facilement les dépendances et interdépendances de manière très fine. Le projet racine se nomme `org.getalp.lexsema` et se compose des sous modules suivants :

4. maven est un système de gestion des dépendances pour les projets Java ; il gère l'ensemble du cycle de vie d'un projet logiciel et permet de récupérer toutes les dépendances automatiquement. http://apache.maven.org

— `lexsema-io` – prend en charge les entrées sorties (chargement de corpus, analyse en chaîne linguistique, écriture des résultats d'évaluation, chargement de sens en vue de l'annotation).
— `lexsema-ontolex` – propose une API générique pour l'implantation d'APIs spécifiques pour les ressources Lemon/Ontolex et comprend une API pour un accès aisé à DBNary.
— `lexsema-ml` – regroupe les outils d'apprentissage automatique de manière cohérente (optimisation, regroupement automatique, filtrage et réduction de dimensionnalité).
— `lexsema-translation` – Gère l'interfaçage avec des services de traduction (Bing, Baidu) et propose une implémentation de la traduction ciblée de mots mettant en jeu DBNary et un algorithme de désambiguïsation lexicale
— `lexsema-util` – Propose un système de cache et contient des classes utilitaires utilisées partout dans le projet.
— `lexsema-similarity` – Permet le calcul de similarités sémantiques à la fois dans un contexte monolingue et dans un contexte multilingue. Il inclut aussi une interface pour traduire (projeter) les symboles sémantiques pour arriver à des mesures de similarité translingue ainsi qu'une API pour l'extension de contextes sémantiques. Supporte le calcul à différents niveaux : sens, mots, phrases.
— `lexsema-wsd` – implante des algorithmes de désambiguïsation lexicale, qui sont nécessaires à l'évaluation des acceptions, notamment pour la tâche de désambiguïsation translingue.
— `lexsema-axalign` – implémente les algorithme de création d'acceptions ainsi que les algorithmes pour l'évaluation.

Chacun des modules se décompose *a minima* en deux sous-modules, un module `-api` contenant l'ensemble des interfaces de programmation du module et une module `-core` contenant les implémentations de base des interfaces de programmation. Il peut ensuite y avoir des sous-modules supplémentaires qui font le lien avec d'autre bibliothèques externes, afin de compartementaliser les nombreuses dépendances.

Par exemple, le module `org.getalp.lexsema-io` contient les sous modules suivants :

— `org.getalp.lexsema-io-api` (comme décrit ci-dessus) ;
— `org.getalp.lexsema-io-core` (comme décrit ci-dessus) ;
— `org.getalp.lexsema-io-babelnet` qui compartementalise les dépendances pour l'accès à BabelNet ;
— `org.getalp.lexsema-io-uimadkpro` qui propose des implémentations de chaînes de traitement basées sur dkpro.

Dans cet exemple, la séparation de `org.getalp.lexsema-io-uimadkpro` du reste est très utile, car dkpro télécharge les modèles des différents outils par maven et les inclut comme des ressource dans l'archive `jar`, ce qui peut mener à des archives jar très volumineuses (de l'ordre du gigaoctet), ainsi les dépendances ne seront incluses que quand il y en aura besoin.

L'objectif à plus long terme et d'assurer la compatibilité descendante et ascendante avec les modules existants dkpro afin de minimiser les duplications de fonctionnalités. Il est évident que dkpro est maintenu de manière professionnelle avec bien plus de ressources financières et une assurance de qualité supérieure

et qu'il est donc inutile de dupliquer des choses en sachant que les révisions de code et le débogage sont particulièrement chronophages.

Conclusions

Ce chapitre a abordé les questions pratiques et les contraintes techniques précédant la construction des acceptions interlingues. Nous avons d'abord présenté le cas particulier de DBNary et avons mis en avant les problèmes structuraux à résoudre avant de pouvoir aligner les éditions de langues au niveau des sens.

Ainsi, dans une première expérience, nous avons utilisé les informations présentes dans Wiktionary afin de rattacher certaines relations de traduction aux sens de mot au niveau de la source et avons atteint une précision de l'alignement similaire à l'état de l'art.

Ensuite, nous avons présenté d'autres informations actuellement inexploitées qui peuvent servir à l'alignement au niveau du sens de relations de traduction supplémentaires.

Enfin, nous avons présenté le problème de l'alignement des éditions de Wiktionary au niveau du sens de manière plus précise, et avons proposé, sur la base de l'état de l'art. des pistes sur les techniques les plus prometteuses pour exploiter au mieux les spécificités des données de DBNary.

L'alignement des différentes éditions de Wiktionary en lui-même, si nous ne pouvont pas l'évaluer, n'a pas beaucoup de sens. Ainsi, nous avons fait le point sur les options disponibles pour la validation et l'évaluation des ressources à base d'acceptions interlingues et avons proposé un protocole pour l'évaluation *in vivo* de telles ressources dans le contexte d'une tâche de substitution lexicale translingue.

Pour finir, nous avons présenté l'environnement logiciel développé dans le courant de cette thèse pour permettre de servir de support pour l'ensemble des expériences et des algorithmes proposés.

Même si la contribution expérimentale de ce chapitre est assez limitée, il convient de noter que l'ensemble des ressources, des algorithmes et des outils nécessaires à la construction et à l'évaluation de ressources par pivot interlingue est implémenté. Le passage à la pratique sera donc relativement aisé, du moins dans un premier temps à petite échelle.

CONTRIBUTIONS. Les contributions à l'état de l'art de ce chapitre sont les suivantes :

1. Traitement préliminaire sur les données de DBNary pour ramener des relations de traductions au niveau des sens à la source.
2. Étude des mesures de similarité translingue pertinentes pour DBNary et des méthodes plausibles pour les combiner.
3. Proposition d'un protocole expérimental pour l'évaluation *in vivo* des RLS à base d'acceptions interlingues.
4. L'architecture logicielle LexSemA qui regroupe l'ensemble des outils et algorithmes pour la construction et l'évaluation de RLS.

Conclusions et travail à venir

Après un examen attentif de l'état de l'art sur les ressources lexicales, les niveaux et standards d'interopérabilité, ainsi que les techniques pour l'interopérabilité sémantique des ressources lexicales nous constatons les points suivants.

- Les formats pour les données lexicales liées et les technologies du Web sémantique offrent une réponse globalement satisfaisante pour l'interopérabilité représentationnelle des RLS.
- Les architectures d'alignement pour l'interopérabilité sémantique, ou du moins leurs implémentations pratiques ne sont pas suffisantes et présentent des défauts importants.
  - Les architectures par transfert nécessitent de calculer et de stocker toutes les alignements par paires entre touts les ressources alignées. Se pose ainsi un problème de passage à l'échelle, et aussi une difficulté d'accès aux données dans un contexte trilingue, quadrilingue, etc.
  - Les architectures à base de pivot naturel utilisent les sens d'une langue comme pivot, ce qui biaise la ressource alignée vis-à-vis de la conceptualisation de cette langue en particulier. On introduit ainsi des phénomènes contrastifs artificiels, ou en d'autre termes, une perte de contraste entre les langues ayant des lexicalisations divergentes similaires.
  - Les architectures à base de pivot artificiel, par exemple avec des acceptions interlingues, permettent de s'affranchir de certains des problèmes des deux autre types d'architecture. Cependant, un problème important qui se pose avec un véritable pivot interlingue est qu'il est très difficile à construire manuellement du fait de la difficulté à trouver des experts compétents. Il est de plus difficile de maintenir une communauté active pour la collaboration de contenu et de corrections. Il manque par ailleurs une vraie formalisation des acceptions interlingues qui permettrait une construction et une évaluation automatiques.
- Il n'existe pratiquement aucune prise en compte de l'interopérabilité dynamique, surtout pour les RLS provenant de ressource collaboratives évoluant sans cesse.

Partant de ces constats, notre premier travail a été d'attaquer le problème de la formalisation des acceptions interlingues. Sur cette idée, nous avons d'abord proposé un algorithme de construction initiale d'un alignement de ressources par acceptions interlingues. Puis, afin de permettre la compatibilité avec une prise en compte de l'interopérabilité dynamique, nous avons proposé des algorithmes pour les opérations de mise à jour pour les acceptions interlingues.

Dans un deuxième temps, nous avons examiné de manière plus pratique la mise en œuvre des algorithmes de construction d'acceptions, au travers d'un cas d'étude pratique, celui de DBNary. DBNary permet d'extraire les différentes éditions de langue de Wiktionary dans un format de données lexicales liées (`lemon`), à ce titre c'est un candidat idéal.

Cependant, l'algorithme de construction nécessite d'avoir préalablement établi tous les alignements de sens bilingues. Or les relation de traduction déjà présentes dans Wiktionary sont établies entre des entrées lexicales et des vocables, ce qui signifie qu'il faut d'abord les ramener au niveau des sens à la source comme à la cible.

Nous avions tout d'abord proposé une approche pour rattacher les sources des relation de traduction au niveau du sens. Ensuite nous avons proposé de manière théorique des mesures de similarité translingue applicables à DBNary, ainsi qu'un algorithme pour aligner l'ensemble des relations de traduction aux sens, à la fois à la source et à la cible.

Dans un troisième temps, nous avons étudié les solutions possibles pour l'évaluation d'acceptions interlingues produites à partir de DBNary. En effectuant d'abord des évaluations in vitro. Nous avons constaté qu'il n'y a que très peu de données issues de Papillon et qu'il serait difficile de les exploiter de manière précise. Ensuite, nous avons proposé une évaluation in vivo au travers de la tâche de substitution lexicale issue de SemEval, ainsi qu'une possible adaptation des données de la tâche de désambiguïsation lexicale translingue.

Pour finir, nous avons présenté LexSemA une boite à outils logicielle générique pour les RLS allant de pair avec ce travail de thèse, qui regroupe et implémente tous les outils nécessaires pour la construction d'acceptions interlingues et pour leur évaluation.

4.4.1 Limitations

Malgré tout cela, la limitation principale des techniques de création d'acceptions interlingues reste très clairement que, pour la construction initiale, il faut établir toutes les paires d'alignements bilingues. Même si l'algorithme d'ajout permet de s'affranchir de la nécessité d'avoir toutes les paires d'alignements, il reste tout de même nécessaire d'aligner toutes les paires marquant une divergence de lexicalisation. Par ailleurs, étant donné que cet alignement se fera par AAS, il y aura nécessairement des erreurs, qui viendront s'accumuler lors de la génération des acceptions.

