Construction d'ontologies à partir de textes. L'apport de l'analyse de concepts formels.

Thibault Mondary

UNIVERSITÉ PARIS 13 — PARIS-NORD
LIPN — UMR CNRS 7030

T H È S E

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PARIS 13

Discipline : Informatique

présentée et soutenue publiquement par

Thibault Mondary

le

vendredi 27 mai 2011

Construction d'ontologies à partir de textes

L'apport de l'analyse de concepts formels

Composition du jury

Rapporteurs :	Gilles KASSEL	Professeur, Université de Picardie, MIS
	Pierre ZWEIGENBAUM	Directeur de recherche, CNRS, LIMSI
Examinateurs :	Nathalie AUSSENAC-GILLES	Directeur de recherche, CNRS, IRIT
	Nathalie PERNELLE	Maître de conférences, Université Paris 11, LRI
	Yannick TOUSSAINT	Chargé de recherche, INRIA, LORIA
Directrices :	Adeline NAZARENKO	Professeur, Université Paris 13, LIPN
	Sylvie DESPRÉS	Professeur, Université Paris 13, LIM&BIO

Remerciements

Je tiens tout d'abord à remercier Sylvie Després et Adeline Nazarenko de leur aide, leur écoute et leurs précieux conseils tout au long de ces quatre années. Leur expérience et leur disponibilité ont permis à ma thèse de se dérouler dans les meilleures conditions possibles.

Je remercie aussi tous les membres du LIPN et plus particulièrement de l'équipe RCLN de la qualité de leur accueil. Merci à tous ceux qui m'ont accompagné dans mon apprentissage de la recherche et de l'enseignement.

Je tiens également à remercier Gilles Kassel et Pierre Zweigenbaum d'avoir accepté de rapporter cette thèse, ainsi que Nathalie Aussenac-Gilles, Nathalie Pernelle et Yannick Toussaint pour leur participation au jury. Merci à eux pour toutes leurs précieuses remarques.

Il ne faut pas oublier de mentionner le projet Quaero, pour avoir financé la fin de ma thèse tout en me permettant de découvrir un autre aspect du monde de la recherche.

Comment ne pas citer mes anciens collègues de bureau et amis, Christophe, Aurélien et Cédric avec lesquels nous avons partagé des moments très enrichissants.

Je remercie enfin ma famille proche de m'avoir apporté soutien et réconfort dans les moments difficiles.

Sommaire

Chapitre 1

Introduction

Les ontologies, structures de connaissances dont l'idée remonte à Aristote, représentent un moyen d'expression, de partage, de mutualisation et de réutilisation des connaissances. Elles sont idéalement utilisables à la fois par les machines et par les humains. Une définition communément acceptée d'une ontologie est celle proposée par (Studer *et al.*, 1998) : *Une ontologie est une spécification formelle d'une conceptualisation d'un domaine, partagée par un groupe de personnes, qui est établie selon un certain point de vue imposé par l'application construite.* Pratiquement, une ontologie est constituée d'un ensemble de concepts organisés à l'aide de relations hiérarchiques et spécialisées. Des axiomes, des règles et des instances peuvent parfois venir la compléter.

L'utilité des ontologies est maintenant reconnue, notamment au sein de la mouvance du web sémantique. Toutefois, leur élaboration reste une tâche fastidieuse et complexe qui requiert à la fois une expertise du domaine que l'on souhaite modéliser et des connaissances en modélisation. L'ingénierie des connaissances (IC) est la discipline qui étudie cette problématique. Elle propose des méthodologies d'élaboration des ontologies qui les englobent depuis la spécification des besoins jusqu'à la maintenance, c'est le *cycle de vie* d'une ontologie.

Le cycle de vie d'une ontologie détermine dans quel ordre les étapes de développement d'une ontologie doivent être effectuées. La Methontology (Fernandez-López et Juristo, 1997) est une formalisation de ce processus, mais il en existe d'autres, notamment pour (Cimiano *et al.*, 2006). Ces méthodologies proposent une vue d'ensemble, mais ne sont pas très détaillées sur le processus de conceptualisation. C'est à ce processus que nous nous intéressons ici. Cette phase de conceptualisation représente la pierre angulaire du développement des ontologies : privées de celle-ci toutes les autres étapes deviennent caduques. Nous appelons conceptualisation l'identification des concepts clés d'un domaine et l'explicitation de leurs caractéristiques.

Cette phase de conceptualisation nécessite une interaction soutenue avec les experts du domaine, qui ne sont pas toujours disponibles. Une piste pour diminuer le travail de l'expert consiste à utiliser conjointement une autre source de connaissances, que sont les textes. Le recours aux textes est légitimé par l'idée qu'ils sont porteurs de connaissances stabilisées et partagées par des communautés de pratiques. Même s'ils ne les remplacent pas totalement, les textes sont plus facilement disponibles que les experts qui manquent de temps pour participer au processus de modélisation. Un autre avantage d'utiliser les textes pour la construction d'ontologies est qu'ils permettent d'y inclure des informations linguistiques. Ces informations vont faciliter l'utilisation de l'ontologie dans des systèmes qui travaillent sur les textes, par exemple dans les tâches

d'extraction d'informations. En revanche, le fait que les connaissances soient inscrites dans les textes peut donner une vision déformée du domaine à modéliser.

Tout l'enjeu de la construction d'ontologies à partir de textes consiste donc à exploiter au mieux les éléments fournis par ceux-ci, tout en s'affranchissant de leur empreinte linguistique et discursive pour élaborer un modèle conceptuel. C'est le défi que cette thèse cherche à relever.

Ce travail s'inscrit dans la tradition héritée du groupe TIA (Terminologie et Intelligence Artificielle) qui s'est intéressé au rôle de la terminologie dans le processus de modélisation des connaissances. L'utilisation des textes pour la construction d'ontologies est étudiée depuis une quinzaine d'années au sein du LIPN, qui a créé la notion d'« ingénierie des connaissances textuelles (ICT [1]) ». Ce domaine de recherches vise à combiner des méthodes de traitement automatique des langues (TAL) avec des stratégies de modélisation issues de l'IC. La méthode Terminae a été conçue dans ce contexte pour formaliser le passage des textes à une ontologie *via* une approche terminologique (Biebow et Szulman, 1999; Aussenac-Gilles *et al.*, 2008). Un nouvel essor a été donné à ce courant de recherches par le projet Dafoe (Szulman *et al.*, 2009) de développement de plateforme de construction d'ontologies de grande taille.

L'une des difficultés de cette approche terminologique, que ce soit dans l'outil Terminae ou dans Dafoe, est la masse de données terminologiques à analyser. Ce travail reste essentiellement manuel. Nous proposons dans cette thèse d'étudier comment des méthodes de regroupement, issues de l'apprentissage, permettraient d'outiller Terminae. Nous nous focalisons sur une approche symbolique, l'Analyse de Concepts Formels (ACF). Cette approche propose une formalisation de la notion de concept dans une structure de treillis. Des travaux antérieurs (Cimiano *et al.*, 2005) avaient montré, en effet, que cette approche propose des regroupements plus pertinents dans une perspective ontologique que des méthodes concurrentes. Nous avons développé une méthode qui exploite à la fois les connaissances terminologiques traditionnelles de Terminae et l'approche ACF initialement proposée sur les mots. Les expériences que nous avons menées mettent cependant en évidence un certain nombre de difficultés. La méthode paraît extrêmement sensible à la nature du corpus d'acquisition utilisé. La taille du treillis obtenu en sortie rend difficile son exploitation. Enfin, la transformation du treillis en ontologie soulève des problèmes d'interprétation non négligeables. Nous avons proposé plusieurs manières de prendre en compte les données textuelles dans l'ACF, adaptées à différents types de corpus, et des guides d'interprétation du treillis en ontologie.

La thèse est structurée, de façon assez classique, en deux parties. La première présente un état de l'art de la construction d'ontologies à partir de textes (COT), structuré en trois chapitres. Dans le premier chapitre, nous définissons la notion d'ontologie au travers d'un survol historique et nous étudions les éléments remarquables des textes

1. Depuis septembre 2000, il existe une option ICT dans le DEA puis dans le master d'informatique de Paris 13.

qui peuvent aider à leur construction. Dans le deuxième chapitre nous étudions les techniques de l'état de l'art qui permettent de passer des textes à une ontologie : nous présentons plusieurs approches de la conceptualisation. Le troisième chapitre de l'état de l'art est consacré à la présentation de l'ACF et à son utilisation avec les textes.

La seconde partie de cette thèse présente nos propositions pour combiner l'ACF avec une approche terminologique de conceptualisation. Elle est structurée en quatre chapitres. Dans le chapitre 5, nous présentons les expériences menées à l'aide d'une première version de notre système, inspirée de l'état de l'art. Nous montrons les problèmes inhérents à l'utilisation de l'ACF sur des données textuelles. Nous apportons des solutions, parfois partielles, à chacun de ces problèmes dans le chapitre 6. Le chapitre suivant présente les expérimentations que nous avons menées avec la version améliorée de notre système et tente d'apprécier son apport. Le dernier chapitre de cette partie est consacré à la question plus générale de l'évaluation des outils de COT.

En annexe, nous présentons la page du manuel de notre système de construction de contexte formel.

Première partie

La Construction d'Ontologies à partir de Textes (COT)

Tout au long des trois chapitres qui suivent, nous présentons un état de l'art concernant la construction d'ontologies à partir de textes (COT). Le premier chapitre, structuré en deux sections, est dédié à la définition de ce que l'on nomme « ontologie » et à la présentation des éléments utiles pour leur construction que l'on peut extraire des textes. Le chapitre suivant est consacré à l'étude des méthodes utilisables pour obtenir ces éléments, tandis que le dernier chapitre de notre état de l'art se focalise sur une méthode de regroupement symbolique, l'analyse de concepts formels (ACF).

Chapitre 2

Des textes et des ontologies

Sommaire

Ce chapitre a pour objectif de présenter les deux éléments sur lesquels la suite de notre travail va reposer, à savoir les ontologies et les textes. La première partie de ce chapitre présente la notion d'ontologie au travers d'un survol historique, tandis que la seconde se concentre sur les éléments qui peuvent aider à la COT et que l'on est susceptible de retrouver dans les textes.

2.1 Les ontologies

Ontologie est un terme de philosophie qui signifie « doctrine ou théorie de l'être en tant qu'être [1] ». Le mot ontologie, introduit aux alentours du XVI^{e} siècle, est plus récent que la discipline qu'il désigne. Cette discipline fut étudiée par les Grecs dans l'Antiquité, notamment par Parménide, Platon puis Aristote. L'origine de cette discipline semble provenir du désir humain de caractériser « ce qui est » afin de mieux le maîtriser et de tenter de comprendre ce que signifie « être ». Pour ce faire, Aristote propose de décrire « ce qui est » par des caractéristiques essentielles (la *Substance*), qui perdurent lors des transformations de l'être, et des caractéristiques accidentelles (ou transitoires) qui ne sont pas stables dans le temps. Il propose de ramener l'ensemble des manifestations de l'être dans le monde à dix catégories (la substance, la quantité, la qualité, la relation,

1. D'après l'Encyclopædia Universalis.

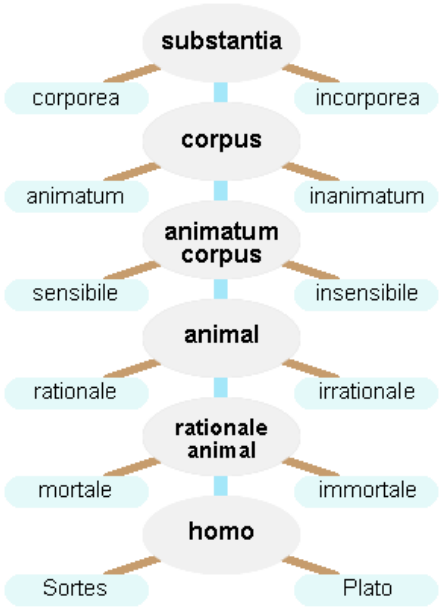

FIGURE 2.1 : Arbre de Porphyre selon Petrus Hispanus

le lieu, le temps, la situation, l'état, l'action, la passion). Porphyre (234-305), dans son *Isagogè*, structure les dix catégories d'Aristote au sein d'un arbre dont le tronc représente des catégories de plus en plus générales et les feuilles associées indiquent les oppositions au sein de la catégorie. Cet arbre, présenté sur la figure 2.1, est nommé Arbre de Porphyre. Porphyre est pour ainsi dire celui qui a, dans la lignée d'Aristote, formalisé la relation de généralisation.

La *taxinomie*[2] est la science de la classification, notamment du vivant. La dénomination fut proposée par Candolle au XIX^{e} siècle. Chaque organisme est classé dans un taxon, une entité conceptuelle qui regroupe les organismes vivants possédant des caractéristiques communes. Le taxon le plus générique est le monde vivant, le plus spécifique est l'espèce.

Bien plus tard, l'informatique a repris ce terme pour désigner une hiérarchie de concepts organisés par des relations de subsomption. L'histoire de l'ontologie en informatique prend racine dans les recherches menées en logique, en intelligence artificielle et en sciences cognitives. Le terme ontologie a été employé pour la première fois dans le domaine de l'informatique en 1980 par John McCarthy. Ce dernier affirmait que « les personnes qui concevaient des systèmes en intelligence artificielle devraient au préalable lister les éléments qui existent dans le monde, produisant de fait une *ontologie* de notre monde » (Smith et Welty, 2001).

2. ou parfois *taxonomie*.

Les systèmes experts, qui comptent parmi les premiers artefacts de l'IA, avaient pour objectif de reproduire les mécanismes cognitifs d'un expert pour une tâche donnée. Un système expert est composé d'une base de faits, d'une base de règles et d'un moteur d'inférence. Il est capable de produire de nouveaux faits par déduction (si P est un fait et si l'on sait que $P \Rightarrow Q$, alors Q est un nouveau fait). Le premier système expert, Dendral (Feigenbaum *et al.*, 1970), voit le jour en 1965. MYCIN (Shortliffe *et al.*, 1975) est un système expert médical qui avait pour but de déterminer la meilleure thérapie en cas d'infection bactérienne, à l'aide d'une base de 600 règles. NEOMYCIN (Clancey et Letsinger, 1981) est une amélioration de MYCIN qui introduit notamment un niveau supplémentaire d'abstraction *via* des méta-règles, c'est-à-dire des stratégies de diagnostic.

Dès lors que les architectes des systèmes d'IA et les concepteurs de logiciels ont commencé à se focaliser sur les connaissances traitées par leur système plutôt que sur les systèmes en eux-mêmes est apparue la problématique de leur modélisation et de leur intégration dans les systèmes. Cette problématique a donné naissance à une nouvelle discipline, l'ingénierie des connaissances (Feigenbaum et MacCorduck, 1983) et à des méthodologies d'acquisition de connaissances, notamment KADS (Akkermans *et al.*, 1993).

Plus spécifiquement, le problème d'expression des connaissances (Kayser, 1997) donnera quant à lui presque simultanément naissance à deux approches, pas forcément antagonistes : les graphes conceptuels et les logiques de description. Ces dernières sont depuis le milieu des années 80 (Brachman et Schmolze, 1985) une réponse à la problématique de l'explicitation formelle (dans le sens qu'un système peut traiter) des connaissances qui soit un compromis entre *expressivité* et *décidabilité*. Nous parlerons plus en détail des logiques de description dans la section 2.1.3.

Les graphes conceptuels, introduits par (Sowa, 1984), sont un formalisme graphique de représentation des connaissances. Ils ont pour ambition de fournir un ensemble de primitives graphiques pour partager des connaissances, qui soient également transposables sous une forme manipulable par l'ordinateur. Un graphe conceptuel est constitué de deux types de noeuds, les concepts qui sont schématisés par des rectangles et les relations conceptuelles qui sont représentées par des ellipses. (Sowa, 1984) définit un opérateur permettant de transformer un graphe conceptuel en formules de logique du premier ordre. Il est possible de raisonner avec les graphes conceptuels en utilisant des algorithmes de la théorie des graphes ou en les transformant en formules logiques du premier ordre. Si le graphe devient trop complexe, la transformation en logique du premier ordre n'est plus possible.

Une fois les premiers modèles de connaissances construits la perspective de devoir réinventer la roue à chaque nouvelle modélisation fit émerger l'idée de partage et de réutilisabilité. Les ontologies apparaissent comme une solution à ces problèmes. Mais finalement qu'est-ce qu'une ontologie au sens informatique du terme ? De quoi est-elle constituée ? C'est ce que nous allons définir dans la sous-section suivante.

2.1.1 Qu'est-ce qu'une ontologie ?

Nous avons vu qu'une ontologie informatique est une tentative de réponse aux problèmes de partage et de réutilisation des connaissances par l'homme et par la machine. Plusieurs auteurs ont tenté de donner une définition de l'ontologie. La plus connue est celle de (Gruber, 1993) : « une ontologie est une spécification explicite d'une conceptualisation ». Cette définition est cependant trop vague et trop générique (Smith et Welty, 2001). Elle a été étendue par (Borst, 1997) : « une ontologie est une spécification formelle d'une conceptualisation partagée » puis par (Studer *et al.*, 1998) : « Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée ». Une discussion complète de la signification des termes de cette définition est proposée en introduction de (Staab et Studer, 2009). Nous en retiendrons qu'une conceptualisation « est une vue abstraite et simplifiée du monde que l'on souhaite représenter dans un but donné », qu'une spécification formelle explicite est la transposition de la conceptualisation dans un langage qui soit utilisable par une machine et enfin que la notion de partage renvoie à une notion de compromis dans la modélisation, un socle de commun accord qui permet de satisfaire une communauté de personnes.

Une acceptation commune des composantes d'une ontologie distingue les concepts, qui peuvent être primitifs (les concepts dont on postule l'existence) ou définis (des concepts qui sont fabriqués à partir d'autres concepts), organisés hiérarchiquement par une relation de subsomption ou plus rarement de partie-tout, des rôles qui sont des relations entre les concepts pouvant être organisés hiérarchiquement, des axiomes sur les concepts ou sur les rôles servant à expliciter la description et parfois des règles logiques. L'ontologie peut également contenir des instances (de concepts ou de rôles), dans ce cas, on parle parfois d'« ontologie peuplée » ou de « base de connaissances ».

Qu'est-ce qu'un concept ? D'après le petit Larousse, un concept est une « idée générale et abstraite que se fait l'esprit humain d'un objet de pensée concret ou abstrait, et qui lui permet de rattacher à ce même objet les diverses perceptions qu'il en a, et d'en organiser les connaissances ».

Cette notion peut s'illustrer à l'aide d'un triangle sémiotique, présenté sur la figure 2.2, dont la première version est due à (Ogden et Richards, 1923). Ce triangle met en avant la différence qui existe entre les objets du monde (abstraits ou concrets), leur désignation dans un langage (textuel, iconographique, etc.) et leur abstraction conceptuelle.