4.4.1.1 Disponibilité de ressources pour la construction initiale

Du fait de la première limitation, il faudra donc pouvoir réaliser l'alignement dans un nombre conséquent de paires de langues. Dans le cas de DBNary, étant donné que mise à part les 10-15 éditions les plus grandes il y a très peu d'entrées, il faudra inévitablement faire appel à d'autres ressources.

Il faut donc avoir des dictionnaires bilingues au niveau des mots pour la plupart des langues couverte par la ressource finale. Même si il n'existe pas de dictionnaires bilingues pour certaines paires, il est possible d'en générer à partir de corpus parallèles à la manière de Ammar et al. (2016). Ainsi, le goulot d'étranglement en ce qui concerne les sources de connaissances sera le même que pour la traduction automatique, la disponibilité de ressources parallèles.

La conséquence principale est qu'il sera très difficile d'intégrer des langues faiblement dotées dans la ressource finale sans au préalable doter la langue, comme c'est le cas dans la plupart des applications de TAL. Même dans le cas de DBNary,

cela signifie que beaucoup parmi les plus petites éditions pourront difficilement être intégrées.

4.4.1.2 Mise en pratique et évaluation réelle

Les contributions de cette thèse sont pour la majeure partie théoriques et le Chapitre 4 a illustré un nombre non négligeable d'obstacles à la construction. Le principal obstacle est l'établissement des alignements bilingues. Du fait de l'accumulation rapide des erreurs pour l'établissement des alignements bilingues préalables, il sera essentiel de travailler à la maximisation de la précision des alignements bilingues.

Les règles axiomatiques régissant les acceptions peuvent bien sûr être utilisées pour détecter certaines incohérences, comme décrit dans la partie du Chapitre 2 décrivant les travaux de Teeraparbseree (2005). Cependant, cette détection est limitée à des cas très précis. Par ailleurs, ces erreurs pourront uniquement être détectées et non corrigées automatiquement. Il faudra donc nécessairement recourir à des correcteurs humains. Il est probable que l'initiative récente portant sur l'index interlingue collaboratif sera une solution viable pour apporter ce type de corrections, vu que les correction portent uniquement sur des alignements bilingues (Bond, Piek Vossen et al., 2016).

Une chose qui manque est bien entendu la mise en pratique, même à petite échelle, malgré le fait que tous les outils pour le faire sont implémentés. Cette mise en pratique sera l'une des premières choses à faire dans nos travaux futurs.

4.4.2 Travail Futur

Cette section porte sur le travail futur à réaliser suite à cette thèse. Les objectifs à court terme sont des projets de recherche courts de 6 à 12 mois et peuvent faire l'objet de sujets de stage de M2R. Les projets à moyen terme sont des projets nécessitant de 2 à 4 ans pour être menés à bien, et peuvent faire l'objet de sujets de thèse de doctorat. Les projets à long terme sont des projets nécessitant de 5 à 10 ans et correspondent à des objectifs de carrière dans l'optique de l'habilitation à diriger les recherches.

4.4.2.1 À court terme

Il y a deux objectifs fondamentaux pouvant être réalisés à court terme :

- Achever l'implémentation des outils logiciels :
  - implémenter les algorithmes de création et de mise à jour ;
  - ajouter une représentation de travail de la ressource sans passer par des requêtes SPARQL afin d'optimiser la performance d'accès ;
  - optimiser et modulariser les dépendances afin d'améliorer la distribution des outils ;
  - assurer l'intégration et la compatibilité au niveau des interfaces de programmation avec les bibliothèques de la distribution dkpro.
- faire des expériences de création et d'évaluation d'acceptions interlingues à petite échelle, avec deux ou trois langues pour lesquelles il existe des jeux de données étalons de référence, par exemple en réutilisant les étalons de référence produits par Matuschek (2015).

— développer les algorithmes de AAS pour DBNary afin de produire des graphes de traduction sens à sens.

4.4.2.2 À moyen terme

Il est relativement limitant de construire et d'évaluer d'abord des alignements bilingues, puis dans une deuxième étape de créer les acceptions interlingues. Il sera intéressant de concevoir un algorithme général de AAS permettant de créer ces alignements d'une manière qui intègre les contraintes axiomatiques pour la construction initiale d'acceptions interlingues. Cette approche peut se baser sur le développement de l'algorithme d'ajout par divergence, qui mènera à la production d'une mesure de divergence translingue intégrant directement ces contraintes.

L'encodage des contraintes structurelles et combinatoires dans la définition de la mesure de divergence permettrait par ailleurs également de réaliser des plongements de mots multilingues projetés sur les acceptions interlingues, sur la base des travaux de (Ammar et al., 2016).

4.4.2.3 À long terme

Les objectifs à long terme principaux sont les suivants.

— Étendre les algorithmes à différents paradigmes de RLS, notamment pour convertir les ressource existantes telles que *BabelNet* dans une architecture d'alignement par pivot interlingue.

— Démocratiser ce type d'architecture dans le domaine et de produire une grande ressource unique intégrant différents niveaux de granularité sémantique et différents domaines.

— Produire une plateforme collaborative généralisée pour la correction et la mise à jour de la ressource produite, avec un système de facettes d'annotations qui permettent de cibler les utilisateurs compétents pour telle ou telle tâche. Cette plateforme pourra par exemple se baser sur l'évolution des travaux sur l'index interlingue collaboratif de Bond, Piek Vossen et al. (2016).

Bibliographie

Acar, Evrim et al. (2010). *Future Directions in Tensor-Based Computation and Modeling*. Rapp. tech. 0908059. University of Cornell, p. 1–20. URL : http://www.issnla2010.ba.cnr.it/FinalReport.pdf.

Agirre, Eneko et Aitor Soroa (2007). « SemEval-2007 Task 02 : Evaluating Word Sense Induction and Discrimination Systems ». In : *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic : ACL, p. 7–12. URL : http://www.aclweb.org/anthology/S/S07/S07-1002 (cf. p. 39).

Agirre, Eneko et Aitor Soroa (2009). « Personalizing PageRank for Word Sense Disambiguation ». In : *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece : ACL, p. 33–41. URL : http://www.aclweb.org/anthology/E09-1005 (cf. p. 49).

Allan, Keith (2001). *Natural Language Semantics*. 1re éd. Wiley-Blackwell (cf. p. 16).

Ammar, Waleed et al. (2016). « Massively Multilingual Word Embeddings ». In : *arXiv preprint arXiv :1602.01925*. arXiv : 1602.01925 (cf. p. 55, 102, 128, 144, 146).

Antoniou, Grigoris et al., éds. (2011). *The Semantic Web : Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*. T. 6643. Lecture Notes in Computer Science. Springer. ISBN : 9783642210334.

Apidianaki, Marianna et Benoît Sagot (2012). « Applying cross-lingual WSD to WordNet development ». In : *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Sous la dir. de Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey : European Language Resources Association (ELRA), p. 23–25 (cf. p. 54).

Arwini, Khadiga et Christopher Dodson (2008). « Introduction to Riemannian Geometry - Lecture Notes in Mathematics ». In : *Information Geometry* 1953.March, p. 19–30–30. DOI : 10.1007/978-3-540-69393-2_2.

Banerjee, Satanjee et Ted Pedersen (2002). « An adapted Lesk algorithm for word sense disambiguation using WordNet ». In : *CICLing 2002*. Mexico City (cf. p. 131).

Bär, Daniel, Torsten Zesch et Iryna Gurevych (2013). « DKPro Similarity : An Open Source Framework for Text Similarity ». In : *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*. Stroudsburg, PA, USA : ACL, p. 121–126 (cf. p. 140).

Barzilay, Regina et Michael Elhadad (1997). « Using Lexical Chains for Text Summarization ». In : *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, p. 10–17.

Bedaride, Paul et Claire Gardent (2008). « Réécriture et Détection d'Implication Textuelle ». In : *Actes de la 15eme conférence sur le Traitement Automatique des Langues Naturelles* (cf. p. 46).

Bérard, Alexandre et al. (2016). « MultiVec : a Multilingual and MultiLevel Representation Learning Toolkit for NLP ». In : *The 10th edition of the Language Resources and Evaluation Conference (LREC 2016)* (cf. p. 128).

Berlin, Jacob et Amihai Motro (2002). « Database Schema Matching Using Machine Learning with Feature Selection ». In : *Advanced Information Systems Engineering*. Sous la dir. d'AnneBanks Pidduck et al. T. 2348. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 452–466. isbn : 9783540437383. doi : 10.1007/3-540-47961-9_32 (cf. p. 44).

Berment, Vincent (2004). « Methods to computerize "little equipped" languages and groups of languages ». Thése. Université Joseph-Fourier - Grenoble I. url : https://tel.archives-ouvertes.fr/tel-00006313 (cf. p. 24).

Bethard, Steven et al. (2016). « Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) ». In : *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California : ACL. url : http://aclweb.org/anthology/S16-1000 (cf. p. 13).

Bhattacharyya, Pushpak (2010). « IndoWordNet ». In : *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Sous la dir. de Nicoletta Calzolari (Conference et al. Valletta, Malta : European Language Resources Association (ELRA). isbn : 2951740867.

Bhingardive, Sudha, Dhirendra Singh et Rudra Murthy (2015). « Unsupervised Most Frequent Sense Detection using Word Embeddings ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1238–1243.

Bohnet, Bernd (2010). « Top Accuracy and Fast Dependency Parsing is not a Contradiction ». In : *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China : Coling 2010 Organizing Committee, p. 89–97. url : http://aclweb.org/anthology/C10-1011 (cf. p. 12).

Boitet, Christian, Mathieu Mangeot et Gilles Sérasset (2002). « The PAPILLON project : cooperatively building a multilingual lexical data-base to derive open source dictionaries & lexicons ». In : *Proceedings of the 2nd workshop on NLP and XML-Volume 17*. ACL, p. 1–3 (cf. p. 33, 134).

Bond, Francis et Ryan Foster (2013). « Linking and Extending an Open Multilingual Wordnet ». In : *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Sofia, Bulgaria : ACL, p. 1352–1362. url : http://www.aclweb.org/anthology/P13-1133 (cf. p. 37, 51, 54).

Bond, Francis, Piek Vossen et al. (2016). « CILI : the Collaborative Interlingual Index ». In : *Proceedings of the eighth meeting of the Global WordNet Conference* (cf. p. 145, 146).

Budanitsky, Alexander et Graeme Hirst (2006). « Evaluating WordNet-based Measures of Lexical Semantic Relatedness ». In : *Computational Linguistics* 32.1, p. 13–47.

Buscaldi, Davide et al. (2012). « IRIT : Textual Similarity Combining Conceptual Similarity with an N-gram Comparison Method ». In : *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1 : Proceedings of the Main Conference and the Shared Task, and Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation*. SemEval '12. Montréal, Canada : ACL, p. 552–556. url : http://dl.acm.org/citation.cfm?id=2387636.2387729 (cf. p. 133).

Camacho-Collados, José, Taher Mohammad Pilehvar et Roberto Navigli (2015). « A Unified Multilingual Semantic Representation of Concepts ». In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*. Beijing, China : ACL, p. 741–751. URL : http://aclweb.org/anthology/P15-1072 (cf. p. 47, 55, 128, 129).

Campenhoudt, Marc Van (2000). « Le sens en terminologie ». In : sous la dir. de Thoiron (Ph.) Béjoint. Presses universitaires de Lyon. Chap. De la lexicographie spécialisée à la terminographie : vers un 'métadictionnaire' ?, p. 127–152 (cf. p. 23, 38).

Carletta, Jean et al. (2003). « The NITE XML Toolkit : Flexible annotation for multimodal language data ». In : *Behavior Research Methods, Instruments, & Computers* 35.3, p. 353–363. ISSN : 0743-3808. DOI : 10.3758/BF03195511 (cf. p. 13).

Castilho, Richard Eckart de et Iryna Gurevych (2011). « A Lightweight Framework for Reproducible Parameter Sweeping in Information Retrieval ». In : *Proceedings of the 2011 workshop on Data infrastructurEs for supporting information retrieval evaluation*. Sous la dir. de Maristella Agosti, Nicola Ferro et Costantino Thanos. DESIRE '11. Glasgow, Scotland, UK : ACM, p. 7–10. ISBN : 9781450309523 (cf. p. 140).

Castilho, Richard de et Iryna Gurevych (2014). « A broad-coverage collection of portable NLP components for building shareable analysis pipelines ». In : *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*. Association for Computational Linguistics et Dublin City University. Chap. A broad-co, p. 1–11. URL : http://aclweb.org/anthology/W14-5201 (cf. p. 12).

Chandar A P, Sarath et al. (2014). « An Autoencoder Approach to Learning Bilingual Word Representations ». In : *Advances in Neural Information Processing Systems 27*. Sous la dir. de Z. Ghahramani et al. Curran Associates, Inc., p. 1853–1861. URL : http://papers.nips.cc/paper/5270-an-autoencoder-approach-to-learning-bilingual-word-representations.pdf (cf. p. 128).

Charalampous, Konstantinos et Antonios Gasteratos (2014). « A Tensor-Based Deep Learning Framework ». In : *Image and Vision Computing* 32.11, p. 916–929. ISSN : 02628856. DOI : 10.1016/j.imavis.2014.08.003.

Cimiano, P. et al. (2011). « LexInfo : A declarative model for the lexicon-ontology interface ». In : *Web Semantics : Science, Services and Agents on the World Wide Web* 9.1, p. 29–51. ISSN : 15708268. DOI : 10.1016/j.websem.2010.11.001.

Cohen, William W., Pradeep Ravikumar et Stephen E. Fienberg (2003). « A Comparison of String Distance Metrics for Name-Matching Tasks. » In : *Proceedings of IJCAI-03 Workshop on Information Integration*, p. 73–78 (cf. p. 43, 118).

Cohen, William W, Pradeep Ravikumar et Stephen E Fienberg (2003). « A Comparison of String Distance Metrics for Name-Matching Tasks. » In : *Proceedings of IJCAI-03 Workshop on Information Integration*, p. 73–78.

Collins (1998). *Cobuild English Dictionary*. Harper Collins Publishers.

Cowie, Jim, Joe Guthrie et Louise Guthrie (1992). « Lexical disambiguation using simulated annealing ». In : *COLING '92*. Nantes, France : ACL, p. 359–365. DOI : 10.3115/992066.992125.

Cunningham, Hamish (2002). « GATE, a General Architecture for Text Engineering ». In : *Computers and the Humanities* 36.2, p. 223–254. ISSN : 0010-4817. DOI : 10.1023/A:1014348124664 (cf. p. 12).

Davey, B. A. et H. A. Priestley (2002). *Introduction to Lattices and Order*. Cambridge University Press. doi : 10.2277/0521784514 (cf. p. 78).

Di Marco, Antonio et Roberto Navigli (2013). « Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction ». In : *Computational Linguistics* 39.3, p. 709–754. issn : 0891-2017. doi : 10.1162/COLI_a_00148.

Dice, Lee R. (1945). « Measures of the amount of ecologic association between species ». In : *Ecology* 26.3, p. 297–302.

Diestel, Reinhard (2005). *Graph Theory*. Sous la dir. de Springer-Verlag. 3rd. Berlin, New York : Springer-Verlag (cf. p. 86).

Dohrn, Hannes et Dirk Riehle (2011). « Design and Implementation of the Sweble Wikitext Parser : Unlocking the Structured Data of Wikipedia ». In : *Proceedings of WikiSym '11*, p. 1–10. url : http://sweble.org/downloads/diwp-preprint.pdf.

Dohrn, Hannes et Dirk Riehle (2013). « Design and Implementation of Wiki Content Transformations and Refactorings ». In : *Proceedings of WikiSym '13*. url : http://sweble.org/downloads/diwctr-final.pdf.

Edmonds, Philip et Graeme Hirst (2002). « Near-Synonymy and Lexical Choice ». In : *Computational Linguistics* 28.2, p. 105–144. issn : 0891-2017. doi : 10.1162/089120102760173625.

Eftekhar, Milad et Nick Koudas (2013). « Partitioning and Ranking Tagged Data Sources. » In : *PVLDB* 6.4, p. 229–240. url : http://dblp.uni-trier.de/db/journals/pvldb/pvldb6.html%5C#EftekharK13.

Ehrmann, Maud et al. (2014). « Representing Multilingual Data as Linked Data : the Case of BabelNet 2.0 ». In : *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014*. P. 401–408 (cf. p. 35).

Eppstein, David, Maarten Löffler et Darren Strash (2010). « Listing All Maximal Cliques in Sparse Graphs in Near-optimal Time ». In : *CoRR* abs/-1006.5440. url : http://arxiv.org/abs/1006.5440 (cf. p. 90, 92).

Erdös, Paul (1966). « On cliques in graphs ». In : *Israel Journal of Mathematics* 4.4, p. 233–234. issn : 1565-8511. doi : 10.1007/BF02771637 (cf. p. 86).

ESPINASSE, Bernard et Erwan TRANVOUEZ (2010). *Cours sur les "METHODES ET OUTILS POUR L'AIDE A LA DECISION"*. url : http://www.lsis.org/espinasseb/Syllabus/syllabus%5C_moad.html.

Etzioni, Oren et al. (2007). « Lexical Translation with Application to Image Search on the Web ». In : *Machine Translation Summit XI*. url : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.7536%5C&rep=rep1%5C&type=pdf.

Faruqui, Manaal et al. (2015). « Retrofitting Word Vectors to Semantic Lexicons ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. Denver, Colorado : ACL, p. 1606–1615. url : http://aclweb.org/anthology/N15-1184 (cf. p. 55).

Farwell, David, Louise Guthrie et Yorick Wilks (1992). « The Automatic Creation of Lexical Entries for a Multilingual MT System ». In : *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*. COLING '92. Nantes, France : ACL, p. 532–538. doi : 10.3115/992133.992153 (cf. p. 57).

Farwell, David, Louise Guthrie et Yorick Wilks (1993). « Automatically creating lexical entries for ULTRA, a multilingual MT system ». In : *Machine*

Translation 8.3, p. 127–145. ISSN : 0922-6567. DOI : 10.1007/BF00982636 (cf. p. 57).

FELLBAUM, C. (1997). *WordNet : An Electronic Lexical Database and Some of its Applications*. The MIT Press. ISBN : 026206197X. URL : http://www.worldcat.org/isbn/026206197X (cf. p. 28).

FELLBAUM, C. (1998). *WordNet : An Electronic Lexical Database*. Illustrated edition. The MIT Press. ISBN : 026206197X. URL : http://www.worldcat.org/isbn/026206197X (cf. p. 14, 18, 28).

FERRÁNDEZ, Óscar et al. (2010). « Aligning FrameNet and WordNet based on Semantic Neighborhoods ». In : *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Sous la dir. de Nicoletta Calzolari (Conference CHAIR) et al. Valletta, Malta : European Language Resources Association (ELRA), p. 19–21 (cf. p. 53).

FERRERO, Jérémy et al. (2016). « A Multilingual, Multi-style and Multi-granularity Dataset for Cross-language Textual Similarity Detection ». In : *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Sous la dir. de Nicoletta Calzolari (Conference CHAIR) et al. Portorož, Slovenia : European Language Resources Association (ELRA). ISBN : 9782951740891 (cf. p. 133).

FERRUCCI, David et Adam LALLY (2004). « UIMA : An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment ». In : *Nat. Lang. Eng.* 10.3-4, p. 327–348. ISSN : 1351-3249. DOI : 10.1017/S1351324904003523 (cf. p. 12).

FITSCHEN, Arne (2004). « Ein Computerlinguistisches Lexikon als komplexes System ». Thèse de doct. Université de Stuttgart (cf. p. 37).

FLATI, Tiziano et Roberto NAVIGLI (2012). « The CQC algorithm : Cycling in graphs to semantically enrich and enhance a bilingual dictionary ». In : *Journal of Artificial Intelligence Research* 43, p. 135–171. ISSN : 10450823. DOI : 10.1613/jair.3456 (cf. p. 51).

FLATI, Tiziano, Daniele VANNELLA et al. (2014). « Two Is Bigger (and Better) Than One : the Wikipedia Bitaxonomy Project ». In : *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. Baltimore, Maryland : ACL, p. 945–955 (cf. p. 31, 36).

FORT, Karën et al. (2014). « Crowdsourcing for Language Resource Development : Criticisms About Amazon Mechanical Turk Overpowering Use ». In : *Human Language Technology Challenges for Computer Science and Linguistics*. Sous la dir. de Zygmunt VETULANI et Joseph MARIANI. T. 8387. Lecture Notes in Computer Science. Springer International Publishing, p. 303–314. ISBN : 9783319089577. DOI : 10.1007/978-3-319-08958-4_25 (cf. p. 29).