Guarino dans (Staab et Studer, 2009) définit une ontologie comme étant avant tout une conceptualisation (D, W, R) avec D l'univers du discours, W un ensemble de mondes possibles et R un ensemble de relations conceptuelles sur $< D, W >$. Par relation conceptuelle, il entend une fonction d'arité pouvant éventuellement être multiple qui relie W à D. Pour spécifier explicitement une conceptualisation il faut soit décrire de manière systématique toutes les extensions des relations conceptuelles dans tous les mondes possibles, soit les contraindre intensionnellement à l'aide d'axiomes comme

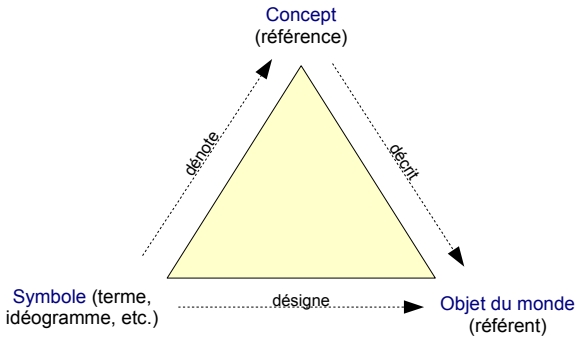

FIGURE 2.2: Triangle sémiotique

la transitivité, la réflexivité, etc. Une ontologie est donc, pour Guarino, « une théorie logique conçue pour capturer les modèles qui correspondent à une conceptualisation en écartant les autres ».

Formalisation

Nous trouvons les définitions de Guarino trop philosophiques et trop éloignées de la problématique de la COT. Nous préférons reprendre les définitions formelles de (Cimiano, 2006), à notre avis plus opérationnelles pour notre problème.

Définition 1 (Ontologie) *Une ontologie est une structure :*

$$\mathcal{O} := (C, \leq_C, R, \sigma_R, \leq_R, \mathcal{A}, \sigma_\mathcal{A}, \mathcal{T}) \tag{2.1}$$

avec :

- $C, R, \mathcal{A}, \mathcal{T}$ *quatre ensembles disjoints dont les éléments sont appelés respectivement* identifiants de concepts, identifiants de relations, identifiants d'attributs *et* types de données *;*
- $\leq_C$ *un treillis partiel sur* C *muni d'une borne supérieure* $root_C$*, appelé* hiérarchie de concepts *;*
- σ_R *une fonction de* R *dans* C^+ *appelée* signature de relation *;*
- $\leq_R$ *un ordre partiel sur* R *appelé* hiérarchie de relation*, où* $r1 \leq_R r2$ *implique* $|\sigma_R(r1)| = |\sigma_R(r2)|$ *et* $\pi_i(\sigma_R(r1)) \leq_C \pi_i(\sigma_R(r2))$ [3] *pour* $1 \leq i \leq |\sigma_R(r1)|$ *;*
- $\sigma_\mathcal{A}$ *une fonction de* $\mathcal{A}$ *dans* $C \times \mathcal{T}$ *appelée* signature d'attribut *;*
- $\mathcal{T}$ *un ensemble de types de données (entiers, chaînes de caractères, etc.).*

Nous omettons sciemment dans la suite la définition formelle d'un système axiomatique pour l'ontologie, car son utilité dépasse le cadre de cette thèse.

3. $\pi_i(n)$ représente le *ième* élément du n-uplet n

Définition 2 (Domaine et co-domaine) *Pour une relation $r \in R$ telle que $\sigma_R(r) = 2$ nous appelons $\pi_1(\sigma_R(r))$ le* domaine *de r et $\pi_2(\sigma_R(r))$ son* co-domaine.

Une ontologie est une structure conceptuelle indépendante de la langue. Lorsque l'on construit une ontologie à partir de textes, ou que l'on utilise une ontologie pour annoter un texte, il est nécessaire de faire le lien entre les niveaux linguistiques et conceptuels. Différents formalismes ont été proposés et sont encore à l'étude. Nous nous contentons ici de supposer que l'ontologie peut être « lexicalisée » au sens où l'entend (Cimiano, 2006) :

Définition 3 (Ontologie lexicalisée) *Une ontologie lexicalisée est une paire :*

$$\mathcal{OL}ex := (\mathcal{O}, Lex) \tag{2.2}$$

avec Lex un lexique.

Définition 4 (Lexique) *Un lexique pour une ontologie $\mathcal{O}$ est une structure :*

$$Lex := (S_C, S_R, S_\mathcal{A}, Ref_C, Ref_R, Ref_\mathcal{A}) \tag{2.3}$$

avec :

- *$S_C, S_R, S_\mathcal{A}$ trois ensembles dont les éléments sont appelés respectivement* dénotations [4] de concepts, de relations et d'attributs *;*
- *Ref_C une relation telle que $Ref_C \subseteq C \times S_C$ appelée* référence lexicale de concept *;*
- *Ref_R une relation telle que $Ref_R \subseteq R \times S_R$ appelée* référence lexicale de relation *;*
- *$Ref_\mathcal{A}$ une relation telle que $Ref_\mathcal{A} \subseteq \mathcal{A} \times S_\mathcal{A}$ appelée* référence lexicale d'attribut.

Il convient de noter que rien ne vient contraindre la nature des éléments des ensembles $S_C, S_R, S_\mathcal{A}$, les *dénotations*. Celles-ci peuvent notamment être des termes (nous le considérerons dans la suite).

Définition 5 (Concepts dénotés) *Pour une dénotation $s \in S_C$, les concepts c dénotés sont définis par :*

$$Ref_C(s) := \{c \in C | (s, c) \in Ref_C\} \tag{2.4}$$

$Ref_R(s)$ et $Ref_\mathcal{A}(s)$ sont définis de manière similaire.

Si $|Ref_C(s)| > 1$ cela signifie que le terme s est polysémique.

4. Dans le livre de Cimiano le terme *sign* est utilisé, d'où le S que nous choisissons de conserver tel quel.

Définition 6 (Dénotations associées) *Pour un concept $c \in C$, les dénotations associées sont définies par :*

$$Ref_C^{-1}(c) := \{s \in S_C | (s, c) \in Ref_C\} \tag{2.5}$$

$Ref_R^{-1}(r)$ et $Ref_{\mathcal{A}}^{-1}(a)$ sont définis de manière similaire.

Si $|Ref_C^{-1}(c)| > 1$ cela signifie que les termes associés à c sont synonymes.

Définition 7 (Base de connaissances) *Une base de connaissances pour une ontologie $\mathcal{O}$ est une structure :*

$$KB := (I, \iota_C, \iota_R, \iota_{\mathcal{A}}) \tag{2.6}$$

avec :

- *I un ensemble dont les éléments sont appelés* instances *;*
- *ι_C une fonction de C dans 2^I appelée* instanciation de concept *;*
- *ι_R une fonction de R dans 2^{I+}, appelée* instanciation de relation *et vérifiant $\iota_R(r) \subseteq \prod_{1 \leq i \leq |\sigma(r)|} \iota_C(\pi_i(\sigma(r)))$;*
- *$\iota_{\mathcal{A}}$ une fonction de $\mathcal{A}$ dans $I \times \bigcup_{t \in \tau} [[t]]$, appelée* instanciation d'attribut *et vérifiant $\iota_{\mathcal{A}}(a) \subseteq \iota_C(\pi_1(\sigma(a))) \times [[\pi_2(\sigma(a))]]$, avec $[[t]]$ les valeurs du type de données $t \in \mathcal{T}$.*

Définition 8 (Lexique d'instances) *Un lexique d'instances pour une base de connaissances KB est une paire :*

$$IL := (S_I, R_I) \tag{2.7}$$

avec :

- *S_I un ensemble dont les éléments sont appelés* dénotations d'instances *;*
- *R_I une relation telle que $R_I \subseteq S_I \times I$ appelée* référence lexicale d'instance.

Définition 9 (Base de connaissances lexicalisée) *Une base de connaissance lexicalisée est une paire :*

$$LKB := (KB, IL) \tag{2.8}$$

Maintenant nous sommes en mesure de définir l'extension et l'intension d'un concept dans une ontologie munie d'une base de connaissances. Intuitivement l'extension d'un concept est constituée de l'ensemble de ses instances, tandis que son intension est représentée par toutes les caractéristiques (les éléments de $\mathcal{O}$) qui le définissent.

Définition 10 (Extension) *L'extension* $[[c]]_{KB} \subseteq I$ *d'un concept* $c \in C$ *est définie récursivement à l'aide des règles suivantes :*

- $[[c]]_{KB} \leftarrow \iota_C(c)$;
- $[[c]]_{KB} \leftarrow [[c]]_{KB} \cup [[c']]_{KB}$, *pour* $c' <_C c$.

L'extension $[[r]]_{KB} \subseteq I^+$ *d'une relation* $r \in R$ *est définie de manière symétrique.*

L'extension $[[a]]_{KB} \subseteq I \times [[\mathcal{T}]]$ *d'un attribut* $a \in \mathcal{A}$ *est définie plus simplement par* $[[a]]_{KB} \leftarrow \iota_{\mathcal{A}}(a)$ *car il n'y a pas de relation de subsomption entre les attributs.*

Définition 11 (Intension) *L'intension d'une ontologie* $\mathcal{O}$ *est une structure :*

$$\mathcal{I} := (\mathcal{L}_I, i_C, i_R, i_{\mathcal{A}}) \tag{2.9}$$

avec :

- $\mathcal{L}_I$ *un langage représentant les caractéristiques (intensionnelles) des concepts, des attributs et des relations ;*
- $i_C : C \to \mathcal{L}_I$, $i_R : R \to \mathcal{L}_I$ *et* $i_{\mathcal{A}} : \mathcal{A} \to \mathcal{L}_I$ *trois correspondances qui associent les concepts, relations et attributs à leurs caractéristiques intensionnelles respectives.*

2.1.2 Des familles d'ontologies

Une fois que les premières ontologies furent construites, des chercheurs ont commencé à vouloir définir des catégories qui permettraient de les caractériser. On a vu ainsi émerger différents axes de classification des ontologies.

(Van Heijst *et al.*, 1997) en propose deux, le premier concerne le type de conceptualisation (ontologie terminologique, ontologie de bases de données, ontologie de modélisation des connaissances), même si ce premier axe nous paraît peu clair (voir aussi (Guarino, 1997), notamment la différence entre une ontologie de modélisation des connaissances et une ontologie terminologique). Le second axe mis en avant par les auteurs est le sujet abordé par la conceptualisation. Ils en distinguent quatre : les ontologies d'application, les ontologies de domaine, les ontologies génériques et les ontologies de représentation. Nous retiendrons deux grandes familles d'ontologies : les ontologies de haut niveau et les ontologies de domaine.

Les ontologies de haut niveau, ou ontologies « fondationnelles », visent à représenter les concepts de sens commun. OpenCyc [5] est une ontologie de haut niveau à vocation encyclopédique issue du projet Cyc [6]. Cyc contient des connaissances du sens commun ainsi que certaines connaissances spécialisées. Elles sont organisées en « microthéories »,

5. http://www.opencyc.org/
6. http://www.cyc.com

des parties de l'ontologie qui sont non-contradictoires. Cyc contient plusieurs centaines de milliers de concepts et plusieurs millions d'assertions (majoritairement hiérarchiques), ainsi que des instances de concepts.

Les ontologies de domaine cherchent, comme leur nom l'indique, à modéliser les connaissances d'un domaine particulier, comme le droit international du travail par exemple. Les ontologies de domaine sont plus spécifiques que les ontologies noyaux de domaine (core-ontologies) qui englobent les concepts généraux d'une discipline, par exemple le droit.

Les ontologies d'application sont spécifiques à une tâche donnée. Elles combinent une ou des ontologies de haut niveau et une ou des ontologies plus spécifiques, ainsi que les connaissances requises par l'application visée.

Les ontologies peuvent également être catégorisées selon le degré de formalisation explicite qu'elles incorporent, c'est-à-dire par leur capacité à contraindre leur interprétation. (Uschold et Gruninger, 2004) propose la dénomination d'ontologie *légère* pour les ontologies qui ne contiennent pas (ou peu) d'axiomes, par opposition aux ontologies dites *lourdes*[7]. Une taxonomie peut être considérée comme une ontologie légère.

Un autre axe de différenciation des ontologies est la granularité de la description conceptuelle et du niveau de détails atteint par les concepts. Cet axe se justifie par le but et le public visés par l'ontologie : à titre d'exemple, une ontologie des deux-roues destinée aux utilisateurs contiendra moins de détails que la même ontologie destinée aux constructeurs.

Dans cette thèse, nous nous plaçons dans un cadre de construction d'ontologies d'applications à partir de corpus textuels. Plus particulièrement, nous souhaitons nous focaliser sur la construction des ontologies de domaine qui composent ces dernières.

2.1.3 Représentation des ontologies

Studer met en avant le côté formel d'une ontologie. Mais où se situe la limite entre une ontologie formelle et une ontologie informelle ? (Uschold et Gruninger, 2004) voit le niveau d'expressivité comme un continuum qui s'étend d'une terminologie à la logique, avec une cassure censée représenter la frontière entre informel et formel (figure 2.3). Plus on se rapproche de l'extrémité droite du graphique, plus le niveau d'ambiguïté diminue et plus le pouvoir expressif augmente. Et plus le pouvoir expressif augmente plus il devient possible de faire des inférences, mais cela jusqu'à un certain point. En effet, si le langage utilisé devient trop expressif il en devient indécidable dans un temps fini.

7. *heavyweight*

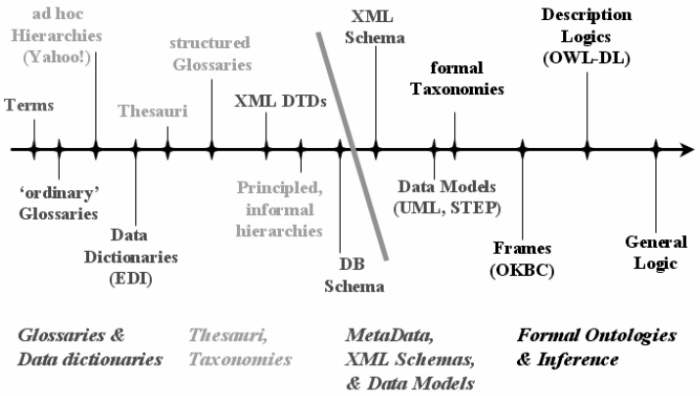

FIGURE 2.3 : Echelle du formalisme pour Uschold

Les logiques de description sont une famille de langages formels de représentation des connaissances. Le but des logiques de description est d'apporter un cadre logique formel (dans le sens exploitable par la machine) à la représentation des connaissances, en réalisant le compromis entre expressivité et décidabilité. L'expressivité représente la richesse des constructions logiques que l'on peut utiliser dans des formules ; plus celle-ci est importante plus il devient possible de contraindre, de préciser le modèle logique afin que ses interprétations « collent » à la réalité. La décidabilité d'une logique indique s'il est possible de trouver un algorithme qui calcule si une formule est un théorème *dans un temps fini*[8].

Les logiques de description couplées à un système de calcul d'inférences permettent de résoudre des problèmes de classification (« à quel(s) concept(s) une instance se rattache-t-elle ? », « quel est le père de ce concept défini ? »), mais aussi de vérifier la consistance (« est-ce que tout concept peut avoir des instances ? »).

La logique propositionnelle est décidable mais insuffisamment expressive pour les besoins de représentation des connaissances tandis que la logique du premier ordre (et *a fortiori* celles d'ordre supérieur) permet d'exprimer plus de connaissances, mais au détriment de la décidabilité. Les logiques de description s'inscrivent entre la logique propositionnelle et la logique des prédicats.

Elles font la distinction entre la `A-BOX`, boîte assertionnelle qui contient les individus du monde représenté et la `T-BOX`, boîte terminologique qui contient quant à elle la description des concepts et des rôles.

Il existe plusieurs logiques de description qui varient en fonction de leur expressivité ; par expressivité nous entendons ici les constructions autorisées dans la T-BOX, par exemple l'union de concepts ou les restrictions de cardinalité sur les rôles. Certaines ont été implémentées au sein de systèmes (Brachman et Schmolze, 1985; MacGregor, 1991; Patel-Schneider *et al.*, 1991) complets, d'autres plus récentes font l'objet d'une

8. Le temps de calcul peut toutefois devenir exponentiel.

standardisation par le W3C, comme OWL, que nous présentons brièvement dans la sous-section suivante.

OWL

OWL [9] est un langage de représentation des connaissances normalisé par le W3C. La première mouture d'OWL était découpée en trois sous-ensembles, OWL-Lite, OWL-DL et OWL-Full, classés par ordre d'expressivité croissante.

OWL-Lite est conçu pour représenter des hiérarchies avec des contraintes limitées ; par exemple OWL-Lite ne permet pas d'exprimer des contraintes de cardinalité autres que 0 ou 1, mais permet en revanche d'exprimer la transitivité ou les propriétés inverses. OWL-Lite permet de définir des concepts intersection, mais pas des concepts union. En contrepartie OWL-Lite est toujours décidable et il est aisé d'implémenter un moteur d'inférence.

OWL-DL (appelé ainsi en référence aux logiques de description) est un surensemble d'OWL-Lite qui offre un maximum d'expressivité tout en maintenant la complétude et la décidabilité des algorithmes d'inférence. OWL-DL offre notamment la possibilité d'exprimer des concepts union, des concepts énumérés, des concepts disjoints et la négation de concepts.

OWL-Full est un surensemble de OWL-DL qui offre la possibilité de recouvrement des types : un concept peut aussi être un individu ou une propriété et réciproquement. La contrepartie de cette expressivité est la perte de la décidabilité : rien ne garantit qu'un moteur d'inférence fournira une réponse en un temps fini.

OWL2

OWL2 [10] est la nouvelle mouture d'OWL. Elle offre de nouvelles constructions permettant une plus grande expressivité des restrictions (par exemple la disjonction des propriétés) et facilitant l'écriture des motifs fréquemment rencontrés en OWL-DL (par exemple un concept union de concepts disjoints).

OWL et OWL2 étant devenus des standards c'est tout naturellement que nous choisissons de les utiliser dans cette thèse. Cependant nous verrons que le choix d'un langage de représentation des ontologies n'est pas un point crucial pour nous, car nous avons choisi de nous concentrer sur les étapes qui interviennent en amont, c'est-à-dire la phase de conceptualisation.

Définition 12 (Conceptualisation) *Nous appelons* conceptualisation *l'identification des concepts clés d'un domaine, et l'explicitation de leurs caractéristiques.*

9. http://www.w3.org/TR/owl-features
10. http://www.w3.org/TR/owl2-overview

2.2 Les textes

Les documents textuels sont depuis l'invention de l'écriture un moyen de communication. Quand ils ont une visée informative, ils expriment des connaissances dans une langue naturelle, mais relèvent généralement d'un objectif déterminé. Plusieurs documents textuels regroupés dans une optique précise constituent un corpus textuel, ou corpus. De manière similaire au paragraphe 2.1.2 et comme dans (Habert *et al.*, 1997), nous opposons les domaines sur lesquels portent les corpus. Les corpus généraux regroupent des textes qui ne sont pas spécifiques à un domaine de connaissance particulier, tandis que les corpus de spécialité sont focalisés sur un champ de connaissance plus restreint, par exemple l'astronomie. Pour (Nazarenko, 2004) cette opposition se formule par le nombre et la diversité des applications auxquelles les ressources peuvent servir.

Les textes contiennent des éléments linguistiques plus ou moins visibles qui peuvent être utiles pour la construction d'ontologies, notamment les candidats-termes, les entités nommées, les relations lexicales ou spécialisées et les classes de mots. Dans la suite de cette section, nous présentons chacun de ces éléments en précisant en quoi ils peuvent servir de support à la COT.

2.2.1 Candidats-termes

L'analyse terminologique d'un corpus permet d'identifier des mots simples ou composés qui semblent avoir un fonctionnement terminologique, c'est-à-dire qui relèvent d'un vocabulaire spécialisé doté d'une sémantique relativement stable et consensuelle au sein d'une situation de communication déterminée (Jacquemin et Bourigault, 2003). L'extraction terminologique repose sur des méthodes statistiques, linguistiques ou le plus souvent mixtes (repérage de séquences d'éléments remarquables, comme un nom suivi d'un adjectif, et filtrage selon des critères de redondance) (Nazarenko *et al.*, 2009).