FRANCIS, W Nelson et Henry KUCERA (1979). *Brown corpus manual* (cf. p. 28).

FRANCOPOULO, Gil et al. (2006). « Lexical Markup Framework ({LMF}) for {NLP} Multilingual Resources ». In : *Proceedings of the Workshop on Multilingual Language Resources and Interoperability*. Sydney, Australia : ACL, p. 1–8. URL : http://www.aclweb.org/anthology/W/W06/W06-1001 (cf. p. 14, 17).

FU, Bo, Rob BRENNAN et Declan O'SULLIVAN (2009). « Cross-Lingual Ontology Mapping – An Investigation of the Impact of Machine Translation ». In : *The Semantic Web : Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings*. Sous la dir. d'Asunción GÓMEZ-PÉREZ, Yong YU et Ying DING. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 1–15. ISBN :

9783642108716. DOI : 10.1007/978-3-642-10871-6_1. URL : http://dx.doi.org/10.1007/978-3-642-10871-6_1 (cf. p. 126).

FYSHE, Alona et al. (2012). « A Compositional and Interpretable Semantic Space ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 32–41.

GAO, Jian-bo, Bao-wen ZHANG et Xiao-hua CHEN (2015). « Engineering Applications of Artificial Intelligence A WordNet-based semantic similarity measurement combining edge-counting and information content theory ». In : *Engineering Applications of Artificial Intelligence* 39, p. 80–88. ISSN : 0952-1976. DOI : 10.1016/j.engappai.2014.11.009.

GAROUFI, Konstantina, Torsten ZESCH et Iryna GUREVYCH (2008). « Representational Interoperability of Linguistic and Collaborative Knowledge Bases ». In : *Proceedings of the KONVENS Workshop on Lexical-Semantic and Ontological Resources – Maintenance, Representation, and Standards*. Berlin, Germany (cf. p. 140).

GAUME, Bruno et al. (2014). « Mesurer la similarité structurelle entre réseaux lexicaux ». In : *Actes de la 21ème conférence Traitement Automatique des Langues Naturelles, Marseille, 2014* (cf. p. 44).

GOIKOETXEA, Josu, Aitor SOROA et Eneko AGIRRE (2015). « Random Walks and Neural Network Language Models on Knowledge Bases ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1434–1439.

GONZALO, Julio, Irina CHUGUR et Felisa VERDEJO (2000). « Sense Clusters for Information Retrieval : Evidence from Semcor and the EuroWordNet InterLingual Index ». In : *Proceedings of the ACL-2000 Workshop on Word Senses and Multi-Linguality*. WorkSense '00. Stroudsburg, PA, USA : ACL, p. 10–18. URL : http://dl.acm.org/citation.cfm?id=1628308.1628310.

GOUWS, Stephan, Yoshua BENGIO et Greg CORRADO (2015). « BilBOWA : Fast Bilingual Distributed Representations without Word Alignments ». In : *Proceedings of the 32nd International Conference on Machine Learning*. T. 37. JMLR : W&CP (cf. p. 55, 128).

GOUWS, Stephan et Anders SÖGAARD (2015). « Simple task-specific bilingual word embeddings ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1386–1390.

GOUWS, Stephan et Anders SØGAARD (2015). « Simple task-specific bilingual word embeddings ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. Denver, Colorado : ACL, p. 1386–1390. URL : http://aclweb.org/anthology/N15-1157 (cf. p. 128).

GUREVYCH, Iryna, Judith ECKLE-KOHLER, Silvana HARTMANN, Michael MATUSCHEK, Christian M. MEYER et al. (2012). « UBY - A Large-Scale Unified Lexical-Semantic Resource Based on LMF ». In : *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*. Avignon, France, p. 580–590 (cf. p. 140).

GUREVYCH, Iryna, Judith ECKLE-KOHLER, Silvana HARTMANN, Michael MATUSCHEK, Christian M MEYER et al. (2012). « Uby : A large-scale unified lexical-semantic resource based on LMF ». In : *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. ACL, p. 580–590 (cf. p. 31, 36).

Gurevych, Iryna, Max Mühlhäuser et al. (2007). « Darmstadt Knowledge Processing Repository Based on UIMA ». In : *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology*. Tübingen, Germany (cf. p. 140).

Haffari, Gholamreza, Ajay Nagesh et Ganesh Ramakrishnan (2015). « Optimizing Multivariate Performance Measures for Learning Relation Extraction Models ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 892–900.

Halliday, M. A. et R. Hasan (1976). *Cohesion in English*. Longman Group Ltd, London, U.K.

Hamp, Birgit et Helmut Feldweg (1997). « GermaNet - a Lexical-Semantic Net for German. » In : *Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. Madrid (cf. p. 37).

Hanoka, Valérie et Benoît Sagot (2012). « WordNet extension made simple : A multilingual lexicon-based approach using wiki resources ». In : *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Sous la dir. de Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey : European Language Resources Association (ELRA), p. 23–25 (cf. p. 54).

Hauberg, Söeren, Oren Freifeld et Mj Black (2012). « A Geometric take on Metric Learning. » In : *Advances in Neural Information Processing Systems*. 1, p. 2024–2032. isbn : 9781627480031. url : https://papers.nips.cc/paper/4539-a-geometric-take-on-metric-learning.pdf.

Hellmann, Sebastian, Jonas Brekle et Sören Auer (2013). « Leveraging the Crowdsourcing of Lexical Resources for Bootstrapping a Linguistic Data Cloud ». In : *Semantic Technology* 7774, p. 191–206.

Hellmann, Sebastian, Jens Lehmann et al. (2013). « Integrating NLP using Linked Data ». In : *Proceedings of the 12th International Semantic Web Conference*. Sydney, Australia (cf. p. 14).

Hirst, G. et David D. St-Onge (1998). « WordNet : An electronic Lexical Database. » In : sous la dir. de C. Fellabaum. Ed. MIT Press. MIT Press. Chap. Lexical chains as representations of context for the detection and correction of malapropisms, p. 305–332 (cf. p. 47).

Hopcroft, John et Robert Tarjan (1973). « Algorithm 447 : efficient algorithms for graph manipulation ». In : *Communications of the ACM* 16.6, p. 372–378. doi : 10.1145/362248.362272 (cf. p. 82).

Hutchinson, Brian, Li Deng et Dong Yu (2013). « Tensor deep stacking networks ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, p. 1944–1957. doi : 10.1109/TPAMI.2012.268.

Iacobacci, Ignacio, Taher Mohammad Pilehvar et Roberto Navigli (2015). « SensEmbed : Learning Sense Embeddings for Word and Relational Similarity ». In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*. Beijing, China : ACL, p. 95–105. url : http://aclweb.org/anthology/P15-1010 (cf. p. 47, 128, 129).

Ide, Nancy et Keith Suderman (2014). « The Linguistic Annotation Framework : a standard for annotation interchange and merging ». In : *Language Resources*

and Evaluation 48.3, p. 395–418. ISSN : 1574-020X. DOI : 10.1007/s10579-014-9268-1 (cf. p. 13).

IDE, Nancy et Jean VERONIS (1998). « Word Sense Disambiguation : The State of the Art ». In : *Computational Linguistics* 24, p. 1–40.

JIANG, J. J. et D. W. CONRATH (1997). « Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy ». In : *ROCLING X* (cf. p. 47).

JIMENEZ, Sergio, Claudia BECERRA et Alexander GELBUKH (2012). « Soft Cardinality : A Parameterized Similarity Function for Text Comparison ». In : **SEM 2012 : The First Joint Conference on Lexical and Computational Semantics ,Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)* (cf. p. 112, 117).

JIMENEZ, Sergio, Fabio GONZALEZ et Alexander GELBUKH (2010). « Text comparison using soft cardinality ». In : *Proceedings of the 17th International Conference on String Processing and Information Retrieval*. Los Cabos, Mexico : Springer-Verlag, p. 297–302. URL : http://dl.acm.org/citation.cfm?id=1928328.1928367 (cf. p. 112, 117).

JOHANSSON, Richard et Luis NIETO (2015). « Embedding a Semantic Network in a Word Space ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1428–1433.

JURGENS, David et Roberto NAVIGLI (2014). « It's All Fun and Games until Someone Annotates : Video Games with a Purpose for Linguistic Annotation ». In : *Transactions of the Association for Computational Linguistics* 2, p. 449–464. URL : https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/421 (cf. p. 34).

JURGENS, David et Mohammad Taher PILEHVAR (2015). « Reserating the awesometastic : An automatic extension of the WordNet taxonomy for novel terms ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. T. 00, p. 1459–1465.

JURGENS, David, Mohammad Taher PILEHVAR et Roberto NAVIGLI (2014). « SemEval-2014 Task 3 : Cross-Level Semantic Similarity ». In : *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland : Association for Computational Linguistics et Dublin City University, p. 17–26. URL : http://www.aclweb.org/anthology/S14-2003 (cf. p. 35).

KAMHOLZ, David, Jonathan POOL et Susan M. COLOWICK (2014). « PanLex : Building a Resource for Panlingual Lexical Translation ». In : *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland : European Language Resources Association (ELRA), p. 3145–3150. URL : http://www.lrec-conf.org/proceedings/lrec2014/pdf/1029%5C_Paper.pdf (cf. p. 38, 54).

KANG, Jaewoo et Jeffrey F. NAUGHTON (2003). « On Schema Matching with Opaque Column Names and Data Values ». In : *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. SIGMOD '03. San Diego, California : ACM, p. 205–216. ISBN : 158113634X. DOI : 10.1145/872757.872783 (cf. p. 44).

KATAJAINEN, Jyrki et Jesper Larsson TRÄFF (1997). « Algorithms and Complexity : Third Italian Conference, CIAC '97 Rome, Italy, March 12–14, 1997 Proceedings ». In : sous la dir. de Giancarlo BONGIOVANNI, Daniel Pierre BOVET et Giuseppe BATTISTA. Berlin, Heidelberg : Springer Berlin Heidel-

berg. Chap. A meticulous analysis of mergesort programs, p. 217–228. ISBN : 9783540683230. DOI : 10.1007/3-540-62592-5_74.

KIPPER, Karin et al. (2006). « Extensive Classifications of English verbs ». In : *Proceedings of the 12th EURALEX International Congress*. Turin, Italy (cf. p. 37).

KLEIN, Dan et Christopher D. MANNING (2003). « Accurate Unlexicalized Parsing ». In : *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. ACL '03. Sapporo, Japan : ACL, p. 423–430. DOI : 10.3115/1075096.1075150. URL : http://dx.doi.org/10.3115/1075096.1075150 (cf. p. 12).

KLYNE, Graham et Jeremy J CARROLL (2004). *Resource Description Framework (RDF) : Concepts and Abstract Syntax*. Sous la dir. de Graham KLYNE et Jeremy CARROLL. URL : http://www.w3.org/TR/rdf-concepts/.

KRIZHANOVSKY, A A (2010). « Transformation of Wiktionary entry structure into tables and relations in a relational database schema ». In : *arXiv preprint arXiv :1011.1368*. arXiv : 1011.1368.

KRIZHANOVSKY, A.A. et A.V. SMIRNOV (2013). « An approach to automated construction of a general-purpose lexical ontology based on Wiktionary ». In : *Journal of Computer and Systems Sciences International* 52.2, p. 215–225. ISSN : 1064-2307. DOI : 10.1134/S1064230713020068 (cf. p. 31).

LAFOURCADE, Mathieu (2006). « Conceptual Vector Learning Comparing Bootstrapping from a Thesaurus or Induction by Emergence ». In : *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'06), European Language Resources Association (ELRA)* (cf. p. 47, 62).

LAFOURCADE, Mathieu et F. GUINAND (2005). *Ants for Natural Language Processing*. Rapp. tech. 05046. LIRMM, Montpellier, 22 p. URL : http://hal-lirmm.ccsd.cnrs.fr/lirmm-00106692 (cf. p. 62, 63).

LAFOURCADE, Mathieu et Alain JOUBERT (2013). « Ressources Lexicales : Contenu, construction, utilisation, évaluation. » In : sous la dir. de Núria GALA et Michael ZOCK. T. Supplementa 30. Linguisticae Investigationes. John Benjamins. Chap. Bénéfices et limites de l'acquisition lexicale dans l'expérience JeuxDeMots, p. 187–216 (cf. p. 34).

LAFOURCADE, Mathieu, Alain JOUBERT et Nathalie Le BRUN (2015). *Games with a Purpose (GWAPS)*. Focus, Cognitive Science and Knowledge Management. Wiley-ISTE (cf. p. 29, 34).

LAFOURCADE, Mathieu et Eugène STANFORD (1999). « Analyse et désambiguïsation lexicale par vecteurs sémantiques ». In : *Actes de TALN'99*. Sous la dir. d'ATALA (cf. p. 39).

LAPALME, Guy et Gilles SÉRASSET (2003). « Batch creation of Papillon entries from DiCo ». In : *Papillon 2003 Workshop*. Sapporo, Japan. URL : https://hal.archives-ouvertes.fr/hal-00340726 (cf. p. 28).

LAPARRA, Egoitz, German RIGAU et Montse CUADROS (2010). « Exploring the Integration of WordNet and FrameNet ». In : *Proceedings of the 5th Global WordNet Conference (GWC 2010)* (cf. p. 50).

LEACOCK, C. et M. CHODOROW (1998). « WordNet : An electronic Lexical Database. » In : sous la dir. de C. FELLABAUM. MIT Press. Chap. Combining Local Context and WordNet Similarity for Word Sense Identification, p. 265–283 (cf. p. 47, 50).

LEFEVER, Els et Veronique HOSTE (2010). « SemEval-2010 Task 3 : Cross-lingual Word Sense Disambiguation ». In : *Proceedings of the 5th International Workshop on Semantic Evaluation*. SemEval '10. Los Angeles, California : ACL, p. 15–20.

URL : http://dl.acm.org/citation.cfm?id=1859664.1859667 (cf. p. 127, 137).

Lefrançois, Maxime et Fabien Gandon (2013). « The Unit Graphs Framework : A graph-based Knowledge Representation Formalism designed for the Meaning-Text Theory ». In : August, p. 30–31.

Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer et al. (2014). « DBpedia-a large-scale, multilingual knowledge base extracted from wikipedia ». In : *Semantic Web Journal* 5, p. 1–29 (cf. p. 30).

Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer et Christian Bizer (2015). « DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia ». In : *Semantic Web Journal* 6.2, p. 167–195.

Lerman, I.C. (1981). « Sur quelques questions liées à l'ensemble des partitions sur un ensemble fini ». In : *Classification et Analyse Ordinale des Données*. BORDAS. Paris : Dunod. Chap. Chapitre 0, p. 17–52.

Lesk, Michael (1986). « Automatic Sense Disambiguation Using Machine Readable Dictionaries : How to Tell a Pine Cone from an Ice Cream Cone ». In : *Proceedings of the 5th annual international conference on Systems documentation*, p. 24–26 (cf. p. 45, 47, 116).

Lesnikova, Tatiana, Jérôme David et Jérôme Euzenat (2015). « Interlinking English and Chinese RDF Data Using BabelNet ». In : *Proceedings of the 2015 ACM Symposium on Document Engineering*. DocEng '15. Lausanne, Switzerland : ACM, p. 39–42. ISBN : 9781450333078. DOI : 10.1145/2682571.2797089 (cf. p. 43, 54, 126).

Levinboim, Tomer et David Chiang (2015). « Multi-Task Word Alignment Triangulation for Low-Resource Languages ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1221–1226.

Li, Yuhua, Zuhair A. Bandar et David McLean (2003). « An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources ». In : *IEEE Trans. on Knowl. and Data Eng.* 15.4, p. 871–882 (cf. p. 47).

Lin, Dekang (1998a). « An Information-Theoretic Definition of Similarity ». In : *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA, p. 296–304 (cf. p. 47).

Lin, Dekang (1998b). « Automatic Retrieval and Clustering of Similar Words ». In : *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*. ACL '98. Montreal, Quebec, Canada : ACL, p. 768–774. DOI : 10.3115/980691.980696. URL : http://dx.doi.org/10.3115/980691.980696 (cf. p. 131).

Lu, Ang et al. (2015). « Deep Multilingual Correlation for Improved Word Embeddings ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 250–256.

Maarouf, Ismail El et al. (2015). « The GuanXi network : a new multilingual LLOD for Language Learning applications ». In : *Proceedings of the Second Workshop on Natural Language Processing and Linked Open Data*. Hissar, Bulgaria, p. 42–51 (cf. p. 54).

Maedche, Alexander et Steffen Staab (2002). « Measuring Similarity between Ontologies ». In : *Knowledge Engineering and Knowledge Management : Ontologies and the Semantic Web*. Sous la dir. d'Asunción Gómez-Pérez et V.Richard Benjamins. T. 2473. Lecture Notes in Computer Science. Springer Berlin Heidelberg, p. 251–263. isbn : 9783540442684. doi : 10.1007/3-540-45810-7_24 (cf. p. 43).

Magnini, Bernardo et Gabriela Cavaglià (2000). « Integrating Subject Field Codes into WordNet ». In : *Proceeding of the 2nd International Conference on Language Resources & Evaluation*, p. 1413–1418 (cf. p. 18).

Manandhar, Suresh et al. (2010). « SemEval-2010 Task 14 : Word Sense Induction & Disambiguation ». In : *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden : ACL, p. 63–68. url : http://www.aclweb.org/anthology/S10-1011 (cf. p. 39).

Manning, C. et H. Schütze (1999). *Foundations of statistical natural language processing*. Cambridge, MA : MIT Press (cf. p. 112).

Martins, Ronaldo Teixeira et Maria das Graças Volpe Nunes (2005). « Universal Network Language : Advances in Theory and Applications ». In : sous la dir. de J. Cardeñosa, A. Gelbukh et E. Tovar. Research on Computing Science 12. IPN. Chap. On the Aboutness of UNL, p. 51–63 (cf. p. 25).

Matuschek, Michael (2015). « Word Sense Alignment of Lexical Resources ». Thèse de doct. Darmstadt : Technische Universitat Darmstadt. url : http://tuprints.ulb.tu-darmstadt.de/4355/ (cf. p. 42–46, 48, 49, 51–54, 132, 133, 145).

Matuschek, Michael, Christian Meyer et Iryna Gurevych (2013). « Multilingual Knowledge in Aligned Wiktionary and OmegaWiki for Translation Applications ». In : *Translation : Computation, Corpora, Cognition* 3.1, p. 87–118. issn : 2193-6986. url : http://t-c3.org/index.php/t-c3/article/view/20.

Mausam et al. (2009). « Compiling a Massive, Multilingual Dictionary via Probabilistic Inference ». In : *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore : ACL, p. 262–270. url : http://www.aclweb.org/anthology/P/P09/P09-1030 (cf. p. 38, 54, 61, 86).

McCarthy, Diana et Roberto Navigli (2007). « SemEval-2007 Task 10 : English Lexical Substitution Task ». In : *Proceedings of the 4th International Workshop on Semantic Evaluations*. SemEval '07. Prague, Czech Republic : ACL, p. 48–53. url : http://dl.acm.org/citation.cfm?id=1621474.1621483 (cf. p. 136).

McCrae, John, Guadalupe Aguado-de-Cea et al. (2012). « Interchanging lexical resources on the Semantic Web ». In : *Language Resources and Evaluation* 46.4, p. 701–719. issn : 1574-020X. doi : 10.1007/s10579-012-9182-3.

McCrae, John, Dennis Spohr et Philipp Cimiano (2011). « Linking Lexical Resources and Ontologies on the Semantic Web with Lemon ». In : *ESWC*. Sous la dir. de Grigoris Antoniou et al. T. 6643. Lecture Notes in Computer Science. Springer, p. 245–259. isbn : 9783642210334 (cf. p. 17).

McKelvie, David et al. (2001). « The MATE workbench : An annotation tool for XML coded speech corpora ». In : *Speech Communication* 33.12, p. 97–112. issn : 0167-6393. doi : 10.1016/S0167-6393(00)00071-6 (cf. p. 13).

Meersman, Robert, Tharam S Dillon et Pilar Herrero, éds. (2010). *On the Move to Meaningful Internet Systems, OTM 2010 - Confederated International Conferences : CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October*

25-29, 2010, Proceedings, Part II. T. 6427. Lecture Notes in Computer Science. Springer. ISBN : 9783642169489.

Mel'čuk, Igor Aleksandrovič (1999). *Dictionnaire explicatif et combinatoire du français contemporain.* Sous la dir. d'I. Mel'čuk. T. IV. Montréal : Presses de l'Université de Montréal. ISBN : 2760617386. URL : http://opac.inria.fr/record=b1132439 (cf. p. 27, 57).