Dans le processus de construction d'ontologies, l'extraction terminologique permet de repérer la trace des concepts qui sont **mentionnés** dans le corpus. Elle sert donc au repérage du vocabulaire conceptuel. Il est erroné, bien que fréquent, de faire l'association « terme = concept ». En effet, un terme peut représenter plusieurs concepts (par exemple le terme « étoile » peut signifier, selon le contexte, « astre en réaction de fusion thermonucléaire » ou bien « point lumineux dans le ciel nocturne ») et un concept peut être dénoté par plusieurs termes s'ils sont suffisamment synonymes. Les termes sont des unités de la langue, tandis que les concepts sont des éléments du modèle conceptuel.

2.2.2 Entités nommées

On désigne par « entités nommées » des expressions textuelles qui s'apparentent à des noms propres. Il s'agit en fait de noms qui désignent de manière relativement univoque (Ehrmann, 2008) des entités référentielles. Le terme est issu des campagnes d'évaluation MUC [11] (Sundheim et Chinchor, 1993) pour ce qu'on appelle aujourd'hui l'extraction d'information. Il désignait au départ les noms de personnes, de lieux, de compagnies, etc., mais a été élargi depuis : expressions numériques, url, noms de gènes ou de médicaments, etc.

La construction d'ontologies utilise principalement les entités nommées comme un moyen pour peupler les ontologies avec des instances de concepts (Magnini *et al.*, 2006; Tanev et Magnini, 2008). L'idée sous-jacente est par exemple de créer une instance du concept *personne* à partir de chacun des noms de personnes différentes repérés dans le corpus. Il s'agit donc à la fois de repérer en corpus les unités textuelles correspondant à des entités pertinentes pour le domaine (noms de personnes ou de gènes, dates, etc. selon les corpus et les applications) et de les associer à un concept de l'ontologie (Fort *et al.*, 2009; Omrane *et al.*, 2010).

2.2.3 Relations lexicales

Les relations lexicales sont des relations linguistiques fréquentes indépendantes du domaine d'application qui ont été formalisées par les linguistes.

L'hyperonymie est représentée par la relation de subsomption entre les réalisations de concepts dans le texte. Elle constitue un guide pour la tâche de conceptualisation. L'hyperonymie est exprimée au niveau de la phrase (« un chat est un mammifère », « Parmi les coiffures nous pouvons distinguer les chapeaux et les couronnes », etc.) ou au niveau du mot composé, par exemple de « moteur gpl » on peut déduire qu'il s'agit d'un moteur.

La synonymie exprime une proximité sémantique entre mots ou expressions. La synonymie n'est jamais parfaite (sinon la langue aurait fait disparaître l'un des mots devenu inutile), il existe toujours une différence liée aux connotations véhiculées, au registre de langue ou encore au contexte d'emploi des mots. Dans le cadre de la COT les relations de synonymie peuvent traduire, selon un choix de modélisation, une relation d'équivalence entre concepts ou un enrichissement de l'ancrage lexical d'un concept, c'est-à-dire des termes qui, pour le domaine de modélisation choisi, dénotent le même concept.

La méronymie, relation de partie à tout, est parfois utilisée pour structurer l'ontologie à la place des relations « est-un » entre concepts (Berland et Charniak, 1999; Lopez

11. Message Understanding Conferences.

et al., 2002; Rosse et Mejino, 2003), mais de manière moins fréquente. L'antonymie, qui exprime les contraires, peut être utile pour le repérage de concepts disjoints. L'antonymie n'est que rarement exprimée dans les textes, mais se retrouve plutôt sous la forme de connaissances implicites.

2.2.4 Relations spécialisées

Les relations spécialisées sont des relations sémantiques qui sont dépendantes d'un domaine particulier. Par exemple dans le domaine du droit nous pourrions avoir la relation « travaille pour » qui aurait pour domaine les employés et pour co-domaine les entreprises. Les relations spécialisées dans le texte sont des indices qui permettent d'enrichir l'ontologie avec majoritairement des relations entre concepts, mais pas obligatoirement : il est possible de modéliser une relation spécialisée sous la forme d'un concept, par exemple la phrase « La voiture **roule sur** la chaussée » peut être modélisée avec la relation `rouler-sur` ou *via* le concept de `véhicule autorisé à circuler`.

2.2.5 Classes sémantiques

Les classes sémantiques sont des regroupements de mots (ou plus rarement de termes) qu'on tend à interpréter comme concepts. L'hypothèse distributionnelle (Harris *et al.*, 1989) stipule que des éléments textuels sont sémantiquement proches s'ils partagent des contextes. Il est possible sous cette hypothèse de regrouper des mots sémantiquement proches en étudiant leurs distributions au sein du corpus (une fenêtre de mots, un contexte syntaxique, etc.). Dans le cadre de la COT ces classes reflètent parfois des concepts que l'expert doit étiqueter, mais pas toujours : il peut arriver que les classes de mots comportent des intrus ou soient issues d'artefacts linguistiques.

2.2.6 Axiomes et règles

Certains textes expriment des connaissances qu'il peut être intéressant de modéliser en tant qu'axiomes ou règles dans une ontologie. Par exemple si nous prenons la phrase « Tout ordinateur possède au moins un processeur » et que nous choisissons de modéliser les concepts d'ordinateur et de processeur reliés par la relation « possède », nous pouvons axiomatiser l'ontologie en ajoutant une contrainte de cardinalité sur cette relation.

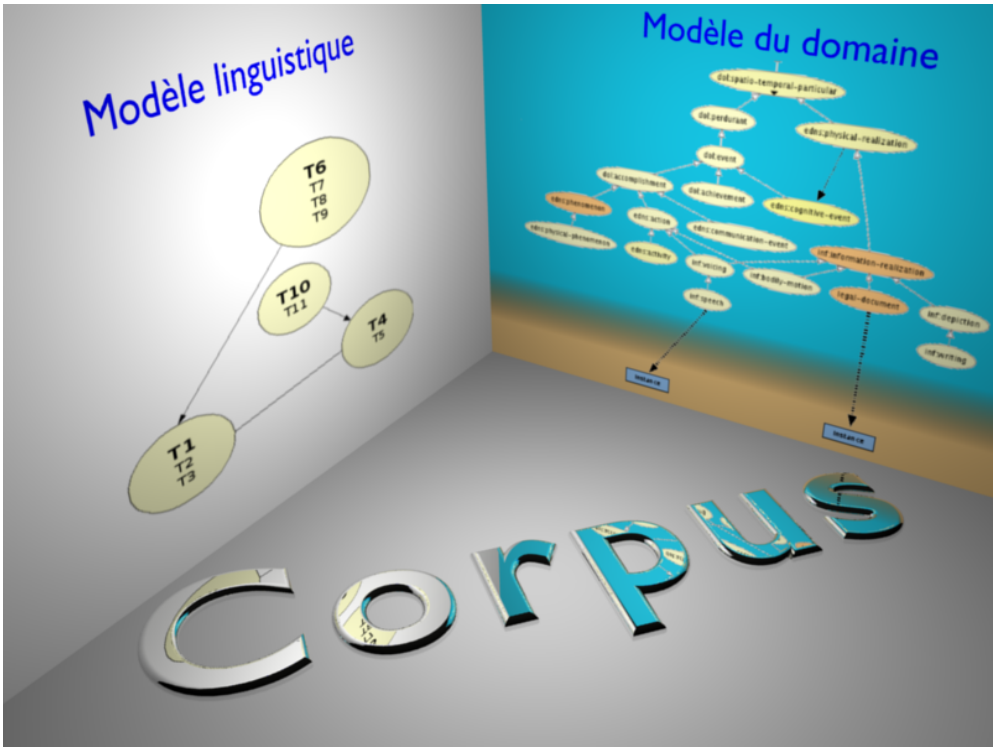

FIGURE 2.4 : Les trois plans de la conceptualisation

2.3 Conclusion

Tout au long de ce chapitre nous avons tenté, au travers d'une approche historique, de définir la notion d'ontologie. Nous avons ensuite étudié les éléments remarquables dans les textes qui peuvent se révéler utiles pour la construction d'ontologies. Ce chapitre peut s'illustrer par le schéma de la figure 2.4, présentée dans (Mondary *et al.*, 2008). Les textes, situés sur le plancher, constituent le socle de la modélisation. D'eux il est possible d'extraire un modèle linguistique qui contiendra les éléments qui sont exprimés, directement ou indirectement, dans les textes. Le plan ontologique, au fond, correspond à un aboutissement de la modélisation. Le passage direct du corpus (sur le plancher) au modèle du domaine correspond à une modélisation manuelle. La semi-automatisation se fait au travers du modèle linguistique ; mais passer de ce dernier au modèle conceptuel ne peut se faire qu'en effectuant d'importants choix de modélisation, qui relèvent à la fois de la granularité de la description désirée et des objectifs de l'ontologie.

Le chapitre suivant est consacré à la présentation des stratégies qui permettent de passer des textes à une ontologie, ou plus précisément à un modèle linguistique qui va aider à la réalisation d'un modèle du domaine.

Chapitre 3

Des textes à l'ontologie

Sommaire

Nous avons vu que les ontologies sont un formalisme de représentation des connaissances très en vogue depuis les quinze dernières années. Mais si leur utilisation au sein de la mouvance du web sémantique est bien établie, la constitution des ontologies est une tâche reconnue comme difficile.

Ce chapitre est structuré en deux parties, la première fera écho au chapitre précédent en présentant des techniques qui permettent d'extraire les éléments du texte pouvant servir lors de la conceptualisation, la seconde partie est consacrée à l'étude des stratégies de COT, illustrée par des outils.

3.1 Comment extraire les éléments remarquables des textes ?

Dans cette section nous faisons l'inventaire des familles de méthodes qui peuvent être envisagées pour aider la conceptualisation dans le processus de COT. Nous pouvons distinguer les approches dites « directes » (ou à base de patrons), qui travaillent au niveau des connaissances exprimées explicitement dans le texte (comme les définitions), des approches « indirectes » qui tentent d'extraire des connaissances tacites disséminées dans les textes.

3.1.1 Approches directes

Il existe dans les textes certaines formes régulières qui constituent des indices d'une relation d'hyperonymie, par exemple « Les chats, les chiens, les poissons rouges et autres

animaux domestiques, etc. ». L'utilisation de ces formes a été popularisée par Hearst (Hearst, 1992), qui proposa des patrons d'extraction de l'hyperonymie. D'une manière générale, les patrons d'extraction sont dépendants de la langue dans laquelle est écrit le texte ; ils peuvent servir dans la COT à repérer les relations de subsomption, mais également les relations spécialisées (Morin, 1999; Séguéla, 2000; Pantel et Pennacchiotti, 2006; Aussenac-Gilles et Jacques, 2008), des axiomes et des règles (Völker *et al.*, 2008). Ces patrons produisent des résultats peu nombreux mais de bonne qualité, selon la nature du texte. Toutefois s'il existe des formes génériques qui expriment l'hyperonymie, il en va autrement des patrons spécialisés qui dépendent du domaine. La reconnaissance des entités nommées repose également sur des patrons : on repère un nouveau nom de personne quand on identifie un nom propre inconnu dans un contexte suffisamment caractéristique des noms de personnes.

Les approches directes explicitent le contenu textuel. Dans le cas de la recherche d'hyperonymie, les résultats sont généralement fiables mais peu nombreux (selon la nature du texte ; par exemple un texte encyclopédique, qui contient beaucoup de définitions, permettra de repérer plus de relations d'hyperonymie qu'un corpus de dépêches de presse (Jacques et Aussenac-Gilles, 2006)). Les approches à base de patrons sont également utilisées dans le repérage des candidats-termes (Daille, 2003; Aubin et Hamon, 2006). Cependant, une approche seulement linguistique se révèle insuffisante pour évaluer la qualité des candidats-termes retrouvés. Il est nécessaire de filtrer les résultats à l'aide d'une mesure statistique.

3.1.2 Approches indirectes

Nous appelons « indirectes » les approches qui regroupent des éléments qui ne sont pas explicitement reliés dans le texte. Certaines de ces approches produisent des classes d'éléments qu'il est ensuite possible d'organiser hiérarchiquement. Elles reposent toutes sur l'hypothèse distributionnelle (Harris *et al.*, 1989) qui suppose que des éléments distincts apparaissant dans des contextes (*cf infra*) similaires ont des chances d'être en relation sémantique et peuvent être regroupés.

L'application de l'hypothèse distributionnelle nécessite de cerner de manière préalable la nature des éléments que l'on souhaite regrouper. Ceux-ci peuvent être définis de manière plus ou moins linguistique : les documents (en recherche d'information), les phrases (par exemple pour le résumé automatique), les mots, les syntagmes, les sujets, etc.

Il convient ensuite de définir le contexte, c'est-à-dire la nature et la taille de l'environnement textuel à prendre en compte pour le calcul de la similarité entre mots. De manière symétrique au choix des éléments nous pouvons considérer des contextes plus ou moins linguistiques, comme une fenêtre de mots, une phrase, un document, une relation syntaxique (Hindle, 1990), etc. Un contexte syntaxique demandera forcément

plus de ressources qu'une fenêtre de mots, notamment la disponibilité d'un analyseur syntaxique pour la langue des textes.

Une fois que l'on sait repérer les éléments et leur contexte, il devient possible de regrouper tous les contextes associés à un élément dans une liste ou un vecteur. Une fois les listes de contextes constituées il est possible, si l'on se donne un critère de similarité (qui peut être booléen), de mesurer une proximité entre deux listes. Ainsi il est naturel de regrouper des éléments dont la distance entre listes de contextes est inférieure à un seuil déterminé.

Une fois les éléments regroupés en classes il serait souhaitable d'organiser ces classes en hiérarchies. Nous distinguons les approches ascendantes et les approches descendantes. Les premières visent à regrouper de petites classes en classes plus volumineuses par agrégation, tandis que les approches descendantes tentent de partitionner de volumineuses classes en ensembles plus petits. L'analyse de concepts formels est un autre paradigme de regroupement qui s'inscrit dans le cadre des approches indirectes, que nous détaillerons dans le chapitre suivant.

3.2 Méthodologies de construction

Nous faisons ici la distinction, parfois floue, entre les stratégies de construction issues d'une réflexion terminologique et celles provenant d'une expérience orientée apprentissage.

3.2.1 Les approches terminologiques

Les approches terminologiques, popularisées depuis le milieu des années 90 par le groupe de travail TIA [1] (Terminologie et Intelligence Artificielle), reposent sur l'identification préalable par des outils de TAL des éléments remarquables du texte (voir la section 2.2) et leur formalisation en ontologie par l'utilisateur.

Terminae

Terminae (Biebow et Szulman, 1999; Aussenac-Gilles *et al.*, 2008) est une *méthode* (schématisée sur la figure 3.1) supportée par un logiciel développé au sein du LIPN par Sylvie Szulman, guidant l'ontologue dans la conception de l'ontologie. Terminae s'appuie sur les résultats des outils de traitement automatique des langues (extracteur de termes, concordancier, détecteur de synonymie, analyseur syntaxique) pour extraire des éléments dans lesquels l'ontologue puise selon ses objectifs de modélisation. Une

1. http://tia.loria.fr/TIA

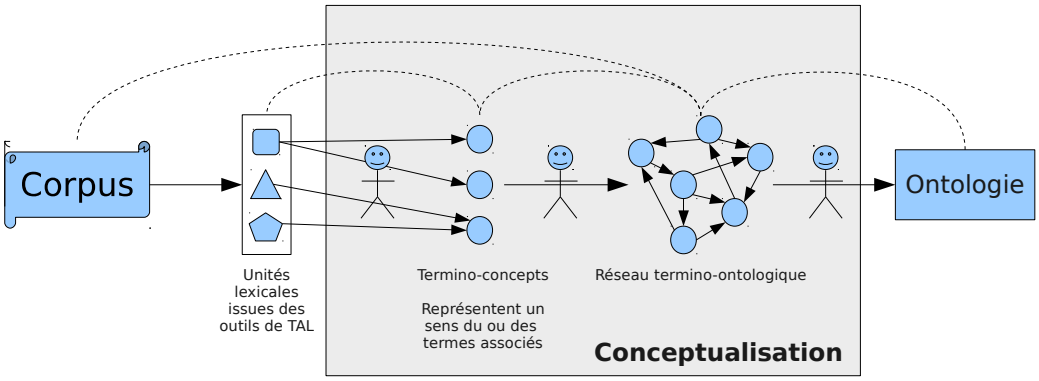

FIGURE 3.1 : Terminae

fiche terminologique centralise pour chaque terme (des recherches sont en cours pour prendre en compte les entités nommées dans la méthode Terminae (Omrane *et al.*, 2010, 2011)) les informations lexicales issues du corpus. Lors de la conceptualisation, l'ontologue associe à un terme un ou plusieurs *termino-concepts* qui représentent chacun un sens du terme. Le termino-concept est lui aussi doté d'une fiche (dite *fiche termino-conceptuelle*) qui centralise ses informations caractéristiques, notamment une définition en langage naturel, la liste des synonymes, etc. Les termino-concepts sont ensuite structurés manuellement en un *réseau termino-ontologique*, qui est ensuite formalisé en une ontologie. La conceptualisation reste manuelle, mais elle est assistée : l'ontologue dispose de différentes vues sur le matériau textuel. Terminae offre une traçabilité entre les concepts de l'ontologie en cours de construction et le corpus (voir la figure 3.1, sur laquelle les lignes pointillées représentent la traçabilité). Terminae distingue clairement le niveau terminologique du niveau conceptuel : le réseau termino-ontologique médiatise le passage du terme au concept. Cette méthode soulève les problèmes que l'on rencontre dans le passage du texte à l'ontologie et montre bien que ce travail est un processus **difficile**. C'est pourquoi nous cherchons au travers de cette thèse des stratégies pour en automatiser certaines parties.

TextToOnto

TextToOnto (Maedche et Staab, 2000) est une extension de la plateforme d'édition d'ontologies KAON (Oberle *et al.*, 2004) dédiée à la COT. TextToOnto propose des modules pour extraire des candidats-termes, des instances de concepts, des règles d'associations, des relations entre concepts, une hiérarchie de concepts. TextToOnto permet également d'enrichir une ontologie existante et de comparer deux ontologies entre elles.

Le module d'acquisition de hiérarchies a pour finalité de repérer de manière automatique les concepts et les relations de subsomption les liant. Pour identifier les concepts, il offre

le choix de se concentrer sur les n mots les plus fréquents (la valeur de n est réglable) ou sur une liste de termes obtenue à l'aide d'un extracteur de termes (intégré dans TextToOnto) et validée par l'utilisateur. La première variante, qui consiste à passer automatiquement des n mots les plus fréquents à des concepts est discutable. En effet, la fréquence n'est pas toujours un bon indicateur de la représentativité conceptuelle ; par exemple dans un texte juridique nous retrouvons souvent le mot « article », qui ne serait pas un concept très utile. En revanche, certains hapax (mots qui n'apparaissent qu'une seule fois) peuvent évoquer des notions intéressantes pour construire le modèle étudié (par exemple « situation présentant un danger »). Le fait qu'ils soient hapax peut être dû dans ce cas à la constitution du corpus.

Une fois sélectionnée la source (termes validés ou mots les plus fréquents) des candidats-concepts, nous avons le choix de l'approche utilisée pour extraire les relations de subsomption entre candidats-concepts : analyse de concepts formels (ACF) ou approches à bases de connaissances (patrons d'extraction + Wordnet (Miller, 1995)). Le système ajoute les candidats-concepts et leurs relations hiérarchiques au modèle en cours de construction. L'utilisateur doit ensuite retoucher à la main les résultats.