Mel'čuk, Igor Aleksandrovič (2006). « Explanatory Combinatorial Dictionary ». In : *Open Problems in Linguistic and Lexicography*. Sous la dir. de Giandomenico SICA. Monza (Italy) : Polimetrica, p. 225–355 (cf. p. 57).

Melo, Gerard de et Gerhard Weikum (2008). « Language as a Foundation of the {Semantic Web} ». In : *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC)*. Sous la dir. de Christian Bizer et Anupam Joshi. T. 401. CEUR WS. Karlsruhe, Germany : CEUR (cf. p. 113).

Melo, Gerard de et Gerhard Weikum (2009). « Towards a Universal WordNet by Learning from Combined Evidence ». In : *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM '09. Hong Kong, China : ACM, p. 513–522. ISBN : 9781605585123. DOI : 10.1145/1645953.1646020 (cf. p. 53).

Melo, Gerard de et Gerhard Weikum (2010). « MENTA : Inducing multilingual taxonomies from Wikipedia ». In : *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, p. 1099–1108 (cf. p. 31).

Meng, Lingling, Runqing Huang et Junzhong Gu (2013). « A Review of Semantic Similarity Measures in WordNet ». In : *International Journal of Hybrid Information Technology* 6.1, p. 1–12.

Meyer, Christian M. (2013). « Wiktionary : The Metalexicographic and the Natural Language Processing Perspective ». Creative Commons Attribution Noncommercial No Derivatives. Dissertation. Darmstadt : Technische Universitat Darmstadt.

Meyer, Christian M et Iryna Gurevych (2010). « Worth its Weight in Gold or Yet Another Resource , A Comparative Study of Wiktionary, OpenThesaurus and GermaNet ». In : *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics*. Sous la dir. d'Alexander Gelbukh. T. 6008. Lecture Notes in Computer Science. Berlin/Heidelberg : Springer, p. 38–49 (cf. p. 112).

Meyer, Christian M. et Iryna Gurevych (2011). « What Psycholinguists Know About Chemistry : Aligning Wiktionary and WordNet for Increased Domain Coverage ». In : *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand : Asian Federation of Natural Language Processing, p. 883–892. URL : http://www.aclweb.org/anthology/I11-1099 (cf. p. 48, 49).

Meyer, Christian M et Iryna Gurevych (2012a). « Electronic Lexicography ». In : sous la dir. de S Granger et M Paquot. Oxford University Press. Chap. Wiktionary, (to appear) (cf. p. 111).

Meyer, Christian M. et Iryna Gurevych (2012a). « OntoWiktionary – Constructing an Ontology from the Collaborative Online Dictionary Wiktionary ». In : *Semi-Automatic Ontology Development : Processes and Resources*. Sous la dir. de Maria Teresa Pazienza et Armando Stellato. Hershey, PA : IGI Global. Chap. 6, p. 131–161. ISBN : 9781466601888. DOI : 10.4018/978-1-46660-188-

8. URL : http://www.christian-meyer.org/research/publications/igi-saod2012/ (cf. p. 37).

Meyer, Christian M. et Iryna Gurevych (2012b). « To Exhibit is not to Loiter : A Multilingual, Sense-Disambiguated Wiktionary for Measuring Verb Similarity ». In : *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*. T. 4. Mumbai, India, p. 1763–1780 (cf. p. 115, 121, 122).

Meyer, Christian M et Iryna Gurevych (2012b). « Wiktionary : a new rival for expert-built lexicons ? Exploring the possibilities of collaborative lexicography ». In : *Electronic Lexicography*. Sous la dir. de Sylviane Granger et Magali Paquot. Oxford : Oxford University Press, (to appear).

Mihalcea, Rada, Ravi Sinha et Diana McCarthy (2010). « SemEval-2010 Task 2 : Cross-lingual Lexical Substitution ». In : *Proceedings of the 5th International Workshop on Semantic Evaluation*. SemEval '10. Los Angeles, California : ACL, p. 9–14. URL : http://dl.acm.org/citation.cfm?id=1859664.1859666 (cf. p. 136).

Mihalcea, Rada, Paul Tarau et Elizabeth Figa (2004). « PageRank on Semantic Networks, with Application to Word Sense Disambiguation ». In : *COLING '04*. Geneva, Switzerland : ACL (cf. p. 49).

Mikolov, Tomas et al. (2013). « Distributed Representations of Words and Phrases and their Compositionality ». In : *Advances in Neural Information Processing Systems 26*. Sous la dir. de C.J.C. Burges et al. Curran Associates, Inc., p. 3111–3119. URL : http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf (cf. p. 39).

Miller, George A. (1995). « WordNet : a lexical database for English ». In : *Commun. ACM* 38.11, p. 39–41. ISSN : 0001-0782.

Miller, Tristan, Chris Biemann et al. (2012). « Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation ». In : *Proceedings of COLING 2012*. Mumbai, India : The COLING 2012 Organizing Committee, p. 1781–1796. URL : http://aclweb.org/anthology/C12-1109 (cf. p. 131).

Miller, Tristan, Nicolai Erbs et al. (2013). « {DKPro} {WSD} : A Generalized {UIMA}-based Framework for Word Sense Disambiguation ». In : *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*. Stroudsburg, PA, USA : ACL, p. 37–42 (cf. p. 140).

Miller, Tristan et Iryna Gurevych (2014). « WordNet-Wikipedia-Wiktionary : Construction of a Three-way Alignment ». In : *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Sous la dir. de Nicoletta Calzolari et al. Reykjavik, Iceland : European Language Resources Association (ELRA). ISBN : 9782951740884. URL : http://www.lrec-conf.org/proceedings/lrec2014/pdf/4%5C_Paper.pdf.

Moro, Andrea et Roberto Navigli (2015). « SemEval-2015 Task 13 : Multilingual All-Words Sense Disambiguation and Entity Linking ». In : *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado : ACL, p. 288–297. URL : http://www.aclweb.org/anthology/S15-2049 (cf. p. 36).

Moro, Andrea, Alessandro Raganato et Roberto Navigli (2014). « Entity Linking meets Word Sense Disambiguation : a Unified Approach ». In : *Tran-*

sactions of the Association for Computational Linguistics (TACL) 2, p. 231–244 (cf. p. 54).

Morris, Jane et Graeme Hirst (1991). « Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text ». In : *Comput. Linguist.* 17.1, p. 21–48.

Nakov, Preslav et al., éds. (2015). *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado : ACL. url : http://www.aclweb.org/anthology/S15-2 (cf. p. 13).

Nasiruddin, Mohammad et al. (2015). « Création rapide et efficace d'un système de désambiguïsation lexicale pour une langue peu dotée ». In : *Actes de la 22ème conférence sur le Traitement Automatique des Langues Naturelles*.

Navarro, Emmanuel et al. (2009). « Wiktionary and NLP : Improving synonymy networks ». In : *Proceedings of the 2009 Workshop on The People,'s Web Meets NLP : Collaboratively Constructed Semantic Resources (People's Web)*. Sous la dir. d'Iryna Gurevych et Torsten Zesch. Suntec, Singapore : ACL, p. 19–27. url : http://www.aclweb.org/anthology/W/W09/W09-3303.pdf.

Navigli, R. et M. Lapata (2010). « An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation ». In : *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.4, p. 678–692. issn : 0162-8828. doi : 10.1109/TPAMI.2009.36 (cf. p. 45).

Navigli, Roberto (2006). « Meaningful Clustering of Senses Helps Boost Word Sense Disambiguation Performance ». In : *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics joint with the 21st International Conference on Computational Linguistics (COLING-ACL)*. Sydney, Australia, p. 105–112 (cf. p. 18, 19).

Navigli, Roberto (2009a). « Using Cycles and Quasi-cycles to Disambiguate Dictionary Glosses ». In : *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '09. Athens, Greece : ACL, p. 594–602. url : http://dl.acm.org/citation.cfm?id=1609067.1609133 (cf. p. 51).

Navigli, Roberto (2009b). « Word sense disambiguation : A survey ». In : *ACM Comput. Surv.* 41.2, 10 :1–10 :69. issn : 0360-0300. doi : 10.1145/1459352.1459355.

Navigli, Roberto (2009c). « Word sense disambiguation : A survey ». In : *ACM Comput. Surv.* 41.2, 10 :1–10 :69. issn : 0360-0300.

Navigli, Roberto, David Jurgens et Daniele Vannella (2013). « SemEval-2013 Task 12 : Multilingual Word Sense Disambiguation ». In : *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, p. 222–231. url : http://www.aclweb.org/anthology/S13-2040 (cf. p. 35).

Navigli, Roberto, Kenneth C. Litkowski et Orin Hargraves (2007). « SemEval-2007 Task 07 : Coarse-Grained English All-Words Task ». In : *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic : ACL, p. 30–35. url : http://www.aclweb.org/anthology/S/S07/S07-1006 (cf. p. 21).

Navigli, Roberto et Simone Paolo Ponzetto (2012a). « BabelNet : The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network ». In : *Artificial Intelligence* 193, p. 217–250. doi : doi:10.1016/j.artint.2012.07.001 (cf. p. 35, 51, 52).

Navigli, Roberto et Simone Paolo Ponzetto (2012b). « BabelRelate ! A Joint Multilingual Approach to Computing Semantic Relatedness ». In : *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*. Toronto, Canada (cf. p. 50).

Navigli, Roberto et Daniele Vannella (2013). « SemEval-2013 Task 11 : Word Sense Induction and Disambiguation within an End-User Application ». In : *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA : ACL, p. 193–201. URL : http://www.aclweb.org/anthology/S13-2035 (cf. p. 39).

Navigli, Roberto et Paola Velardi (2005). « Structural Semantic Interconnections : A Knowledge-Based Approach to Word Sense Disambiguation ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.7, p. 1075–1086. ISSN : 0162-8828. DOI : 10.1109/TPAMI.2005.149 (cf. p. 45).

Nguyen, Hong-Thai, Christian Boitet et Gilles Sérasset (2007). « PIVAX, an online contributive lexical database for heterogeneous MT systems using a lexical pivot ». In : *Proceedings of SNLP 2007* (cf. p. 25).

Niemann, Elisabeth et Iryna Gurevych (2011). « The People's Web meets Linguistic Knowledge : Automatic Sense Alignment of Wikipedia and WordNet ». In : *Proceedings of the Ninth International Conference on Computational Semantics*. Oxford, UK, p. 205–214 (cf. p. 49).

Ogren, Philip V et Steven J Bethard (2009). « Building Test Suites for UIMA Components ». In : *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*. SETQA-NLP '09. Stroudsburg, PA, USA : ACL, p. 1–4. ISBN : 9781932432329. URL : http://dl.acm.org/citation.cfm?id=1621947.1621948 (cf. p. 12).

Pennec, X (2007). *Statistical Computing on Manifolds for Computational Anatomy*. Habilitation à Diriger La Recherche. Université Nice Sophia Antipolis, p. 1–16.

Pennec, Xavier (2006). « Intrinsic statistics on Riemannian manifolds : Basic tools for geometric measurements ». In : *Journal of Mathematical Imaging and Vision* 25.1, p. 127–154. ISSN : 09249907. DOI : 10.1007/s10851-006-6228-4.

Pennec, Xavier, Pierre Fillard et Nicholas Ayache (2006). « A riemannian framework for tensor computing ». In : *International Journal of Computer Vision* 66.1, p. 41–66. DOI : 10.1007/s11263-005-3222-z.

Pereira, Fernando, Naftali Tishby et Lillian Lee (1993). « Distributional Clustering Of English Words ». In : *Proceedings of the ACL*, p. 183–190 (cf. p. 131).

Pianta, Emanuele, Luisa Bentivogli et Christian Girardi (2002). « MultiWordNet : developing an aligned multilingual database ». In : *Proceedings of the First International Conference on Global WordNet*. Mysore, India, p. 21–25 (cf. p. 14).

Pilehvar, Mohammad Taher (2015). « NASARI : a Novel Approach to a Semantically-Aware Representation of Items ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 567–577.

Pilehvar, Mohammad Taher et Roberto Navigli (2014). « A Robust Approach to Aligning Heterogeneous Lexical Resources ». In : *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Baltimore, Maryland : ACL, p. 468–478. URL : http://www.aclweb.

org/anthology/P/P14/P14-1044%20http://acl2014.org/acl2014/P14-1/pdf/P14-1044.pdf (cf. p. 52, 53, 75, 131).

Pilehvar, Mohammad Taher et Roberto Navigli (2015a). « An Open-source Framework for Multi-level Semantic Similarity Measurement ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 76–80.

Pilehvar, Mohammad Taher et Roberto Navigli (2015b). « From Senses to Texts : An All-in-one Graph-based Approach for Measuring Semantic Similarity ». In : *Artificial Intelligence* 228. Sous la dir. d'Elsevier, p. 95–128 (cf. p. 130).

Pilehvar, Taher Mohammad, David Jurgens et Roberto Navigli (2013). « Align, Disambiguate and Walk : A Unified Approach for Measuring Semantic Similarity ». In : *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. Sofia, Bulgaria : ACL, p. 1341–1351. url : http://aclweb.org/anthology/P13-1132 (cf. p. 47, 140).

Pirró, Giuseppe et Jérôme Euzenat (2010). « A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness ». In : *Proceedings of the 9th International Semantic Web Conference 'ISWC 2010*. T. 6496. Lecture Notes in Computer Science. Springer Verlag, p. 615–630 (cf. p. 43, 47).

Pirrò, Giuseppe et Jérôme Euzenat (2010). « A Semantic Similarity Framework Exploiting Multiple Parts-of Speech ». In : *OTM Conferences (2)*. Sous la dir. de Robert Meersman, Tharam S Dillon et Pilar Herrero. T. 6427. Lecture Notes in Computer Science. Springer, p. 1118–1125. isbn : 9783642169489 (cf. p. 116).

Ponzetto, Simone Paolo et Roberto Navigli (2009). « Large-scale Taxonomy Mapping for Restructuring and Integrating Wikipedia ». In : *Proceedings of the 21st International Jont Conference on Artifical Intelligence*. IJCAI'09. Pasadena, California, USA : Morgan Kaufmann Publishers Inc., p. 2083–2088. url : http://dl.acm.org/citation.cfm?id=1661445.1661778 (cf. p. 50).

Quemada, B (1985). « Les noms des mots ou des noms pour les mots. À propos de la terminologie lexicologique ». In : *Linguistica computazionale*. Studies in Honor of Roberto Busa S.J. IV, V.

Rada, R. et al. (1989). « Development and Application of a Metric on Semantic Nets ». In : *IEEE Transactions on Systems, Man, and Cybernetics* 19.1, p. 17–30 (cf. p. 47, 50).

Resnik, Philip (1995). « Using Information Content to Evaluate Semantic Similarity in a Taxonomy ». In : *IJCAI'95*. Montreal, Canada, p. 448–453 (cf. p. 47).

Rogers, D. et T. Tanimoto (1960). « A Computer Program for Classifying Plants ». In : *Science* 132.3434, p. 1115–1118 (cf. p. 47).

Roget (1989). *New Roget's Thesaurus*. P.S.I. isbn : 9780938261407.

Ruiz-Casado, Maria, Enrique Alfonseca et Pablo Castells (2005). « Advances in Web Intelligence : Third International Atlantic Web Intelligence Conference, AWIC 2005, Lodz, Poland, June 6-9, 2005. Proceedings ». In : sous la dir. de Piotr S. Szczepaniak, Janusz Kacprzyk et Adam Niewiadomski. Berlin, Heidelberg : Springer Berlin Heidelberg. Chap. Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets, p. 380–386. isbn : 9783540319009. doi : 10.1007/11495772_59 (cf. p. 49).

Sabou, Marta, Kalina Bontcheva et Arno Scharl (2012). « Crowdsourcing Research Opportunities : Lessons from Natural Language Processing ». In : *Proceedings of the 12th International Conference on Knowledge Management and*

Knowledge Technologies. i-KNOW '12. Graz, Austria : ACM, 17 :1–17 :8. ISBN : 9781450312424. DOI : 10.1145/2362456.2362479 (cf. p. 29).

SAJOUS, Franck et Nabil HATHOUT (2015). « GLAWI, a free XML-encoded Machine-Readable Dictionary built from the French Wiktionary ». In : *Proceedings of the of the eLex 2015 conference*. Herstmonceux, England, p. 405–426 (cf. p. 31).

SAJOUS, Franck, Emmanuel NAVARRO et al. « Advances in Natural Language Processing : 7th International Conference on NLP, IceTAL 2010, Reykjavik, Iceland, August 16-18, 2010 ». In : sous la dir. de Hrafn LOFTSSON, Eiríkur RÖGNVALDSSON et Sigrún HELGADÓTTIR (cf. p. 54).

SALTON, Gerard (1983). *Introduction to Modern Information Retrieval*. Sous la dir. de McGraw Hill Higher EDUCATION. Mcgraw Hill Computer Science Series. Mcgraw-Hill College. ISBN : 1856044807 (cf. p. 39).

SAUSSURE, Ferdinand de (1983). *Course in General Linguistics*. London : Duckworth.

SCHAFFER, R. et R. SEDGEWICK (1993). « The Analysis of Heapsort ». In : *Journal of Algorithms* 15.1, p. 76–100. ISSN : 0196-6774. DOI : 10.1006/jagm.1993.1031.

SCHWAB, Didier (2005). « 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 doct. Université de Montpellier (cf. p. 49).

SCHWAB, Didier, Lim Lian TZE et Mathieu LAFOURCADE (2007). « Nommage de sens à l'aide des vecteurs conceptuels ». In : *Actes de TALN'2007*. Sous la dir. d'ATALA. Toulouse (cf. p. 39).

SECO, Nuno, Tony VEALE et Jer HAYES (2004). « An Intrinsic Information Content Metric for Semantic Similarity in WordNet ». In : *Proceedings of ECAI'2004*. Valencia, Spain, p. 1089–1090.

SENNRICH, Rico et Beat KUNZ (2014). « Zmorge : A German Morphological Lexicon Extracted from Wiktionary ». In : *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Sous la dir. de Nicoletta CALZOLARI et al. Reykjavik, Iceland : European Language Resources Association (ELRA). ISBN : 9782951740884. URL : http://www.lrec-conf.org/proceedings/lrec2014/pdf/116%5C_Paper.pdf.

SÉRASSET, Gilles (1994). « Interlingual Lexical Organisation for Multilingual Lexical Databases in NADIA ». In : *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*. COLING '94. Kyoto, Japan : ACL, p. 278–282. DOI : 10.3115/991886.991933 (cf. p. 17, 23, 33, 57–59, 78, 81).

SÉRASSET, Gilles (2012). « Dbnary : Wiktionary as a LMF based Multilingual RDF network ». In : *Proceeding of the Eighth Language Resources and Evaluation Conference, LREC 2012*. Istanbul. URL : citeulike-article-id:11487829%20http://www.lrec-conf.org/proceedings/lrec2012/pdf/387%5C_Paper.pdf.

SÉRASSET, Gilles (2015). « Dbnary : Wiktionary as a Lemon Based RDF Multilingual Lexical Resource ». In : *Semantic Web Journal - Special issue on Multilingual Linked Open Data* 6.4, p. 355–361. DOI : 10.3233/SW-140147 (cf. p. 32, 33).

SÉRASSET, Gilles et Andon TCHECHMEDJIEV (2014). « Dbnary : Wiktionary as Linked Data for 12 Language Editions with Enhanced Translation Relations ». In : *3rd Workshop on Linked Data in Linguistics : Multilingual Knowledge Resources and Natural Language Processing*. Reyjkjavik. URL : http://www.lrec-

conf.org/proceedings/lrec2014/workshops/LREC2014Workshop-LDL2014%20Proceedings.pdf.

Silva, Mário J et al., éds. (2007). *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*. ACM. isbn : 9781595938039.

Smucker, Mark D, James Allan et Ben Carterette (2007). « A comparison of statistical significance tests for information retrieval evaluation ». In : *CIKM*. Sous la dir. de Mário J Silva et al. ACM, p. 623–632. isbn : 9781595938039.

Song, Yangqiu et Dan Roth (2015). « Unsupervised Sparse Vector Densification for Short Text Similarity ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1275–1280.

Steinlin, Jacques et al. (2004). « De l'article lexicographique à la modélisation objet du dictionnaire et des liens lexicaux ». In : *Proceedings of the 11th EURALEX International Congress*. Sous la dir. de Geoffrey Williams et Sandra Vessier. Lorient, France : Université de Bretagne-Sud, Faculté des lettres et des sciences humaines, p. 177–186. isbn : 2952245703 (cf. p. 16, 27, 33).