L'originalité de TextToOnto réside dans son idée de combiner des candidats-termes et l'ACF. Contrairement à Terminae, TextToOnto ne s'appuie pas sur une méthode pour guider l'utilisateur lors de la construction de l'ontologie. Les niveaux terminologiques et ontologiques ne sont pas différenciés. Nous nous inspirons de l'approche terminologique de l'ACF proposée par TextToOnto, mais nous souhaitons aller plus loin dans l'étude de l'interprétation des résultats en ontologie.

Upery

Upery (Bourigault, 2002) est un outil qui est le prolongement de l'analyseur syntaxique Syntex (Bourigault et Fabre, 2000; Bourigault *et al.*, 2005). Il a pour objectif de proposer à l'utilisateur des rapprochements de syntagmes (unités syntaxiques) en fonction des relations syntaxiques qu'ils partagent. Les éléments rapprochés sont à la fois des syntagmes maximaux, c'est-à-dire des mots composés les plus longs possibles, et des syntagmes réduits (des mots simples).

Reprenons l'exemple de Didier Bourigault. À partir de la phrase « Les roches cristallines résistent à l'érosion » Upery cherchera à rapprocher les éléments *roche, roche cristalline, cristalline, érosion* à l'aide des contextes *roche-ADJ, résister-SUJ, résister-à-érosion-SUJ, résister à, résister-SUJ-roche-à, résister-SUJ-roche-cristalline-à*. Pour cela, Upery propose trois mesures de proximité : le coefficient a, qui représente le nombre de contextes partagés par deux éléments, le coefficient *prox*, qui indique combien un contexte est spécifique à un terme (le « taux de dispersion » du contexte) et le coefficient j qui représente quant à lui le rapport entre le nombre de contextes partagés et le nombre total de contextes. Le logiciel Upery calcule pour chaque couple de termes les trois

coefficients et présente à l'utilisateur les couples pour lesquels ils dépassent un seuil fixé empiriquement. Le logiciel présente également des *doubles cliques*, c'est-à-dire des ensembles de termes et de contextes tels que chacun des termes apparait dans chacun des contextes.

Upery ne fournit pas de hiérarchie entre les éléments rapprochés. Il ne peut pas être intégré à Terminae pour structurer une liste de termes. Nous retenons de cet outil l'idée de rapprocher à la fois les mots simples et les mots composés, et de présenter à l'utilisateur les raisons de ces rapprochements.

3.2.2 Les approches non terminologiques

Les approches non terminologiques, le plus souvent issues de la tradition « apprentissage », sont celles qui partent du postulat qu'il est possible de conceptualiser de manière presque automatique en partant des textes, sans passer par une étape d'extraction et de validation terminologique. Dans certains cas les mots composés ne sont pas pris en compte, seuls les mots simples sont regroupés.

Text2Onto

Text2Onto (Cimiano et Völker, 2005) est un outil conçu pour construire des ontologies à partir de textes de manière complètement automatique (voir la figure 3.2). Il est composé de modules qui extraient à partir des textes des « concepts » (ou plutôt des candidats-termes), des relations entre ces concepts (relations d'équivalence, hiérarchiques, etc.) et des instances de concepts. Chaque module peut utiliser différents algorithmes et combiner leurs résultats : on peut ainsi combiner des patrons d'extraction « à la Hearst » et une ressource comme WordNet pour construire une hiérarchie. Text2Onto utilise l'architecture GATE (Cunningham *et al.*, 2002) pour prétraiter les textes. Text2Onto est le successeur officiel de TextToOnto, bien que n'offrant pas de fonctionnalités d'édition d'ontologies (Text2Onto n'utilise pas KAON. Cependant dans sa dernière version il est proposé sous la forme d'un plugin pour NeOn [2], un environnement d'ingénierie d'ontologies).

Les algorithmes d'extraction (TF.IDF, fréquence, etc.) sont combinables ; les résultats produits sont dotés d'une mesure de confiance entre 0 et 1. La sortie de Text2Onto, appelée POM (pour *Probabilistic Ontology Model*), est indépendante des langages de représentation des connaissances.

De notre point de vue, Text2Onto se présente comme une boîte à outils, mais n'est adossé à aucune méthode de construction. L'ontologue doit lui-même sélectionner les

2. http://neon-toolkit.org/wiki/Main_Page

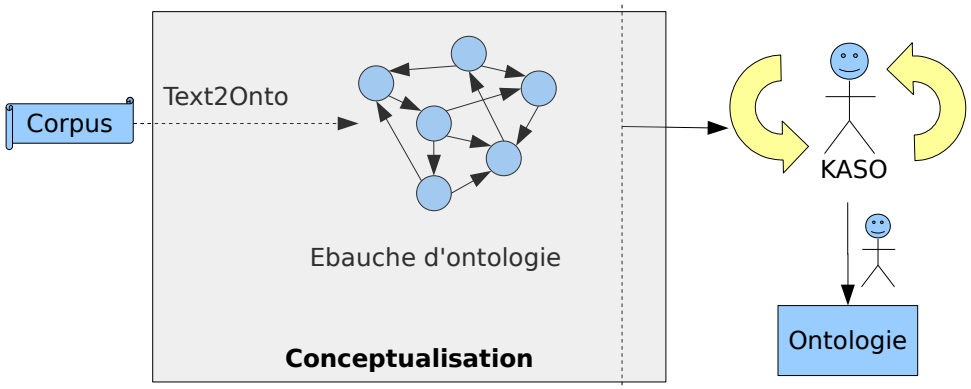

FIGURE 3.2: Text2Onto + KASO

algorithmes à utiliser. Il peut accepter ou rejeter les résultats obtenus, mais ne peut ni les modifier ni revenir aux parties des documents dont ils sont issus.

Le système KASO (Wang *et al.*, 2006), dont la conception est centrée utilisateur, est couplé à Text2Onto pour affiner le POM produit à l'aide de méthodes d'acquisition de connaissances telles que la mise en échelle (*laddering*) (Shadbolt *et al.*, 1999) et le tri de cartes (Upchurch *et al.*, 2001).

La nécessité d'avoir recours à des étapes en aval de Text2Onto montre les limites de l'approche tout automatique pour la conceptualisation qui, à notre avis, ne peut se passer de l'intervention humaine.

OntoGen

OntoGen (Fortuna *et al.*, 2006), implémente une approche semi-automatique pour la construction d'ontologies de thèmes (topic ontologies) à partir de collections de documents (voir la figure 3.3). C'est un outil interactif qui suggère à l'expert du domaine des classes de documents (les ronds sur la gauche de la figure) censées représenter des concepts. Il propose une dénotation sous la forme des mots-clés les plus représentatifs et leur associe automatiquement des instances (les documents). Il permet de visualiser l'ontologie en cours de construction. OntoGen exploite des algorithmes de fouille de textes non supervisés (k-means (Hartigan et Wong, 1979), LSI (Deerwester *et al.*, 1990)) ou supervisés (svm active learning (Tong et Koller, 2000)) selon une approche descendante. À chaque étape, le classifieur travaille sur la sous-collection associée au concept qui vient d'être construit.

OntoGen propose à l'expert, et c'est à ce dernier de choisir la proposition correcte parmi celles qui lui sont présentées. C'est une approche semi-automatique de la conceptualisation : les outils de classification de documents sont utilisés pour préparer le travail de conceptualisation, l'expert du domaine est guidé dans une démarche descendante,

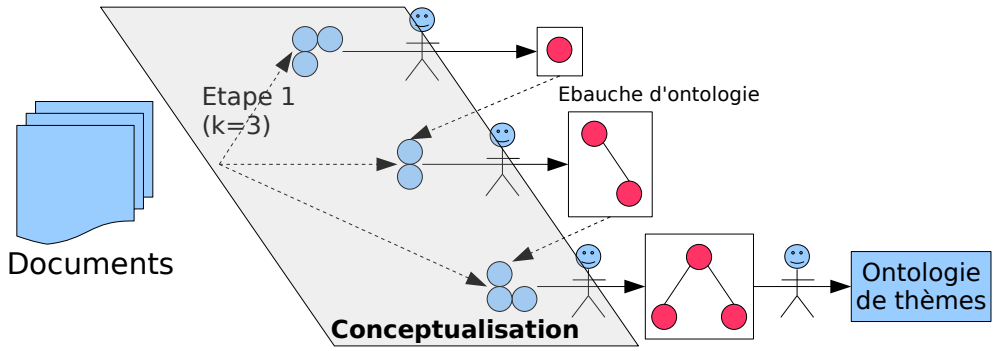

FIGURE 3.3: OntoGen

mais c'est lui qui construit les concepts et choisit quelles zones de l'ontologie affiner. Il convient d'insister sur le fait qu'OntoGen se focalise sur la construction d'ontologies de thèmes, et que les instances des concepts sont les documents.

Asium

Asium (Faure et Nedellec, 1999) est une méthode d'apprentissage interactive non supervisée pour la construction de hiérarchies à partir de textes. Asium utilise une analyse syntaxique pour extraire les sujets et des compléments d'objet. À l'aide des verbes associés et du nombre d'occurrences des sujets/compléments, Asium construit des classes de base qu'il regroupe niveau par niveau en proposant à chaque fois à l'utilisateur de valider les regroupements.

Une classe de base Asium est constituée des *noms* qui apparaissent en tant que sujet ou complément d'objet d'un même verbe. Par exemple de l'analyse syntaxique de la phrase « L'arbitre de touche chronomètre la partie. », Asium extrait les classes de base `<chronométrer>[sujet:arbitre(1)]` et `<chronométrer>[objet:partie(1)]`. Asium va ensuite regrouper les classes de base en utilisant un algorithme interactif par niveau qui repose sur la distance Asium. Cette distance prend en compte la fréquence des éléments partagés entre deux classes, le nombre d'éléments en commun et la cardinalité des classes. Une généralisation des classes est effectuée en ajoutant des connaissances obtenues par induction qui doivent être validées par l'utilisateur.

Par rapport à Upery, Asium produit une hiérarchie de regroupements. Certains d'entre eux peuvent être issus de connaissances inductives non exprimées dans le corpus. Cependant, Asium ne permet de regrouper que des mots simples.

3.3 Conclusion

Dans ce chapitre nous avons présenté d'une part la dichotomie qui existe entre les approches directes et les approches indirectes pour repérer les éléments utiles à la COT dans le texte. Nous avons ensuite illustré ces approches avec des outils qui s'inscrivent plus ou moins nettement dans une approche terminologique. Ces approches se distinguent par leur degré d'automatisation et la place du travail manuel de conceptualisation qu'il reste à effectuer pour obtenir une ontologie. Dans le cadre de cette thèse, nous choisissons de nous placer dans une approche terminologique, que nous souhaitons outiller à l'aide d'une méthode qui formalise le regroupement conceptuel, l'analyse de concepts formels.

Chapitre 4

Analyse de Concepts Formels (ACF)

Sommaire

Ce chapitre présente l'analyse de concepts formels[1] (ACF) et son utilisation avec des corpus dans la littérature. Après avoir présenté le formalisme mathématique de l'ACF, nous évoquons deux utilisations possibles de l'ACF et des textes pour la COT.

4.1 Présentation générale

L'analyse de concepts formels (Ganter et Wille, 1999) est un procédé qui permet de découvrir tous les regroupements possibles d'éléments ayant des caractéristiques communes. La structure résultante est appelée *treillis de concepts*. L'analyse de concepts formels est un sous-domaine des mathématiques appliquées qui correspond à la mathématisation des concepts de « concept » et de « hiérarchie de concepts ».

La notion centrale de l'ACF est le **contexte formel**. C'est un triplet $\mathbb{K} = (G, M, I)$ dans lequel G est un ensemble d'*objets formels*, M un ensemble d'*attributs formels* et I une relation binaire entre G et M appelée *relation d'incidence* de $\mathbb{K}$ et vérifiant $I \subseteq G \times M$. Un couple $(g, m) \in I$ signifie que l'objet $g \in G$ possède l'attribut $m \in M$. Plus intuitivement, un contexte formel est un tableau dans lequel les lignes sont les objets formels, les colonnes sont les attributs formels et une croix est placée à l'intersection d'une ligne et d'une colonne si l'objet possède l'attribut correspondant.

1. Nous préférons parler d'« analyse de concepts formels » plutôt que d'« analyse formelle de concepts » pour montrer la distinction entre les concepts formels et les concepts de l'ontologie.

À partir d'un contexte formel, l'ACF calcule les **concepts formels**, c'est-à-dire tous les regroupements d'objets formels ayant des attributs formels en commun. Pour cela, il est nécessaire de définir un opérateur sur les objets (respectivement sur les attributs) qui indique les attributs que possèdent tous les éléments d'un ensemble d'objets (respectivement d'un ensemble d'attributs). C'est ce que nous écrivons de manière plus formelle dans le paragraphe qui suit.

Soit $\mathbb{K}$ un contexte formel. Pour tout $A \subseteq G$ et $B \subseteq M$, on définit $A' = \{m \in M | \forall g \in A, (g,m) \in I\}$ et $B' = \{g \in G | \forall m \in B, (g,m) \in I\}$. A' est l'ensemble des attributs communs à tous les objets de A et B' est l'ensemble des objets possédant tous les attributs de B. Un couple (A,B) tel que $A \subseteq G$ et $B \subseteq M$, vérifiant $A' = B$ et $B' = A$ est appelé **concept formel** ; A est appelé l'**extension** du concept formel et B son **intension**.

$\mathfrak{B}(G,M,I)$ représente l'ensemble des concepts formels associés au contexte $\mathbb{K} = (G,M,I)$. Les concepts de $\mathfrak{B}(G,M,I)$ sont ordonnés par une relation $\leq$ telle que $(A1,B1) \leq (A2,B2) \leftrightarrow A1 \subseteq A2 \wedge B2 \subseteq B1$. $\mathfrak{B}(G,M,I)$ muni de la relation d'ordre $\leq$ forme un treillis noté $\underline{\mathfrak{B}}(G,M,I)$. L'infimum et le supremum de $T = \underline{\mathfrak{B}}(G,M,I)$, que nous nommerons dans la suite *top* et *bottom*, sont définis respectivement par :

$$\bigwedge_{t \in T}(A_t, B_t) = (\bigcap_{t \in T} A_t, (\bigcup_{t \in T} B_t)'')$$

$$\bigvee_{t \in T}(A_t, B_t) = ((\bigcup_{t \in T} A_t)'', \bigcap_{t \in T} B_t)$$

Le *bottom* est le concept formel dont l'intension est composée de tous les attributs formels. Son extension est généralement vide mais peut contenir les objets qui possèdent tous les attributs (une ligne complète dans la représentation matricielle du contexte formel). Le *top* est le concept formel dont l'extension est composée de tous les objets formels. Généralement d'intension vide, il peut contenir les attributs qui s'appliquent à tous les objets (une colonne complète dans la représentation matricielle du contexte formel).

4.1.1 Représentation et visualisation des concepts formels

Le treillis des concepts formels peut être représenté de manière visuelle dans un diagramme de Hasse. Chaque noeud de ce diagramme représente un concept formel, chaque arc représente une relation de subsomption. Pour améliorer la lisibilité du treillis il est possible de le représenter sous une forme réduite, qui consiste à reporter les étiquettes des intensions sur le concept le plus général possible dans le treillis et les étiquettes des extensions sur le concept le plus spécifique (voir la figure 4.1).

TABLEAU 4.1 : Signalétique de la représentation ConExp

	Concept formel « standard »
	Concept formel avec extension propre
	Concept formel avec intension propre
	Concept formel « blanc »

ConExp (Yevtushenko, 2000) est un outil qui permet de tracer des treillis de concepts à partir de contextes formels et qui offre la possibilité d'explorer interactivement le treillis. Il implémente une signalétique de référence dans la représentation des concepts formels, présentée dans le tableau 4.1. Les concepts formels que nous appelons « standard » sont caractérisés par l'existence d'une intension propre et d'une extension propre. L'*intension propre* d'un concept formel est le plus grand sous-ensemble de son intension qui n'est partagé par aucun concept formel plus général. L'*extension propre* d'un concept formel est le plus grand sous-ensemble de son extension qui n'est partagé par aucun concept formel plus spécifique. Nous avons choisi dans cette thèse d'utiliser ConExp pour représenter nos treillis, car cet outil offre une représentation compacte et modulable du treillis.

4.1.2 Exemple

Un contexte formel est présenté dans le tableau 4.2, le treillis associé sur la figure 4.1. Chaque rond dans le treillis représente un concept formel, que nous avons numéroté pour pouvoir y faire référence. Les attributs formels associés sont représentés dans les étiquettes grisées, les objets formels dans les étiquettes blanches. Les attributs (l'intension des concepts) sont représentés le plus haut possible et les objets le plus bas possible. Sur la figure 4.1, le concept 5 est $(Obj3, Attr1, Attr3, Attr4)$ et le concept 2 est $(Obj1, Obj3, Obj4, Attr4)$. Le concept 2, représenté à l'aide d'un rond évidé en bas, est sans objet propre ce qui signifie que d'autres concepts partagent des éléments de son extension. Respectivement, le concept 5 est sans attribut propre, d'autres concepts partageant un sous-ensemble de ses attributs. Le concept 4, dit concept blanc, est à la fois sans objet propre et sans attribut propre. Le concept 3 possède à la fois une intension propre et une extension propre. Les concepts 1 et 6 sont appelés *top* et *bottom* et correspondent respectivement à la conjonction des objets et des attributs.

TABLEAU 4.2: Contexte formel 1

	Attr 1	Attr 2	Attr 3	Attr 4
Obj 1	×			×
Obj 2		×	×	
Obj 3	×		×	×
Obj 4		×	×	×

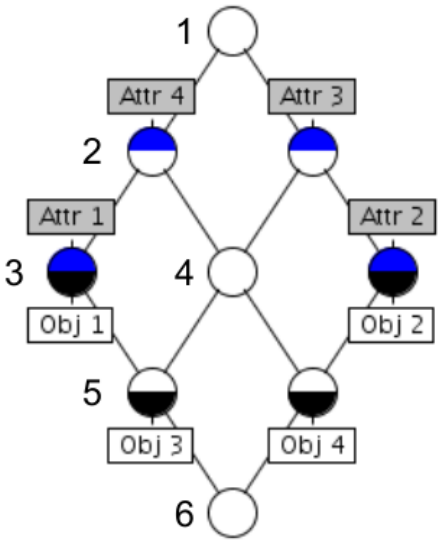

FIGURE 4.1: Treillis associé au contexte formel du tableau 4.2

4.2 ACF et textes

4.2.1 Construire une ontologie avec l'ACF

(Cimiano *et al.*, 2005) et (Bendaoud *et al.*, 2007) proposent de construire le contexte formel en s'appuyant sur les idées de (Hindle, 1990). En partant d'une analyse syntaxique des phrases obtenue automatiquement à l'aide d'un analyseur syntaxique, ils construisent le contexte formel en extrayant de cette analyse les noms et les verbes. Les objets du contexte formel sont les noms apparaissant en tant que sujets ou compléments d'objet directs. Les attributs du contexte formel sont les verbes de la phrase considérée, complétés du suffixe -ABLE lorsqu'ils proviennent d'un COD.

Prenons à titre d'exemple le texte suivant : *Une grève paralyse l'entreprise. Les travailleurs refusent le travail dominical. Les dirigeants refusent la grève.* En ne gardant que la tête lemmatisée des sujets et des compléments, nous obtenons le contexte du tableau 4.3. Le treillis résultant est présenté sur la figure 4.2.