Stevenson, Angus (2010). *Oxford dictionary of English*. Oxford, United Kingdom : Oxford University Press (cf. p. 18, 27).

Šuster, Simon, Ivan Titov et Gertjan van Noord (2016). « Bilingual Learning of Multi-sense Embeddings with Discrete Autoencoders ». In : *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*. San Diego, California : ACL, p. 1346–1356. url : http://aclweb.org/anthology/N16-1160 (cf. p. 128).

Tchechmedjiev, Andon (2012). « État de l'art : mesures de similarité sémantique locales et algorithmes globaux pour la désambiguïsation lexicale à base de connaissances (State of the art : Local Semantic Similarity Measures and Global Algorithmes for Knowledge-based Word Sense Disambiguation). » In : *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 3 : RECITAL,ATALA/AFCP,* Grenoble, p. 295–308 (cf. p. 42–44, 116).

Tchechmedjiev, Andon et al. (2014). « Attaching Translations to Proper Lexical Senses in DBnary ». In : *3rd Workshop on Linked Data in Linguistics : Multilingual Knowledge Resources and Natural Language Processing*. url : http://www.lrec-conf.org/proceedings/lrec2014/workshops/LREC2014Workshop-LDL2014%20Proceedings.pdf (cf. p. 45, 110).

Teeraparbseree, Aree (2005). « Méthodes et outils pour la création automatique et l'évaluation de structures de bases lexicales multilingues (symétriques) à lexies et axies ». Thèse de doct. Universié Joseph Fourier (cf. p. 59–62, 64–67, 81, 129, 135, 145).

Toral, Antonio, Rafael Muñoz et Monica Monachini (2008). « Named Entity WordNet ». In : *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Sous la dir. de Nicoletta Calzolari (Conference Chair) et al. http ://www.lrec-conf.org/proceedings/lrec2008/. Marrakech, Morocco : European Language Resources Association (ELRA). isbn : 2951740840 (cf. p. 50).

Tversky, Amos (1977a). « Features of Similarity ». In : *Psychological Review* 84.4, p. 327–352 (cf. p. 47).

Tversky, Amos (1977b). « Features of Similarity ». In : *Psychological Review*. T. 84. 4, p. 327–352 (cf. p. 112, 116).

Uchida, Hiroshi et Meiying Zhu (2001). « The Universal Networking Language beyond Machine Translation ». In : *Proceedings of the International Symposium on Language in Cyberspace*. Seoul, Korea. URL : http://www.undl.org/publications/UNL-beyond%20MT.html (cf. p. 25).

Vandendorpe, Christian (2008). « Le phénomène Wikipédia : une utopie en marche ». In : *Le Débat* 148, p. 17–30 (cf. p. 29).

Verspoor, K. et al. (2009). « From Form to Meaning : Processing Texts Automatically. Proceedings of the the biennal GSCL Conference ». In : sous la dir. de C. Chiarcos, Eckhart de Castilho et M. Stede. Tübingen, Germany : Narr Verlag. Chap. Abstracting the types away from a UIMA type system, p. 249–256 (cf. p. 13).

Vial, Loïc, Andon Tchechmedjiev et Didier Schwab (2016). « Extension lexicale de définitions grâce à des corpus annotés en sens ». In : *Actes de la conférence conjointe JEP-TALN-RECITAL 2016, volume 2 : TALN*. Sous la dir. d'ATALA (cf. p. 131).

Vossen, Piek. (1998). *EuroWordNet : a multilingual database with lexical semantic networks*. Dordrecht [The Netherlands] ; Boston : Kluwer Academic (cf. p. 28, 37).

Vossen, Piek et al. (2010). « KYOTO : an open platform for mining facts ». In : *Proceedings of the 6th Workshop on Ontologies and Lexical Resources*. Beijing, China : Coling 2010 Organizing Committee, p. 1–10. URL : http://www.aclweb.org/anthology/W10-3301 (cf. p. 32).

Vulić, Ivan et Marie-Francine Moens (2015a). « Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction ». In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*. Beijing, China : ACL, p. 719–725. URL : http://aclweb.org/anthology/P15-2118 (cf. p. 128).

Vulić, Ivan et Marie-Francine Moens (2015b). « Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction ». In : *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*. Beijing, China : ACL, p. 719–725. URL : http://aclweb.org/anthology/P15-2118 (cf. p. 55).

Wagner, C. (2008). « Breaking the Knowledge Acquisition Bottleneck Through Conversational Knowledge Management ». In : *Information Resources Management Journal* 19.1, p. 70–83.

Waite, Aurelien et William Byrne (2015). « The Geometry of Statistical Machine Translation ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 376–386.

Wang, Dong et Yiye Lin (2015). « Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation ». In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1006–1011.

Wei, Bifan et al. (2013). « DFT-extractor : A System to Extract Domain-specific Faceted Taxonomies from Wikipedia ». In : *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13 Companion. Rio de Janeiro, Brazil : International World Wide Web Conferences Steering Committee, p. 277–280.

ISBN : 9781450320382. URL : http://dl.acm.org/citation.cfm?id=2487788.2487922 (cf. p. 31).

WITT, Andreas et al. (2009). « Multilingual language resources and interoperability ». In : *Language Resources and Evaluation* 43.1, p. 1–14. ISSN : 1574-020X. DOI : 10.1007/s10579-009-9088-x (cf. p. 22, 60).

WU, Stephen et al. (2013). « MayoClinicNLP–CORE : Semantic representations for textual similarity ». In : *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1 : Proceedings of the Main Conference and the Shared Task*. Altanta, Georgia : ACL, p. 148–154 (cf. p. 112).

WU, Zhibiao et Martha PALMER (1994). « Verbs semantics and lexical selection ». In : *Proceedings of the 32nd annual meeting on ACL*. T. 2. ACL '94. Las Cruces : ACL, p. 133–138 (cf. p. 47, 50).

ZADEH, Pourya, Reshad HOSSEINI et Suvrit SRA (2016). « Geometric Mean Metric Learning ». In : *Proceedings of The 33rd International Conference on Machine Learning*, p. 2464–2471 (cf. p. 133).

ZESCH, Torsten, Christof MÜLLER et Iryna GUREVYCH (2008a). « Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary ». In : *Proceedings of the 6th International Conference on Language Resources and Evaluation*. electronic proceedings. Marrakech, Morocco (cf. p. 31).

ZESCH, Torsten, Christof MÜLLER et Iryna GUREVYCH (2008b). « Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary ». In : *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

ZESCH, Torsten, Christof MÜLLER et Iryna GUREVYCH (2008c). « Using Wiktionary for Computing Semantic Relatedness ». In : *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI*. Chicago, Illinois, USA.

ZESCH, Torsten, Christof MÜLLER et Iryna GUREVYCH (2008d). « Using Wiktionary for Computing Semantic Relatedness ». In : *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*. AAAI'08. Chicago, Illinois : AAAI Press, p. 861–866. ISBN : 9781577353683. URL : http://dl.acm.org/citation.cfm?id=1620163.1620206 (cf. p. 50).

ZIPF, George K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA).

ZOU, Y. Will et al. (2013). « Bilingual Word Embeddings for Phrase-Based Machine Translation ». In : *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA : ACL, p. 1393–1398. URL : http://aclweb.org/anthology/D13-1141 (cf. p. 55).

Abstract

When it comes to the construction of multilingual lexico-semantic resources, the first thing that comes to mind is that the resources we want to align should share the same data model and format (representational interoperability). However, with the emergence of standards such as LMF and their implementation and widespread use for the production of resources in the form of lexical linked data (Ontolex), representational interoperability has ceased to be a major challenge for the production of large-scale multilingual resources. However, as far as the interoperability of sense-level multilingual alignments is concerned, a major challenge is the choice of a suitable interlingual pivot. Many resources make the choice of using English senses as the pivot (e.g. BabelNet, EuroWordNet), although this choice leads to a loss of contrast between English senses that are lexicalized with different words in other languages. The use of acception-based interlingual representations, a solution proposed over 20 years ago, could be viable. However, the manual construction of such language-independent pivot representations is very difficult due to the lack of expert speaking enough languages fluently and algorithms for their automatic constructions have never materialized, mainly because of the lack of a formal axiomatic characterization that ensures the preservation of their correctness properties. In this thesis, we address this issue by first formalizing acception-based interlingual pivot architectures through a set of axiomatic constraints and rules that guarantee their correctness. Then, we propose algorithms for the initial construction and the update of interlingual acception-based multilingual resources by exploiting the combinatorial properties of pairwise bilingual translation graphs. Secondly, we study the practical considerations of applying our construction algorithms on a tangible resource, DBNary (a lexical linked data resource extracted from Wiktionary).

Résumé

Lorsqu'il s'agit de la construction de ressources lexico-sémantiques multilingues, la première chose qui vient à l'esprit, est la nécessité que les ressources à aligner partagent le même format de données et la même représentation (interopérabilité représentationnelle). Avec l'apparition de standards tels que LMF et leur adaptation au web sémantique pour la production de ressources lexico-sémantiques multilingues en tant que données lexicales liées ouvertes (Ontolex), l'interopérabilité représentationnelle n'est plus un verrou majeur. Cependant, en ce qui concerne l'interopérabilité des alignements multilingues, le choix et la construction d'un pivot interlingue est l'un des obstacles principaux. Pour nombre de ressources (par exemple BabelNet, EuroWordNet), le choix est fait d'utiliser l'anglais, ou une autre langue, comme pivot interlingue. Ce choix mène à une perte de contraste dans les cas où des sens du pivot ont des lexicalisations différentes dans la même acception dans plusieurs autres langues. L'utilisation d'un pivot à acceptions interlingues, solution proposée il y a déjà plus de 20 ans, pourrait être viable. Néanmoins, leur construction manuelle est trop ardue du fait du manque d'experts parlant assez de langues et leur construction automatique pose problème du fait de l'absence d'une formalisation et d'une caractérisation axiomatique permettant de garantir leurs propriétés. Nous proposons dans cette thèse de d'abord formaliser l'architecture à pivot interlingue par acceptions, en développant une axiomatisation garantissant leurs propriétés. Nous proposons ensuite des algorithmes de construction initiale automatique en utilisant les propriétés combinatoires du graphe des alignements bilingues ainsi que des algorithmes de mise à jour. Dans un deuxième temps, nous étudions les implications de l'application de ces algorithmes sur DBNary (une ressource en données lexicales liées ouvertes extraite à partir de Wiktionary).