TABLEAU 4.3 : Contexte formel exemple

	paralyser	paralyser-ABLE	refuser	refuser-ABLE
grève	×			×
entreprise		×		
travailleur			×	
dirigeant			×	
travail				×

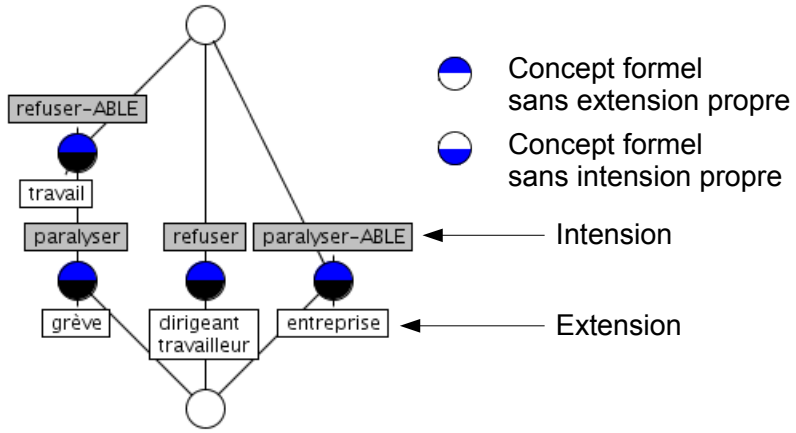

FIGURE 4.2 : Treillis exemple

La construction d'un contexte formel est déjà un passage du niveau du discours au niveau d'un modèle linguistique (voir la figure 2.4). Les occurrences et répétitions

observables dans les textes sont analysées en termes de présence / absence. Cette réduction d'information sera dans la suite pratique pour prendre en compte les termes : il n'est pas nécessaire de se soucier de leur compositionnalité comme dans une approche distributionnelle. Par exemple le terme « base de données » contient également le terme « donnée ». À la différence de l'analyse distributionnelle qui repose sur la fréquence des mots et qui impose de décider si « données » apparaît une ou deux fois (dans un ou deux contextes), l'ACF peut considérer sans difficulté qu'une occurrence apparaît dans plusieurs contextes à la fois.

L'approche automatique de l'ACF pour la COT de (Cimiano *et al.*, 2005) ne prend en compte que les mots. Nous pensons que la traduction automatique en ontologie proposée n'est pas viable, car elle mélange concepts et instances. Nous nous inspirons de la méthode utilisée pour construire un contexte formel à partir des textes et nous la combinons avec une analyse terminologique. Nous souhaitons aller plus loin dans l'étude de l'interprétation en ontologie : les concepts formels peuvent-ils se traduire en concepts ontologiques ?

4.2.2 Utiliser l'ACF pour enrichir une ontologie

Pactole (Bendaoud *et al.*, 2008) est un système d'enrichissement d'ontologies à partir de textes centré sur l'utilisation de l'analyse de concepts formels et de l'analyse de concepts relationnels. Ce système utilise des textes pour caractériser des concepts d'une ontologie existante en utilisant l'analyse de concepts formels. Toutes les ressources existantes (ontologies/thésaurus et textes) sont transformées en autant de contextes formels, puis ceux-ci sont fusionnés en utilisant l'apposition de contextes (une mise en commun des attributs formels par union des objets formels) ; le treillis de concepts formels est calculé avec le contexte fusionné. Des règles de dérivation sont utilisées pour exprimer automatiquement les constituants des treillis en logique de description.

Les étiquettes des concepts sont issues de l'ontologie initiale. Les concepts dont l'intension est uniquement issue des textes ne sont pas étiquetés, ils restent caractérisés par leurs intensions.

Ce système, bien que proche dans sa démarche, suit un but différent du nôtre : nous souhaitons utiliser l'ACF pour aider l'ontologue à conceptualiser, tandis que PACTOLE cherche à enrichir une ontologie avec des textes et ne se préoccupe pas de dénoter les concepts (ni de les présenter à l'utilisateur). Nous retenons la méthode de transformation du treillis en logique de descriptions.

4.3 Conclusion de l'état de l'art

Nous avons vu au travers de cet état de l'art que la COT *via* une approche tout automatique n'est pas réaliste, tandis qu'une approche terminologique qui offre une séparation claire entre les niveaux linguistique et conceptuel semble plus adaptée à la réalisation d'une ontologie de qualité. Toutefois, elle nécessite un important travail manuel. L'ACF est une méthode de regroupement que nous trouvons intéressante à intégrer à la COT dans le sens où elle permet d'obtenir une vue différente, « de plus haut niveau » sur le corpus qui faciliterait le travail de conceptualisation. Le choix de l'ACF par rapport à d'autres stratégies d'analyse distributionnelle se justifie par (1) le fait qu'elle se situe davantage vers le niveau conceptuel que les analyses distributionnelles, (2) qu'elle présente un aspect booléen qui facilite la prise en compte des mots composés par rapport à une analyse distributionnelle, (3) qu'elle ne déforme pas le corpus textuel *via* des inférences et des seuils arbitraires et (4) qu'elle est censée, d'après (Cimiano *et al.*, 2005), fournir les meilleurs résultats (en termes de F-mesure) par rapport aux méthodes de clustering hiérarchique. Dans le chapitre suivant, nous allons présenter notre système qui combine l'ACF avec l'analyse terminologique et les premières expérimentations que nous avons faites et les problèmes qu'elles ont mis en évidence.

Deuxième partie

L'ACF pour outiller la construction d'ontologies à partir de textes

Tout au long des quatre chapitres qui composent cette partie nous présentons notre contribution qui vise à outiller une approche terminologique avec l'ACF. Le premier chapitre est consacré à la présentation d'une première version de notre système et des expérimentations qui en découlent. Les problèmes que nous mettons en avant trouvent des solutions dans le chapitre suivant. Le troisième chapitre propose d'utiliser la nouvelle version de notre système lors d'expériences qui ont pour but d'évaluer qualitativement nos treillis. Le dernier chapitre est consacré à l'évaluation technologique de certains composants de notre système, et propose le plan d'une évaluation orientée utilisateurs.

Chapitre 5

Premières expérimentations, des problèmes

Sommaire

La construction d'ontologie à partir de textes (COT) est une tâche difficile qui ne peut se passer de l'intervention humaine. L'approche terminologique de COT, popularisée par le groupe TIA [1] et par la méthode Terminae, repose sur l'exploration d'une liste de candidats-termes. Cette liste, même si elle peut être ordonnée par les fréquences d'apparition des candidats-termes dans le corpus, reste non structurée. L'analyse de concepts formels est utilisée ici comme une méthode complémentaire d'exploration, de découverte et de prise en main d'un corpus ainsi que comme une méthode de structuration de la liste de candidats-termes.

Ce chapitre présente les expérimentations que nous avons réalisées afin d'étudier le comportement de l'analyse de concepts formels couplée à une approche terminologique pour aider à la construction d'ontologies à partir de textes. Ces différentes expérimentations vont mettre en évidence les limites de l'approche ACF appliquée aux textes.

1. http://tia.loria.fr/TIA

5.1 Présentation des corpus utilisés pour nos expériences

Nous avons expérimenté notre système sur trois corpus recouvrant trois domaines distincts. Le premier domaine étudié est l'astronomie, le deuxième le droit et le troisième la cuisine. Nous avons choisi trois styles de rédaction différents pour nous permettre d'étudier la robustesse de notre approche. Le premier style de rédaction est le style encyclopédique, il est illustré par le corpus de l'astronomie. Nous expérimentons le style impératif à l'aide des recettes de cuisine, tandis que le corpus de droit représente une forme d'expression plus complexe, avec des phrases très longues. Tous nos corpus sont en français [2], l'analyseur syntaxique utilisé par notre système ne fonctionnant qu'avec cette langue. Nous avons fait en sorte d'utiliser des corpus de taille identique, environ 55 000 mots [3], pour pouvoir mettre en avant le lien qui peut exister entre la nature du corpus et la qualité de nos résultats. La taille de cinquante-cinq mille mots a été retenue comme étant un compromis entre la quantité d'informations contenue dans le corpus et la lisibilité par l'humain.

5.1.1 Un corpus difficile : le droit

Le domaine du droit est représenté par des documents en français rédigés par l'Organisation Internationale du Travail [4]. Le choix de ce corpus provient de notre participation, en début de thèse, au projet « Régimes de dialogue social : Droits des travailleurs, négociation collective et définition négociée des politiques » [5].

L'OIT édite des conventions [6] qui ont le statut de traités internationaux. Les états ratifient ces conventions et les font appliquer. Des irrégularités dans l'application sont signalées par les syndicats, les entreprises, etc. à l'OIT qui édite chaque année un recueil de rapports. L'objectif de ce projet est d'isoler des catégories de violations des conventions pour obtenir des statistiques par pays, continent, etc. en indexant les nouveaux rapports de violations *via* une ontologie des violations. L'ontologie à construire devait permettre de spécifier des relations entre les violations, notamment des chaînes de causalités.

Le projet s'étant achevé, nous n'avons pas mené à son terme la construction de l'ontologie ; néanmoins nous avons choisi au début de notre thèse de conserver un

2. La justification de ce choix se trouve dans la description du corpus de droit.
3. Le calcul du nombre de mots s'effectue en comptant les tokens de Treetagger, à l'exception des symboles de ponctuation et des nombres.
4. http://www.ilo.org/ilolex/french/
5. http://www.ruig-gian.org/research/projects/project_f.php?ID=16
6. Il existe actuellement 188 conventions, publiquement accessibles sur http://www.ilo.org/ilolex/french/convdisp1.htm

> Après avoir décidé que ces propositions prendraient la forme d'une convention internationale, adopte, ce huitième jour d'octobre mil neuf cent quatre-vingt-sept, la convention ci-après, qui sera dénommée Convention sur la protection de la santé et les soins médicaux (gens de mer), 1987. Article 1 1. La présente convention s'applique à tout navire de mer, de propriété publique ou privée, qui est immatriculé dans le territoire de tout Membre pour lequel la convention est en vigueur et qui est normalement affecté à la navigation maritime commerciale. 2. Dans la mesure où, après consultation des organisations représentatives d'armateurs à la pêche et de pêcheurs, l'autorité compétente considère que cela est réalisable, elle doit appliquer les dispositions de la présente convention à la pêche maritime commerciale. 3. En cas de doute sur la question de savoir si un navire doit être considéré comme affecté à la navigation maritime commerciale ou à la pêche maritime commerciale aux fins de la présente convention, la question doit être réglée par l'autorité compétente après consultation des organisations d'armateurs, de marins et de pêcheurs intéressées.

FIGURE 5.1 : Extrait du corpus de droit

sous-ensemble du corpus que nous avions constitué pour construire l'ontologie des violations. Notre corpus est constitué de quatorze conventions [7] pour un total de 53 321 mots.

Un extrait du corpus est présenté sur la figure 5.1. Nous pouvons remarquer que les phrases sont plutôt complexes, ce qui ne facilite pas les traitements, que ce soit au niveau de l'analyse syntaxique ou de la compréhension du texte.

5.1.2 Une langue simplifiée : la cuisine

Nous avons choisi de tester l'ACF sur un type de corpus particulier, les recettes de cuisine. Ce sont des textes dont les phrases sont simples, courtes et à l'impératif. La figure 5.2 présente un extrait du corpus ; il est constitué de 392 recettes que nous avons récoltées sur internet et pèse 54 527 mots.

Le choix d'un corpus de cuisine vient à la fois de son style particulier qui devrait engendrer un minimum d'erreurs lors de l'analyse syntaxique et du fait que ce genre de corpus a été utilisé pour valider Asium, ce qui indique qu'il est *a priori* viable pour la COT.

7. 29, 81, 129, 143, 157, 162, 164, 165, 167, 168, 170, 174, 176 et 184.

DIPLOMATE A LA CONFITURE.
Pour 6 personnes : 200 g de biscuits à la cuiller, 250 g de marmelade d'abricots, 2 dl d'eau, 30 g de sucre semoule, 150 g de crème fraîche, 1/2 dl de lait, 30 g de sucre glace, 1 sachet de sucre vanillé, 2 c à soupe de kirsch.
Dans un bol, mettre l'eau et le sucre semoule, ajouter le kirsch. Il faut attendre que la totalité du sucre soit bien dissoute. Choisir un moule à charlotte ou à flan. Placer les biscuits à la cuiller un à un dans l'eau sucrée parfumée au kirsch et les disposer au fur et à mesure sur le fond du moule et les parois, le côté plat des biscuits mis contre la paroi du moule. Remplir ensuite le moule en alternant une couche de marmelade et une couche de biscuits imbibés. Terminer par une couche de biscuits. Poser une assiette et un poids sur le moule laisser au frais plusieurs heures ou une nuit. Au moment de servir démouler et entourer le diplomate de crème fraîche liquéfiée au préalable avec un peu de lait aromatisé à la vanille et sucré au sucre glace.

FIGURE 5.2 : Extrait du corpus des recettes de cuisine

5.1.3 Des textes explicites : l'astronomie

Le corpus de l'astronomie est un corpus rédigé dans un style *encyclopédique*, que nous avons constitué en vue d'obtenir un corpus plus directement explicite que celui du droit. Le choix de ce domaine est le résultat d'une expertise personnelle qui nous a permis de construire une ontologie de référence (voir la section 7.4).

Le corpus est constitué d'articles de Wikipedia français obtenus en sélectionnant ceux décrivant les objets célestes usuels ainsi que leur évolution : *amas globulaire, amas ouvert, amas stellaire, astéroïde, comète, étoile binaire, étoile, évolution des étoiles, galaxie, géante bleue, géante rouge, naine blanche, naine brune, naine jaune, naine noire, naine rouge, naissance des étoiles, nébuleuse, nova, planète, pulsar, quasar, satellite naturel, Soleil, supergéante rouge, supernova, trou noir*. Le corpus contient 54 121 mots. La figure 5.3 présente un extrait du corpus.

5.2 Présentation de notre premier système

Notre premier système permet de combiner l'ACF avec une approche terminologique. C'est un système modulaire, dont le processus de fonctionnement global est présenté sur la figure 5.4. Le système s'articule autour du module d'extraction du contexte formel.

Amas globulaire.
En astronomie, un amas globulaire est un amas stellaire très dense, contenant typiquement une centaine de milliers d'étoiles distribuées dans une sphère dont la taille varie de 20 à quelques centaines d'années-lumière. Leur densité est ainsi nettement plus élevée que celle des amas ouverts. Les étoiles de ces amas sont généralement des géantes rouges.
On compte 150 amas globulaires dans notre Galaxie. Mais il en existe sans doute d'autres, indétectables car masqués par le centre galactique. Les amas globulaires font partie du halo galactique, ils orbitent autour du centre galactique à une distance variant de 1 à 100 kilo-Parsec. C'est par leur étude que Harlow Shapley, en 1918, a pu déterminer la position du Soleil au sein de la Galaxie. Comme les amas globulaires contiennent les étoiles les plus âgées d'une galaxie, ils contribuent également de façon importante à l'étude de l'évolution des étoiles et des galaxies.

FIGURE 5.3: Extrait du corpus des objets célestes

Ce module requiert en entrée une analyse syntaxique en dépendance et une liste de termes. Il produit un contexte formel dans lequel les objets formels sont les sujets et les compléments d'objet directs, et les attributs formels sont les verbes associés, suffixés de -ABLE dans le cas des COD. Par exemple la phrase « Jean mange une glace » produira le contexte formel composé des relations `Jean:manger` et `glace:manger-ABLE`.

La liste de candidats-termes est acquise à l'aide de YaTeA (Aubin et Hamon, 2006), un extracteur de termes développé au LIPN qui combine patrons linguistiques et désambiguïsation statistique. Nous avons choisi cet extracteur de termes en raison de sa bonne intégration au sein de Terminae et aussi parce qu'il fonctionne sur le français et sur l'anglais.

L'analyse syntaxique en dépendances est confiée au système Syntex (Bourigault *et al.*, 2005). Nous avons choisi ce système en raison des bons résultats qu'il a obtenus lors de la campagne EASY, et parce que contrairement à l'anglais, en français les analyseurs syntaxiques en dépendances ne sont pas légion. De plus l'historique de Syntex dans la construction d'ontologies (au travers d'Upery) a fait de lui le candidat que nous avons retenu. Ces deux systèmes (YaTeA et Syntex) nécessitent en entrée un corpus étiqueté morpho-syntaxiquement ; nous avons confié cette tâche au TreeTagger (Schmid, 1994).

Une fois le contexte formel extrait, l'analyse de concepts formels est effectuée à l'aide du logiciel ConceptExplorer (Yevtushenko, 2000). Ce logiciel prend en charge les étapes de calcul des concepts formels et d'affichage du treillis résultant. Son interface lui permet d'explorer le treillis de manière dynamique, mais manque en revanche d'outils

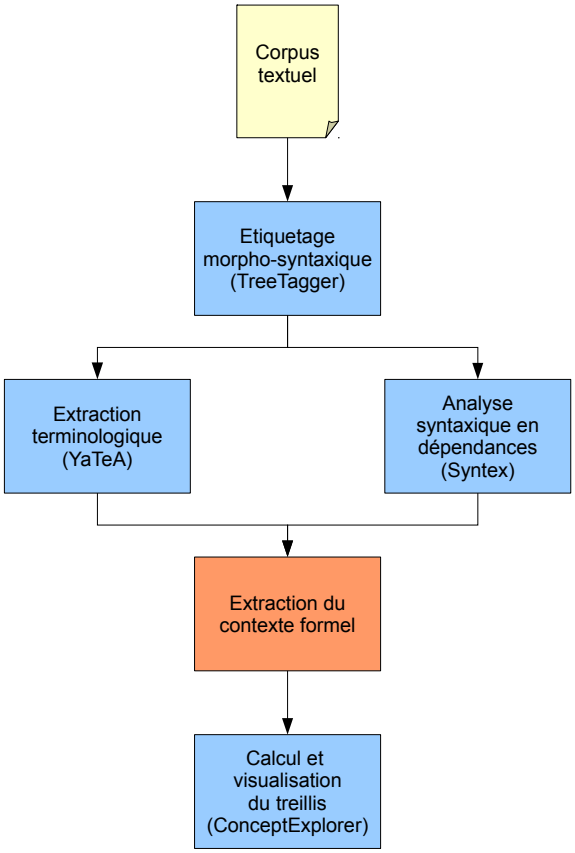

FIGURE 5.4: Vue globale du processus de fonctionnement de notre système

de recherche et d'affichage partiel des concepts formels. C'est cependant l'outil qui nous a paru le plus intuitif à utiliser, sa représentation des concepts formels est *de facto* devenue un standard.

Les temps de calculs nécessaires pour construire les concepts formels à partir du contexte formel ne sont pas rédhibitoires, c'est plus l'affichage du treillis qui pose problème à partir d'un certain nombre de concepts formels (comme nous le verrons plus loin). Le calcul du contexte formel à l'aide du module d'extraction présenté ci-après n'est pas instantané mais reste accessible (environ trente secondes pour un corpus de 50 000 mots sur un ordinateur de bureau classique). Notre implémentation n'est pas optimisée pour la vitesse d'exécution. Toutefois, elle n'est théoriquement pas limitée par la taille du corpus.

Le module d'extraction du contexte formel est la pierre angulaire de notre système. Il a pour objectif de repérer et d'extraire les objets et les attributs formels d'un texte, en vue d'appliquer l'ACF. La figure 5.5 détaille sa logique. Ce module prend en entrée une liste de termes sous forme fléchie et lemmatisée (la double forme sera utile pour la procédure d'appariement décrite plus loin) et une analyse syntaxique en dépendances, dont nous

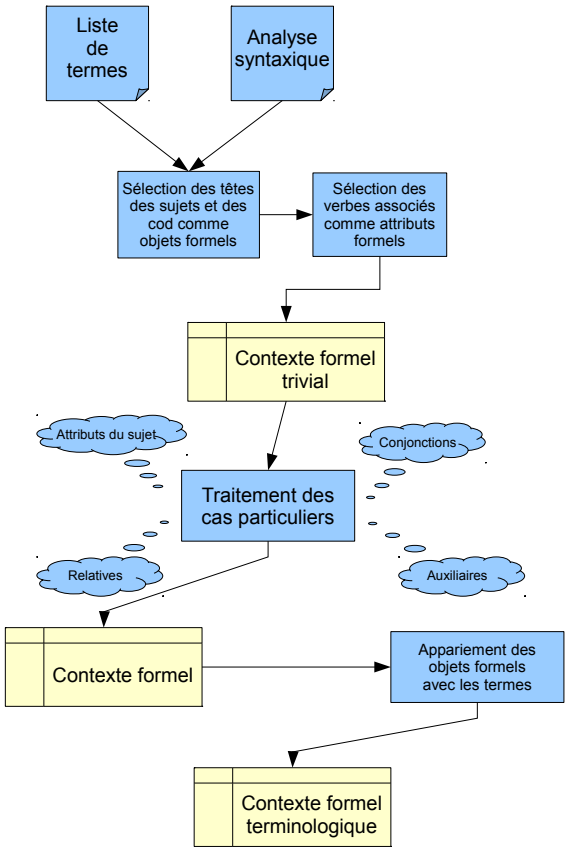

FIGURE 5.5 : Le module d'extraction du contexte formel en détail

présentons un exemple sur la figure 5.6. Pour chaque phrase, notre système sélectionne les têtes des sujets et des compléments d'objet directs pour les inclure dans le contexte formel comme objets formels. Le verbe associé aux objets sélectionnés est ajouté comme attribut formel ; ceux qui apparaissent dans une relation COD sont suffixés de « -ABLE » dans le contexte formel, afin d'exprimer une faculté, une possibilité. Nous avons choisi de ne pas utiliser les adjectifs épithètes comme attributs formels, car notre objectif est de prendre en compte les termes et ceux-ci en contiennent souvent. Nous utilisons donc les adjectifs épithètes uniquement à l'intérieur des termes qui apparaissent comme objets formels.

Le contexte formel ainsi construit est appelé « contexte formel trivial » en raison de certaines tournures phrastiques moins régulières qui nécessitent une analyse plus poussée pour produire les objets / attributs formels attendus. Le module de traitement des cas particuliers se charge de corriger le contexte formel trivial en adaptant la recherche des sujets / compléments. Notre système prend en charge les auxilaires, les adjectifs attributs et d'une manière moins exhaustive les propositions relatives et les conjonctions de coordination. Le tableau 5.1 illustre ces cas particuliers et leur transformation en objets formels par notre système.

```
<SEQ id="T_1">
   <TXT>Jean regarde Marie .</TXT>
   <tokens>
     <t i="1" l="Jean" f="Jean" c="NomPr" p="NP"/>
     <t i="2" l="regarder" f="regarde" c="VCONJS" p="V"/>
     <t i="3" l="Marie" f="Marie" c="NomPr" p="NP"/>
     <t i="4" l="." f="." c="Typo" p="T"/>
   </tokens>
   <dependances>
     <d r="SUJ" s="1" c="2"/>
     <g r="SUJ" s="2" c="1"/>
     <g r="OBJ" s="2" c="3"/>
     <d r="OBJ" s="3" c="2"/>
   </dependances>
 </SEQ>
```

FIGURE 5.6 : Analyse syntaxique en dépendances, le format Syntex XML

TABLEAU 5.1 : Exemples de tournures particulières traitées par notre système

Exemple	Contexte trivial
Attribut du sujet : « Jean est heureux »	Jean : être (attendu *être heureux*)
Conjonction : « J'aime le pain et le chocolat »	et (att. *pain, chocolat*) : aimer-ABLE
Relative : « Le voisin qui parle anglais »	qui (att. *voisin*) : parler
Auxiliaire avoir : « Jean a regardé Marie »	Jean :avoir (att. *regarder*), Marie :regarder-ABLE

Cette étape de traitement des cas particuliers du contexte formel trivial soulève la question plus large du passage du niveau syntaxique au niveau sémantique. Trouver l'antécédent d'un pronom relatif ou le verbe attaché à un auxiliaire sont des problèmes que l'on peut traiter de manière relativement fiable et simple. En revanche, d'autres phénomènes comme l'anaphore pronominale, la conjonction de coordination et la négation sont bien moins triviaux. Cela suggère l'existence d'une couche supérieure de l'analyse syntaxique qui prendrait en compte tous les artefacts de la langue pour se rapprocher du niveau sémantique.

Obtenir ce niveau d'analyse nécessiterait éventuellement de s'affranchir de la frontière de la phrase, notamment pour résoudre les anaphores, mais permettrait un repérage systématique des objets et des attributs formels qui seraient notoirement plus compréhensibles. C'est ce que tente d'obtenir notre module de traitement des cas particuliers, sans pour autant prétendre à l'exhaustivité. Les objets formels du contexte construit sont constitués des têtes des sujets / compléments, ce sont des mots. Nous avons à notre disposition une liste de termes, que nous n'avons pas utilisée jusqu'à maintenant. Elle va nous servir à enrichir les objets formels, à les préciser.

Appariement des termes et des objets formels Dans l'analyse syntaxique en dépendances, les sujets et compléments d'objet sont repérés par leur tête, ce sont des mots. L'analyse terminologique nous donne un moyen de préciser les éléments intéressants au sein des objets du contexte formel, c'est-à-dire de remplacer un objet formel *mot* par un objet formel *terme* (par exemple en astronomie l'objet formel « supernova » serait précisé en « supernova à effondrement de cœur »). L'ACF nous permettra ensuite de structurer les termes qui apparaissent en corpus en fonction des attributs qu'ils partagent.

L'appariement se fait en deux étapes et nécessite une liste de termes sous forme fléchie et lemmatisée. Pour chaque phrase, le système recherche en texte intégral les termes qui y apparaissent, d'abord sous leur forme fléchie puis en cas d'échec sous forme lemmatisée, d'où la nécessité de conserver une liste sous forme fléchie avec les variations. La deuxième étape intervient pour chaque objet formel repéré dans la phrase. Le système va tenter d'apparier l'objet formel avec l'un des termes repérés lors de l'étape précédente (en privilégiant sa forme fléchie) en parcourant la sortie de Syntex (voir les balises « t » de la figure 5.6). Le lecteur est en droit de se demander pourquoi nous n'avons pas simplement choisi de reconstituer des groupes nominaux en suivant les dépendances indiquées par Syntex. La réponse est que cela est possible (cela fera même l'objet d'une extension de notre système présentée plus loin) mais que les groupes ainsi reconstitués n'ont pas subi de validation statistique comme ceux d'un extracteur de termes, ce qui signifie qu'ils seront probablement très bruités (voir pour cela la section 6.3.2).

Dès lors que l'appariement s'est déroulé de manière correcte, le mot objet formel est remplacé par le lemme du terme repéré, les attributs formels associés sont conservés. À la légitime question « pourquoi avoir choisi de travailler sur le corpus sous une forme non lemmatisée ? Cela nécessite une extraction terminologique qui inclut les variations. » nous répondrons que cela permet : un de tenter de nous affranchir des erreurs de mises en correspondance des lemmes (l'extracteur de termes et l'analyseur syntaxique n'utilisent pas forcément les mêmes stratégies de lemmatisation) et deux de permettre à l'utilisateur de corriger voire d'étendre la lemmatisation proposée par le lemmatiseur (par exemple TreeTagger lemmatise supernova et supernovas par « supernova » mais « supernovæ » en « supernovæ », alors que c'est le pluriel ; l'utilisateur a donc la possibilité en amont d'indiquer au logiciel que toutes ces formes fléchies se regroupent en « supernova »).

Une fois les objets formels appariés avec la liste de candidats-termes, nous obtenons un *contexte formel terminologique*. Dans la section suivante, nous appliquons notre système aux trois corpus présentés plus haut. Cette expérience va nous permettre de mettre en avant les problèmes soulevés par cette approche.

Tableau 5.2: Candidats-termes repérés

Corpus	Nombre de mots	Nombre de CT
Astronomie	54 121	4881
Droit	53 321	3124
Cuisine	54 527	3607

Tableau 5.3: Mesures quantitatives de l'application de l'ACF

	Astronomie	Droit	Cuisine
Objets formels	774	432	456
Attributs formels	1089	416	463
Concepts formels	1555	514	1777
–> % nb de mots	2,87%	0,96%	3,25%
–> dont /extension/ > 1	1416	446	1560
Hauteur du treillis	9	6	11
Nombre d'arêtes	4170	1179	5371

5.3 Injecter la connaissance terminologique au sein du processus d'ACF

La première étape de l'application de notre système consiste à procéder à une extraction terminologique. Le tableau 5.2 présente le nombre de candidats-termes extraits en fonction du corpus. Nous pouvons remarquer que le corpus d'astronomie est plus riche en candidats-termes que les deux autres. Ceci est dû à sa nature encyclopédique.

Le tableau 5.3 présente une ACF « état de l'art », c'est-à-dire sans prendre en compte l'aspect terminologique du corpus tel que présenté dans la section précédente. Elle va nous servir de référence afin d'étudier quantitativement et qualitativement l'impact de l'intégration de candidats-termes dans le processus d'ACF. Elle présente pour chaque corpus le nombre d'objets, d'attributs et de concepts formels, ainsi que la proportion de concepts formels par rapport au nombre de mots du corpus. La table présente également le nombre de concepts formels « de regroupement », c'est-à-dire ceux dont l'extension contient plus d'un élément, ainsi que le nombre d'arêtes du treillis. La hauteur du treillis indiquée représente le nombre de nœuds du chemin le plus long allant du « top » au « bottom ».

Le tableau 5.4 est à mettre en regard avec le tableau 5.3. Elle présente l'impact quantitatif de l'ajout de connaissances terminologiques dans l'ACF. Cet ajout se fait au moment de la construction du contexte formel, en précisant les objets formels du contexte à l'aide des candidats-termes (voir la section 5.2). Nous appelons le treillis qui en résulte un treillis *terminologique*.

TABLEAU 5.4 : Impact de l'ajout de connaissances terminologiques à l'ACF : treillis

	Astronomie	Droit	Cuisine
Candidats-termes	4881	3124	3607
Objets formels	1526 (+752)	770 (+338)	1010 (+554)
Attributs formels	1089	416	463
Concepts formels	1351 (-204)	513 (-1)	1252 (-525)
–> % nb de mots	2,49%	0,96%	2,29%
–> dont /extension/ > 1	1087 (-329)	415 (-31)	1003 (-557)
Hauteur du treillis	8 (-1)	5 (-1)	9 (-2)
Nombre d'arêtes	3241 (-929)	1112 (-67)	3460 (-1911)

Nous pouvons remarquer que la prise en compte des candidats-termes au sein du processus d'ACF laisse inchangé le nombre d'attributs formels, ce qui est normal puisque les candidats-termes sont des groupes nominaux qui sont injectés afin de préciser les objets formels, et que les relations syntaxiques utilisées pour constituer les attributs formels sont extérieures aux syntagmes nominaux. En revanche, le nombre de ces derniers double. Nous pouvons constater que bien que le nombre d'attributs formels reste inchangé et que le nombre d'objets formels double la quantité de concepts formels diminue (ou reste constant dans le cas du corpus de droit). Cela s'explique par le fait que les attributs formels sont plus dispersés dans le contexte formel. L'ajout de connaissances terminologiques ne semble pas compliquer le treillis.

Ces premières expérimentations nous permettent de dégager trois problèmes interdépendants, que nous présentons maintenant.

5.3.1 Problème 1 : Sensibilité au corpus

En observant les tableaux 5.3 et 5.4 nous constatons que le nombre de concepts formels varie du simple au triple. Or, les trois corpus sont de taille identique (environ 55 000 mots). Cela signifie que l'ACF est susceptible de « mieux fonctionner » sur un corpus que sur un autre ; reste à définir ce que signifie « mieux fonctionner ». Intuitivement on peut supposer qu'une plus grande quantité de concepts formels est susceptible de fournir de meilleurs regroupements. Encore faut-il définir la notion de qualité d'un regroupement, question qui sera soulevée dans la section 5.3.3. En supposant établi le lien entre les mesures quantitatives caractéristiques du treillis et sa qualité potentielle pour la COT, il devient envisageable de trouver dans le texte des indicateurs *a priori* de la phase d'ACF, voire d'analyse syntaxique qui reste coûteuse. Ces indicateurs seraient plus rapides à calculer qu'un ACF syntaxique complet et auraient une utilité au moment de la constitution du corpus, en indiquant rapidement à l'utilisateur une prévision de la quantité de concepts formels.

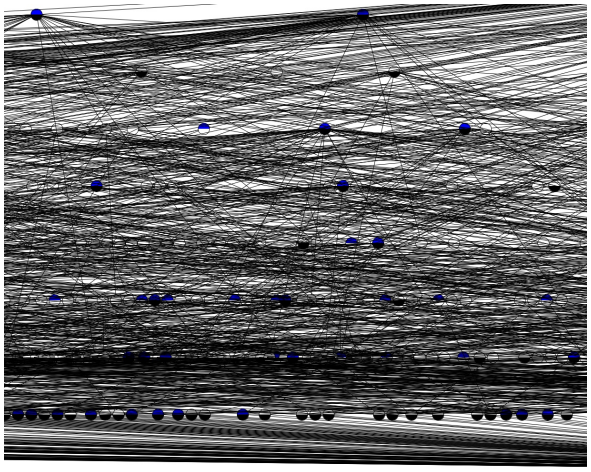

FIGURE 5.7: Treillis terminologique du corpus de la cuisine

5.3.2 Problème 2 : Trop de concepts formels

Le nombre de concepts formels est dépendant du nombre d'objets et d'attributs formels. La taille du treillis est au maximum en 2^k, avec $k = min(|objets|, |attributs|)$. La construction des contextes formels à partir de textes selon la méthode syntaxique est généralement très productive en objets et attributs (au minimum un attribut et un objet formel par phrase, sans prendre en compte les répétitions au fil du texte). Cela entraîne que même de petits textes peuvent produire un nombre important de concepts formels. La représentation en treillis devient rapidement incompréhensible : empiriquement au-delà de quelques centaines de nœuds il devient difficile à lire. La figure 5.7 montre une partie du treillis terminologique de la cuisine, qui ne comporte pas moins de 5371 arêtes.

5.3.3 Problème 3 : Utilisation et interprétation du treillis

Une fois que nous disposons d'un treillis de concepts formels se pose la question de savoir *comment s'en servir*. Nous pouvons opter pour une approche comme dans (Cimiano *et al.*, 2005) et construire directement une hiérarchie de concepts à partir de l'ACF, sans présenter de treillis à l'utilisateur, mais nous pensons qu'une approche tout automatique de la COT n'est pas viable. En effet le texte contient des réalisations de connaissances ontologiques exprimées en langage naturel. Or, la langue introduit des biais vis-à-vis des connaissances qu'elle exprime (ellipses, ambiguïtés, métonymie, etc.) qu'il est nécessaire de lever avant de pouvoir envisager une conceptualisation. C'est ce que nous allons vérifier sur un exemple jouet en rapport avec le domaine de l'astronomie. Soit le texte présenté sur la figure 5.8. Nous faisons l'hypothèse pour notre exemple que l'extraction terminologique a repéré les termes *étoile bleue, étoile,*

- Une étoile bleue est très chaude.
- Une étoile brille.
- Les naines rouges sont des étoiles.
- Une galaxie regroupe des étoiles.
- Cette étoile est rouge et âgée.
- Cette étoile est bleue et jeune.
- Une naine blanche était autrefois une étoile.
- La galaxie d'Andromède se rapproche de la Voie Lactée.
- Des étoiles géantes explosent parfois en supernova.
- Une supernova ne dure pas très longtemps.

FIGURE 5.8 : Exemple pour l'interprétation du treillis

naine rouge, galaxie, naine blanche, galaxie d'Andromède, Voie Lactée, étoile géante, supernova. Le contexte formel issu de l'incorporation des termes au sein du processus d'ACF est présenté dans le tableau 5.5, le treillis résultant sur la figure 5.9.

Nous pouvons constater que le treillis terminologique ne comporte pas de relations de subsomption, mais qu'il contient en revanche des erreurs :

- **naine blanche** et **naine rouge** sont regroupées, car elles partagent l'attribut formel *être étoile*. Cela est correct pour une naine rouge, mais incorrect pour une naine blanche : la phrase dit qu'une naine blanche *était autrefois* une étoile. En effet une naine blanche est un objet céleste issu de l'effondrement d'une étoile moyennement massive, ce n'est pas une étoile car son cœur n'est le siège d'aucune réaction thermonucléaire. Cela montre qu'il est important de tenir compte des temps dans la construction du contexte formel pour l'astronomie.
- **étoile** possède les attributs contradictoires *être jeune*, *être âgée*, *être rouge*, *être bleue*. Cela est dû au fait que notre texte parle de *cette étoile*, c'est-à-dire de plusieurs étoiles différentes qui ont des propriétés différentes (au sein d'une constellation par exemple). Le système de construction de contexte formel ne sait pas repérer lorsque le texte parle des étoiles en général (comme dans « Une étoile brille ») ou d'une étoile particulière (mis à part le déterminant « Cette »). On aurait pu avoir la séquence de phrases « La nébuleuse d'Orion abrite de nombreuses étoiles. Les étoiles sont jeunes. » qui aurait abouti au même constat.

Le treillis ne comporte pas de relation de subsomption ; dans cet exemple ceci est normal, car la phase d'ajout des connaissances terminologiques se fait en *remplaçant* l'objet formel simple par le terme associé. Une autre approche serait de *compléter* le contexte formel avec le terme, en faisant l'hypothèse, linguistiquement fondée, que si « étoile jeune » est sujet (ou COD) alors « étoile » l'est aussi. Dans ce cas le contexte formel issu de notre exemple serait celui du tableau 5.6 et son treillis celui de la figure 5.10.

TABLEAU 5.5: Contexte formel terminologique pour exemple d'interprétation

	être chaude	briller	être étoile	regrouper	être rouge	être âgée	se rapprocher de	exploser	durer	regrouper-ABLE	se rapprocher de-ABLE	être bleue	être jeune
étoile		×			×	×				×		×	×
étoile bleue	×												
naine rouge			×										
galaxie				×									
naine blanche			×										
étoile géante								×					
galaxie d'Andromède							×						
Voie Lactée											×		
supernova									×				

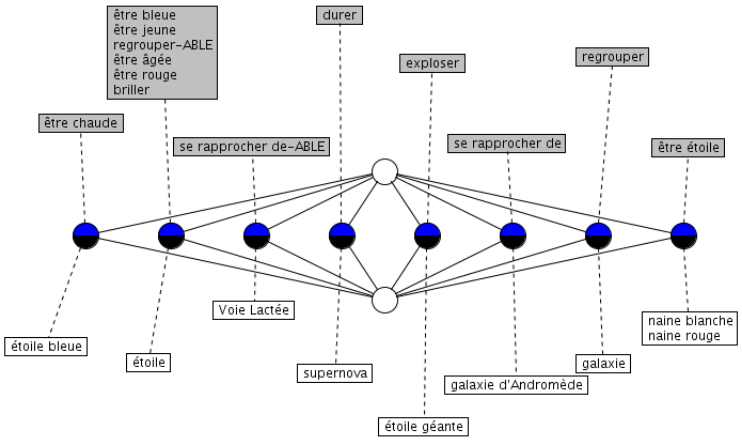

FIGURE 5.9: Treillis terminologique à interpréter en ontologie

TABLEAU 5.6 : Contexte formel terminologique pour exemple d'interprétation (2)

	être chaude	briller	être étoile	regrouper	être rouge	être âgée	se rapprocher de	exploser	durer	regrouper-ABLE	se rapprocher de-ABLE	être bleue	être jeune
étoile	×	×			×	×		×		×		×	×
étoile bleue	×												
naine rouge			×										
galaxie				×			×						
naine blanche			×										
étoile géante								×					
galaxie d'Andromède							×						
Voie Lactée											×		
supernova									×				
naine			×										

L'ajout des propriétés « linguistiques » dans le contexte formel a permis d'augmenter la structuration des concepts dans le treillis, mais les relations de subsomption que l'on aurait envie d'observer *via* les extensions ne sont pas cohérentes, elles semblent inversées.

Du point de vue ontologique, si un concept A est plus général qu'un concept B cela signifie que B possède toutes les caractéristiques de A en plus des siennes. Du point de vue du treillis, en considérant que les attributs formels sont les caractéristiques ontologiques et en associant l'extension avec la dénotation du concept nous obtenons une représentation d'héritage « de haut en bas », comme présenté sur le tableau 5.7 et sur la figure 5.11 (A est plus général que B qui est plus général que C).

TABLEAU 5.7 : L'héritage ontologique dans un contexte formel

Héritage ontologique	Attr 1	Attr 2	Attr 3
A	×		
B	×	×	
C	×	×	×

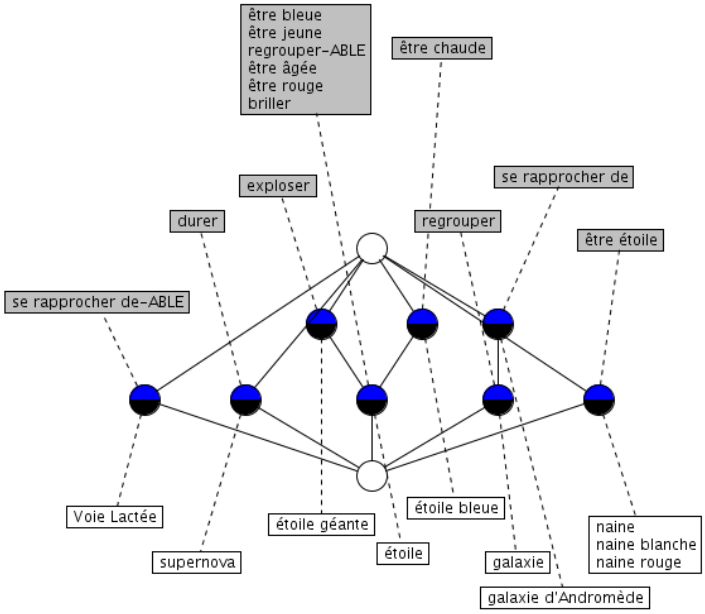

FIGURE 5.10: Treillis terminologique à interpréter en ontologie, variante

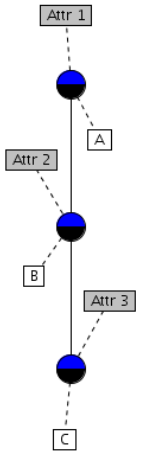

FIGURE 5.11: L'héritage ontologique dans un treillis

Dans le treillis de la figure 5.10 on lit que *galaxie* est plus spécifique que *galaxie d'Andromède* et que *étoile géante* et *étoile bleue* sont plus généraux que *étoile*, ce qui est correct du point de vue des propriétés repérées dans le texte, mais incorrect du point de vue ontologique ; ce phénomène vient probablement de notre désir d'associer l'extension du concept formel, qui est simplement la présentation des occurrences dans le texte, avec sa dénotation ontologique et l'interprétation ontologique des attributs formels issus des contextes syntaxiques. Nous reviendrons plus en détail sur ce problème dans la section 6.3.

5.4 Conclusion

Ce chapitre nous a permis de mettre en avant trois écueils lors de la combinaison d'une approche terminologique avec une méthode de regroupement symbolique : la difficulté, voire l'impossibilité de transformer automatiquement le treillis en ontologie, la sensibilité de la méthode au corpus utilisé et la prolifération du nombre de concepts formels qui rend le treillis illisible. Les chapitres suivants seront consacrés à la présentation de nos propositions pour dépasser ces trois difficultés et aux expériences associées.

Chapitre 6

Propositions

Sommaire

Nous avons constaté dans le chapitre précédent que l'analyse de concepts formels, si elle s'avère une méthode intuitive pour fournir une vue sur le corpus qui va aider l'ontologue dans son travail, souffre néanmoins de défauts qui peuvent s'avérer rédhibitoires : l'explosion combinatoire du nombre de concepts formels qui rend la représentation en treillis inutilisable, la sensibilité à la nature du corpus et enfin l'optimisation de l'utilisabilité du treillis par l'ontologue. Dans ce chapitre nous présentons les pistes que nous avons explorées pour résoudre ces problèmes.

6.1 La dépendance au corpus : la détecter au plus tôt

Nous avons constaté dans la section 5.3.1 que des corpus de taille similaires pouvaient produire une quantité très variable de concepts formels (nous mettons pour l'instant de côté la question, pourtant cruciale, de savoir si la qualité du treillis est liée à la quantité de concepts formels). À quoi est dû ce phénomène ? Peut-on exhiber dans les textes des indicateurs simples à repérer qui permettraient de le prévoir au plus tôt dans la chaîne de traitement, dans le meilleur des cas avant l'analyse syntaxique ?

TABLEAU 6.1 : Taille du vocabulaire

	Droit	Astronomie	Cuisine
Nombre de concepts formels	*513*	*1351*	*1252*
Taille du vocabulaire lemmatisé	2415	3978	2114
Taille du vocabulaire fléchi	3599	5827	3043

6.1.1 Des indicateurs au niveau du vocabulaire ?

Nous avons tout d'abord pensé à observer le vocabulaire des corpus, c'est-à-dire le nombre de mots différents. Notre intuition est qu'un vocabulaire plus riche est susceptible de produire plus de concepts formels. Le tableau 6.1 présente les caractéristiques des corpus au niveau du vocabulaire, à la fois fléchi et lemmatisé. À sa lecture, nous constatons que notre intuition se révèle erronée : le corpus de la cuisine contient un vocabulaire plus petit que celui du droit mais est deux fois plus productif en concepts formels, tandis que celui de l'astronomie, presque deux fois plus riche en vocabulaire, ne produit qu'une petite centaine de concepts supplémentaires.

Si la taille du vocabulaire est un mauvais indicateur de la productivité en concepts formels, peut-être que sa redondance en corpus en est un meilleur, l'ACF ne repose-t-il pas sur les attributs partagés ? Pour mesurer la redondance, nous évaluons le *taux de répétition* du vocabulaire (Muller, 1977), selon la formule :

Définition 13 (Taux de répétition ou redondance d'un vocabulaire)

$$q_1(Voc) = 1 - \frac{|Voc_1|}{|Voc|}$$

dans laquelle Voc représente le vocabulaire du corpus et Voc_1 ses hapax [1]. Plus cette valeur est élevée, plus le corpus est redondant (les cas extrêmes correspondent à : (1) un texte constitué uniquement d'hapax, $q_1 = 0$ et (2) un texte avec un seul mot répété n fois, $q_1 = 1$).

Le tableau 6.2 présente les mesures de redondance pour les vocabulaires fléchis et lemmatisés. Nous constatons que le corpus de l'astronomie est celui qui contient le moins de redondance, tant au niveau fléchi que lemmatisé. Le corpus de la cuisine est sans surprise le plus redondant. Malheureusement, nous n'observons pas de corrélation entre la redondance des vocabulaires et la quantité de contextes formels obtenus.

Ces deux expériences montrent que l'analyse du vocabulaire du corpus, que ce soit au niveau de sa taille ou au niveau de sa redondance, ne permet pas de prédire la quantité de concepts formels produite. Nous nous sommes donc intéressé à d'autres caractéristiques simples du corpus : le nombre d'occurrences des mots, le nombre de phrases et leur taille.

1. Élements de vocabulaire qui n'apparaissent qu'une seule fois.

TABLEAU 6.2 : Redondance du vocabulaire

	Droit	Astronomie	Cuisine
Nombre de concepts formels	*513*	*1351*	*1252*
Redondance du voc. lemmatisé	66,37%	57,11%	66,49%
Redondance du voc. fléchi	61,04%	51,05%	61,38%

TABLEAU 6.3 : Caractéristiques phrastiques des corpus

	Droit	Astronomie	Cuisine
Nombre de concepts formels	*513*	*1351*	*1252*
Nombre de mots	53321	54121	54527
Nombre de phrases	970	2287	4074
Phrase la plus longue (en mots)	824	200	78
Nombre moyen de mots par phrase	55	23	13

6.1.2 Des indicateurs au niveau des phrases ?

Le tableau 6.3 présente des informations sur les phrases des corpus. Nous remarquons que si le nombre de mots est globalement identique pour chaque corpus, le nombre de phrases varie du simple au quadruple (sans surprise, c'est le corpus des recettes de cuisine qui contient le plus de phrases). Mais cet indicateur n'est, lui non plus, pas représentatif : le corpus de l'astronomie contient deux fois moins de phrases que celui de la cuisine, mais produit (un peu) plus de concepts formels. La taille moyenne des phrases et la taille maximale d'une phrase se comportent de façon similaire et ne sont pas non plus des indicateurs fiables. Si l'on regarde uniquement les corpus de droit et de cuisine, cela fonctionne ; cependant le corpus de l'astronomie vient perturber la donne.

6.1.3 Des indicateurs morpho-syntaxiques ?

La dernière étape pour trouver des indicateurs dans le corpus avant d'effectuer son analyse syntaxique consiste à regarder la quantité de noms, de verbes et d'adjectifs [2] au sein du corpus, les autres catégories morpho-syntaxiques n'étant pas prises en compte dans le processus de construction du contexte formel. Nous proposons également de différencier les verbes de modalité des autres verbes. En effet ces verbes qui expriment une possibilité, une obligation, etc. n'apportent pas en eux-mêmes d'information utile à la conceptualisation à propos des caractéristiques intrinsèques de leurs sujets/compléments. La définition d'une liste exhaustive des verbes de modalité en français étant un domaine de recherche en soi (Chu, 2008), nous nous en tenons à

2. ne sont pas utilisés tels quels mais vont se retrouver dans les termes.

Tableau 6.4 : Caractéristiques morpho-syntaxiques des corpus

	Droit	Astronomie	Cuisine
Nombre de concepts formels	*513*	*1351*	*1252*
Nombre de noms communs	13848 (*1117* $\neq$)	13376 (*1537* $\neq$)	18136 (*1018* $\neq$)
Nombre de verbes	7451 (*563* $\neq$)	7216 (*861* $\neq$)	8450 (*553* $\neq$)
Dont modaux	*742 (9,95%)*	*337 (4,67%)*	*134 (1,58 %)*

une liste plus restreinte mais consensuelle : *devoir, pouvoir, falloir, savoir, vouloir*. Le tableau 6.4 présente les résultats de l'analyse morpho-syntaxique pour chaque corpus, en nombre d'occurrences, en précisant entre parenthèses le nombre de noms communs et de verbes différents.

Nous pouvons remarquer que le nombre de noms communs est quasiment identique pour le corpus du droit et celui de l'astronomie. Ce n'est donc pas du tout un indicateur du nombre de concepts formels. Même constatation pour le nombre de verbes. Ceci est décevant, car nous pensions qu'il existait une relation étroite entre le nombre de verbes et le nombre de concepts formels potentiels. L'analyse de la quantité de verbes de modalité semble différencier le corpus de la cuisine et celui du droit, mais comme précédemment le corpus de l'astronomie infirme notre intuition.

Tout au long de ces expériences, nous avons tenté de trouver dans les corpus des indicateurs de surface qui nous permettraient d'estimer rapidement le nombre de concepts formels. Si nous considérons uniquement les corpus du droit et de la cuisine, la taille moyenne des phrases et le taux de verbes de modalité constituent les meilleurs candidats indicateurs. Cependant, au regard du corpus de l'astronomie, ces derniers ne sont plus valides ; tout au plus peut-on émettre l'hypothèse de l'existence d'un seuil à partir duquel le nombre de concepts formels semble chuter. La quantité de verbes de modalité, si elle n'est pas un bon indicateur de la quantité des concepts formels, nous donne en revanche probablement une indication quant à la qualité informative ramenée au domaine de ces derniers. Une caractérisation plus fine des corpus permettrait peut-être de mettre en avant d'autres indicateurs plus fiables.

6.2 L'explosion combinatoire du nombre de concepts formels

Le problème de l'explosion combinatoire inhérente à notre utilisation de l'ACF revient à un problème de sélection / filtrage qui peut être abordé à plusieurs niveaux : lors de la constitution du corpus, lors de la construction du contexte formel, une fois l'intégralité du contexte formel construit (comme dans (Cimiano *et al.*, 2005)), lors de la construction du treillis de concepts (comme dans (Pernelle *et al.*, 2002; Stumme

et al., 2002)). Le problème peut aussi s'envisager à l'aide d'une combinaison de ces filtrages. Nous avons choisi de nous concentrer dans un premier temps sur le filtrage lors de l'étape de la construction du contexte formel. En effet, à cette étape nous gardons encore le lien avec le matériau d'origine, c'est-à-dire les textes. Cela nous permet d'envisager un filtrage plus contextualisé, donc théoriquement plus fiable qu'un filtrage sur le contexte formel intégralement constitué. Ce type de filtrage *a posteriori* sera brièvement étudié à la fin de cette section.

6.2.1 Filtrage au moment de la constitution du corpus

Nous pouvons commencer à prendre en compte l'explosion combinatoire du nombre de concepts formels très tôt dans la chaîne de traitement, au moment de la constitution du corpus. L'idée est de pouvoir rapidement donner une estimation du nombre de concepts formels sans attendre le calcul intégral du contexte formel. Ainsi, au moment de la constitution du corpus l'utilisateur pourrait apprécier rapidement la productivité des documents en concepts formels.

Dans (Mondary et Després, 2009) nous avions étudié la majorité des indicateurs présentés dans la section précédente [3] pour prédire la quantité de concepts formels à partir d'une analyse de surface du corpus. Le nombre de verbes de modalité s'avérait être le plus corrélé avec la quantité de concepts formels produits. Toutefois, les expériences menées sur un nouveau corpus (l'astronomie) ont brouillé nos premières conclusions. Il faudrait pousser plus avant l'analyse du corpus pour exhiber des indices plus précis.

6.2.2 Filtrage lors de la construction du contexte formel

Pour tenter de contenir l'explosion combinatoire inhérente à notre utilisation de l'ACF nous proposons d'intervenir directement sur la sélection des objets formels lors de la construction du contexte formel. L'un de nos objectifs étant de combiner une approche terminologique avec l'ACF, nous avons initialement choisi de remplacer, lorsque c'est possible, la tête d'un candidat objet formel par le terme associé, *tout en conservant les objets formels qui ne correspondent à aucun terme* [4]. Nous avons vu que cette approche fait déjà un peu diminuer le nombre de concepts formels (voir la section 5.3.2). Nous pouvons considérablement augmenter le filtrage en conservant *uniquement* les objets formels que le système a réussi à apparier avec un terme, ce qui nous donne actuellement trois variantes de constructions du contexte formel [5] :

3. Excepté la redondance du vocabulaire et le nombre de verbes / noms communs différents.
4. Il peut s'agir de termes non repérés par l'extracteur ou de mots simples.
5. Dans chaque variante nous avons incorporé une liste de mots vides minimale, modifiable par l'utilisateur, destinée à compenser les erreurs d'étiquetage.

variante usuelle : Inclure dans le contexte formel en tant qu'objets formels les têtes lemmatisées des sujets et des compléments d'objet et en tant qu'attributs formels leurs verbes associés (munis des prépositions le cas échéant), à condition que ces têtes, obtenues après résolution des cas particuliers[6], soient des noms (ce qui supprime le problème de résolution des anaphores pronominales).

variante termlist : Semblable à la variante usuelle, mais tente d'apparier les têtes lemmatisées des sujets et des compléments d'objet avec une liste de termes lemmatisés fournie en entrée par l'utilisateur. Si l'appariement échoue (par exemple si la liste de termes contient « étoile à neutrons, naine rouge » l'appariement va en partie échouer pour la phrase « Les naines jaunes ne deviennent pas des étoiles à neutrons »), inclure seulement la tête (dans notre exemple, « naine »). Les attributs formels sont incorporés tels quels.

variante termlistonly : Inclure dans le contexte uniquement les sujets ou compléments d'objet qui s'apparient avec un terme de la liste fournie en entrée. Les verbes associés aux sujets ou compléments sont inclus dans le contexte comme attributs. Cette variante suppose que les termes intéressants sont connus à l'avance. Ils peuvent être obtenus soit à partir des sorties d'un extracteur de termes (mais dans ce cas peut se poser le problème du filtrage) soit directement à partir des connaissances d'un expert du domaine.

Nous allons maintenant étudier comment la troisième[7] variante de construction du contexte formel fait diminuer le nombre de concepts formels dans le treillis. Le tableau 6.5, à mettre en regard avec les tableaux 5.4 et 5.3, présente l'évaluation quantitative de l'application de cette variante. Les figures 6.1 et 6.2 synthétisent le contenu de ces trois tableaux.

En analysant la figure 6.1 nous constatons que la variante termlistonly fait diminuer dans tous les cas le nombre de concepts formels, mais pas de manière absolument homogène. Nous constatons également à la lecture de cette figure que le nombre d'objets formels augmente avec la variante termlist (ce qui est normal car l'appariement des têtes des syntagmes avec les termes produit plus d'objets formels : « naine rouge et naine jaune » produisent deux objets formels là où la tête « naine » n'en produit qu'un seul), puis diminue légèrement avec la variante termlistonly, sans pour autant revenir au niveau de la variante usuelle. Pour interpréter ces variations, notamment pourquoi la variante termlistonly repère plus d'objets formels que la variante usuelle, il est nécessaire de prendre en compte la méthode utilisée pour constituer la liste de termes, ici l'utilisation d'un extracteur de termes. Une petite liste de termes produite par un expert du domaine aurait sûrement davantage fait diminuer le nombre d'objets formels.

6. Voir la section 5.2.
7. Nous avons déjà mesuré l'impact de la deuxième variante, `termlist`, dans la section 5.3.

TABLEAU 6.5: Caractéristiques du treillis pour la variante `termlistonly`

	Astronomie	Droit	Cuisine
Candidats termes	4881	3124	3607
Objets formels	923	426	607
Attributs formels	603	220	207
Concepts formels	615	234	277
–> % nb de mots	2,49%	0,96%	2,29%
–> dont /extension/ > 1	267	110	129
Hauteur du treillis	4	5	5
Nombre d'arêtes	1241	456	567

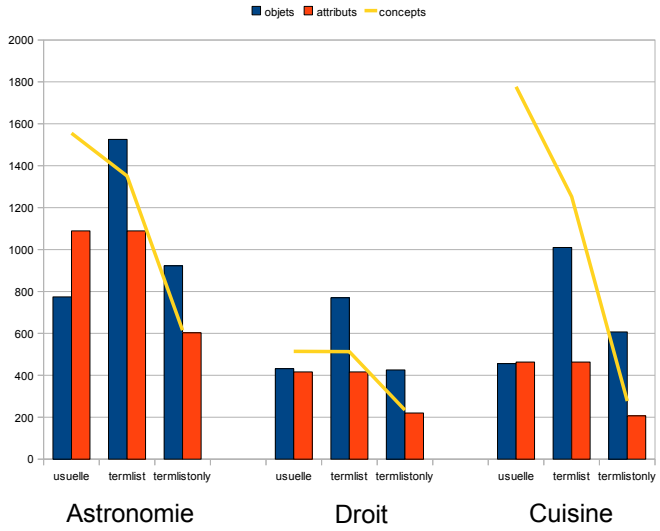

FIGURE 6.1: Concepts, objets et attributs, en occurrences

TABLEAU 6.6: Taux de diminution du nombre de concepts formels dans la variante `termlistonly`, par rapport aux deux autres variantes

	Astronomie	Droit	Cuisine
Candidats termes	4881	3124	3607
% termlist	54,5%	54,4%	77,9%
% usuelle	60,4%	54,4%	84,4%

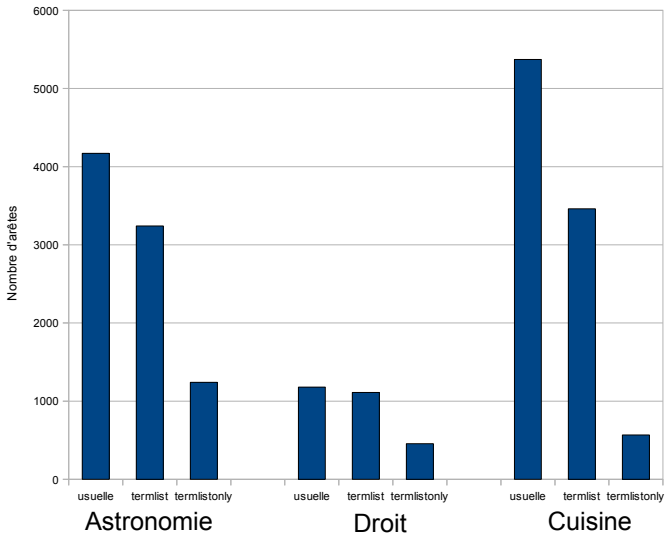

FIGURE 6.2 : Nombre d'arêtes des treillis

TABLEAU 6.7 : Quantité de sujets et de COD dans chaque corpus

	Sujets	COD	Sujets+COD
Astronomie	3505	2068	5573
Droit	1814	2008	3822
Cuisine	1245	4206	5451

Le tableau 6.6 présente, pour chaque corpus, le taux de diminution du nombre de concepts de la variante termlistonly, par rapport aux variantes usuelles et termlist. Nous pouvons y constater que l'incidence de la variante termlistonly est identique pour les corpus de l'astronomie et du droit. Le corpus de la cuisine est beaucoup plus sensible à cette variante. Cela indique soit que les termes de la cuisine permettent plus de regroupements que ceux des autres corpus, c'est-à-dire que des termes différents apparaissent souvent avec des verbes identiques, soit qu'ils apparaissent moins souvent en tant que sujet ou COD. Comme les phrases sont typiquement plus courtes que celles des deux autres corpus, la deuxième hypothèse paraît peu plausible. Le tableau 6.7 indique la quantité de sujets et de COD repérés par Syntex pour chaque corpus. On peut y voir que le corpus de cuisine compte quasiment le même nombre de sujets+COD que celui de l'astronomie : ceci vient appuyer notre première hypothèse.

6.2.3 Filtrage sur le contexte formel construit

Une fois le contexte formel totalement construit à l'aide de l'une des variantes présentées précédemment, il est possible de le filtrer pour en réduire la taille selon des critères

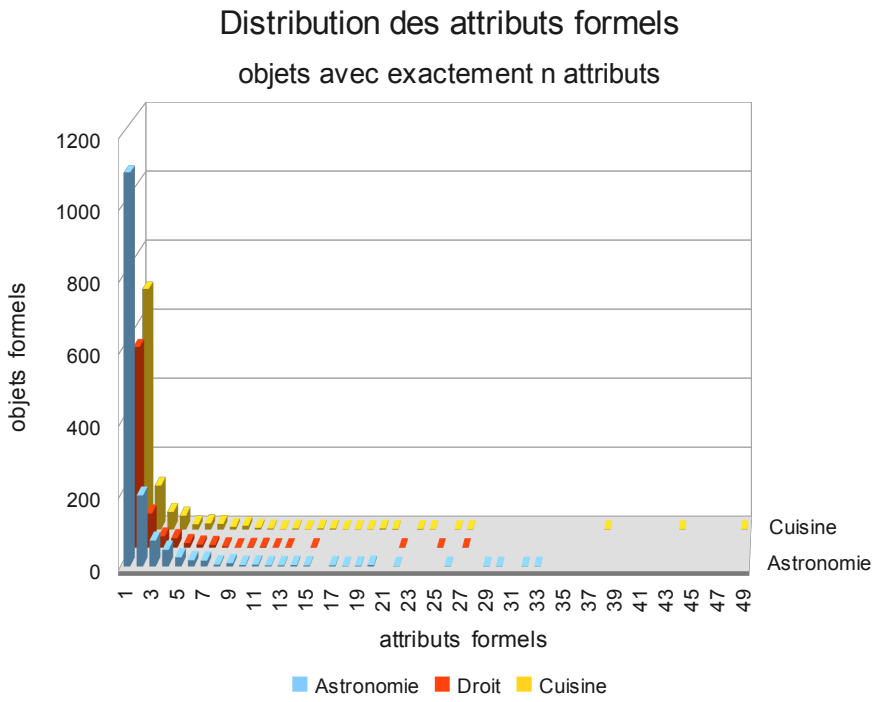

FIGURE 6.3 : Distribution des attributs formels, méthode termlist

numériques ; le problème est alors de fixer les seuils. Nous envisageons deux stratégies de filtrage sur le contexte formel : la première, sur les objets formels, ne retiendra que ceux qui possèdent entre i et j attributs. La seconde opère quant à elle sur les couples $(objet, attribut)$, en fonction de leur probabilité conditionnelle.

Filtrage des objets formels en fonction de leur nombre d'attributs

Ce filtrage permet de réduire le nombre d'objets formels. Cela signifie que son utilité dépend de la variante de construction utilisée : dans le cas de termlistonly les objets formels sont déjà supposés pertinents, il n'est donc *a priori* pas nécessaire de les filtrer à nouveau.

L'intuition sous-jacente est que les objets formels qui possèdent trop peu d'attributs apparaissent peu dans le corpus [8] et sont moins « importants » par rapport au domaine. Ceux qui en possèdent beaucoup apparaissent avec de nombreux verbes, cela semble indiquer que ce sont soit des cas de polysémie, soit des éléments complexes à décrire. La figure 6.3 présente pour chaque corpus la distribution des attributs par rapport aux objets formels, pour la variante termlist.

8. Cela n'est pas la même chose que le nombre d'occurrences des objets formels en corpus : nous parlons ici d'objets formels qui apparaissent avec des attributs formels *différents*.

Nous pouvons constater sur cette figure que la majorité des objets formels possèdent seulement un attribut [9] et qu'il existe un « trou » plus ou moins prononcé au-delà d'un certain nombre d'attributs. Les objets formels situés au-delà sont soit polysémiques, soit complexes à décrire, soit liés à des attributs formels erronés.

Pour le corpus de cuisine, la frontière semble se situer aux alentours d'une trentaine d'attributs. L'objet formel `œuf` possède 27 attributs ; l'objet suivant est `pâte` avec 37 attributs, suivi de `fonds` avec 43 et `artichaut` avec 49 attributs. Cela traduit à la fois qu'il est possible de préparer l'artichaut de nombreuses façons, que le corpus comporte de nombreuses recettes à base d'artichaut. Le trou entre `œuf` et `artichaut` ne semble pas interprétable, c'est probablement un artefact statistique lié à la petite taille du corpus.

Pour le corpus de droit, l'objet formel ayant le plus d'attributs est `membre` avec 27 attributs. Cet objet formel est clairement à considérer comme du bruit par rapport au domaine [10]. La frontière semble se situer entre 16 et 23 attributs. À 16 nous avons `travailleur` qui est pertinent pour le domaine tandis qu'avec 23 attributs, `personne` se comporte comme du bruit [11]. Dans le cas de ce domaine, la distinction entre objets formels pertinents et bruités à l'aide de la frontière semble fonctionner.

Enfin pour le corpus de l'astronomie, la frontière est beaucoup moins marquée, peut-être entre 22 attributs et 26 attributs. L'objet formel ayant le plus d'attributs est `objet` avec 33 attributs, c'est clairement du bruit [12] dans un corpus sur les objets célestes. Il est suivi de près par `galaxie` qui possède 32 attributs et n'est lui pas du bruit. De même, les objets situés après la frontière présumée de 22 attributs sont `planète`, `soleil`, `masse`, tous pertinents pour le domaine.

L'analyse des trois corpus nous conduit à invalider notre hypothèse de séparation des objets formels utiles du bruit à l'aide de la frontière dans la répartition des attributs formels. L'idée qui consisterait à éliminer systématiquement l'objet formel ayant le plus d'attributs serait sans conséquence pour les corpus de l'astronomie et du droit, mais inadaptée pour la cuisine.

Si nous analysons les objets formels ayant un seul attribut, nous constatons la présence de bruit dû aux erreurs d'étiquetage. Il existe néanmoins des éléments intéressants sous la forme de termes polylexicaux longs, par exemple `gaz du rémanent de supernova` ou `vitesse de condensation du nuage moléculaire` pour l'astronomie, `vinaigrette à la ciboulette hachée` pour la cuisine et `engin de terrassement et de manutention` pour le droit.

9. Qui peut être partagé avec d'autres objets formels.
10. Il apparait le plus souvent dans « Membre de l'Organisation du Travail ».
11. Il est souvent employé pour désigner des termes plus complexes comme « personnes morales privées ». Nous reviendrons sur ce problème dans ce mémoire.
12. Plus précisément, c'est un concept tellement général dans le domaine des objets célestes qu'il n'apporte plus d'information.

Filtrage par taux de dispersion des attributs

Le filtrage des objets formels a montré ses limites en fonction du corpus utilisé ; il est de plus inadapté par nature à la variante de construction termlistonly. Nous proposons de filtrer le contexte formel construit en attribuant un score à chaque attribut formel.

Le score attribué va représenter le *taux de dispersion* d'un attribut dans un contexte formel, c'est-à-dire indiquer si l'attribut se retrouve avec un seul objet ou est dispersé au travers du contexte formel. L'intuition sous-jacente est qu'un attribut formel totalement spécifique à un objet ne sera pas susceptible de produire un concept formel regroupant [13], il serait plutôt un attribut propre de l'objet, tandis qu'un attribut trop dispersé risque soit de provenir d'une erreur d'analyse syntaxique, soit être trivial pour le domaine. Pour chaque attribut *attr* d'un contexte formel, nous définissons son taux de dispersion de la manière suivante :

Définition 14 (Taux de dispersion)

$$dispersion(attr) = 1 - 1/f(attr) \tag{6.1}$$

avec $f(attr)$ le nombre d'objets formels qui possèdent l'attribut attr.

Le taux de dispersion vaut 0 si l'attribut n'est porté que par un seul objet (cela se traduira par un élément de l'intension propre d'un concept formel) et tend vers 1 si l'attribut est présent dans tous les objets (cela ajoutera une intension au concept formel TOP du treillis).

En guise d'exemple prenons le contexte formel du tableau 6.8, extrait du contexte formel des recettes de cuisine construit à l'aide de la variante `termlist`. Le taux de dispersion de chaque attribut est présenté dans le tableau 6.9.

Nous proposons maintenant d'étudier la dispersion des attributs sur les contextes formels construits à partir de nos trois corpus à l'aide de la variante `termlist`. Nous analysons plus en détails les attributs dont le taux de dispersion est situé aux extrêmes.

Pour le corpus de l'astronomie, les dix attributs ayant les plus forts taux de dispersion sont, par ordre décroissant `posséder-ABLE`, `permettre`, `former-ABLE`, `posséder`, `devoir`, `atteindre-ABLE`, `produire-ABLE`, `observer-ABLE`, `former`, `exister-ABLE`. Mis à part `observer-ABLE`, ces attributs sont cependant à considérer comme du bruit. Il existe 576 attributs avec un taux de dispersion nul. Ces attributs ne permettent pas d'engendrer des concepts formels car ils ne sont, par définition, partagés par aucun objet formel. En revanche, nous avons l'intuition qu'ils pourraient parfois évoquer des notions utiles pour l'ontologue.

13. qui contient plus d'un objet formel comme extension.

TABLEAU 6.8: Contexte formel de la cuisine

	filtrer-ABLE	verser-ABLE	retirer-ABLE	nettoyer-ABLE	demouler-ABLE	casser-ABLE	déposer-ABLE	effeuiller-ABLE	couvrir-ABLE	éplucher-ABLE
bouillon de légumes	×	×	×							
casserole		×	×	×					×	
champignon				×		×	×			
charlotte					×					
chocolat		×			×	×				
chou				×				×		×
cresson				×			×	×		

TABLEAU 6.9: Taux de dispersion des attributs du contexte 6.8

dispersion(filtrer-ABLE)	0
dispersion(verser-ABLE)	0,67
dispersion(retirer-ABLE)	0,5
dispersion(nettoyer-ABLE)	0,75
dispersion(démouler-ABLE)	0,5
dispersion(casser-ABLE)	0,5
dispersion(déposer-ABLE)	0,5
dispersion(effeuiller-ABLE)	0,5
dispersion(couvrir-ABLE)	0
dispersion(éplucher-ABLE)	0

TABLEAU 6.10 : Quantité d'attributs de dispersion nulle par corpus et par variante

	usuelle	termlist	termlistonly
Astronomie	576 (53 %)	576 (53 %)	355 (59 %)
Droit	155 (37,2 %)	155 (37,2 %)	97 (44 %)
Cuisine	192 (41,5 %)	192 (41,5 %)	109 (52,6 %)

Pour le corpus de droit, les dix attributs les plus dispersés sont `devoir`, `porter-ABLE`, `ratifier-ABLE`, `enregistrer-ABLE`, `assurer-ABLE`, `entrer`, `prendre-ABLE`, `concerner-ABLE`, `lier`, `communiquer` ; aucun n'est très spécifique, ils semblent plutôt relever du sens commun pour le domaine (le vocabulaire du domaine juridique est difficile à cerner, car de nombreux termes de la langue générale acquièrent un sens particulier, voir (Lame, 2002), pages 93–99). Il existe 155 attributs spécifiques.

Pour le corpus de la cuisine, les dix attributs les moins spécifiques sont `ajouter-ABLE`, `verser-ABLE`, `mélanger-ABLE`, `retirer-ABLE`, `mettre-ABLE`, `cuire-ABLE`, `couperABLE`, `préparer-ABLE`, `incorporer-ABLE`, `joindre-ABLE`. Contrairement aux deux autres corpus, ces attributs évoquent des caractéristiques intéressantes pour le domaine, par exemple `verser-ABLE` évoque un concept de préparation non solide. 192 attributs sont spécifiques.

Le tableau 6.10 récapitule la quantité et la proportion des attributs formels de dispersion nulle pour nos trois corpus et nos trois variantes de construction du contexte formel. Nous pouvons constater que comme le passage de la variante usuelle à la variante termlist ne modifie pas les attributs formels, la proportion d'attributs de dispersion nulle reste elle-aussi inchangée. Le corpus de l'astronomie contient une plus grande proportion d'attributs non dispersés que les deux autres, ceci est probablement dû à sa nature encyclopédique. Le passage à la variante termlistonly entraîne une diminution du nombre d'attributs de dispersion nulle qui s'explique par la diminution globale du nombre d'attributs. En revanche, leur proportion augmente : près de 60% des attributs formels du corpus de l'astronomie ont une dispersion nulle !

Nous proposons de présenter à l'utilisateur les attributs formels munis de leur taux de dispersion afin de lui permettre de fixer rapidement des seuils minimum et maximum. Ce filtrage est à manier avec précaution : s'il est tentant d'éliminer automatiquement tous les attributs spécifiques, ceux-ci peuvent parfois contenir des indications utiles pour la conceptualisation, mais à condition d'être compréhensibles par le lecteur (nous allons étudier ce point dans la section suivante). Il est également important de noter que la suppression des attributs de dispersion nulle entraîne pour la méthode termlistonly la perte d'une grande quantité d'information qui pourrait se révéler utile dans la caractérisation des objets représentés dans le treillis. Les attributs les plus dispersés sont soit des artefacts de langage, soit des connaissances triviales du domaine ; dans le corpus de la cuisine, ces connaissances semblent pouvoir malgré tout être utiles à l'ontologue.

6.3 L'interprétation du treillis

Dans cette section nous nous plaçons dans le cas où nous possédons un treillis de taille raisonnable, dans le sens qu'il soit lisible par l'utilisateur. Nous avons vu dans la section 5.3.3 que si l'on applique systématiquement le critère, pourtant valide d'un point de vue linguistique, de propagation de l'attribut formel sur la tête d'un objet formel composé nous obtenons des aberrations sous forme de subsomptions inversées au moment de la lecture du treillis. Cela signifie que la construction du contexte formel est une étape sensible dans le processus de COT. Un treillis doit représenter le plus fidèlement possible les informations contenues dans les textes. Pour cela, nous proposons tout d'abord d'améliorer la qualité des attributs formels présentés, puis nous nous pencherons sur le problème de la subsomption inversée.

6.3.1 Des attributs formels plus précis

Jusqu'à maintenant, nous avons utilisé les verbes simples comme attributs formels. Les attributs formels sont censés être les caractéristiques des objets formels, mais plus celles-ci sont précises, mieux l'objet est caractérisé. Par précision nous entendons à la fois des attributs formels évocateurs pour l'utilisateur et corrects par rapport à ce que le texte veut exprimer.

Attributs formels étendus et élargis

Pour augmenter la précision, dans le sens « évocateur pour l'utilisateur », des attributs formels, nous proposons de prendre en compte non plus seulement le verbe [14], mais le syntagme verbal étendu. Par exemple, l'objet formel `dentelle du cygne` se voit muni de l'attribut `être exemple`. Cela n'est pas très explicite, sinon dans l'évocation d'un cas particulier. En reconstituant le syntagme dont le noyau est `exemple`, nous obtenons l'attribut formel `être exemple de rémanent de supernova`, qui est beaucoup plus explicite. L'objet `galaxie` possède l'attribut `différer`. Que peut-il bien signifier ? Son expansion en syntagme nous apprend que l'attribut est `différer par taille`, qui est beaucoup plus intéressant pour la modélisation.

Définition 15 (Attribut formel étendu) *Un attribut formel est dit* étendu *s'il est constitué du verbe et d'un complément prépositionnel associé.*

Définition 16 (Attribut formel élargi) *Un attribut formel est dit* élargi *s'il est constitué du verbe et de son complément d'objet direct sous forme étendue.*

14. Une prise en compte intelligente du verbe est déjà effectuée, par exemple dans le cas d'un attribut du sujet c'est le verbe + l'adjectif qui est utilisé.

Nous faisons la distinction entre les attributs formels *étendus* et les attributs formels *élargis*. Ces derniers sont caractérisés par le fait que leur partie étendue est susceptible d'apparaître en tant qu'objet formel dans le contexte.

L'expansion en syntagme des attributs est intéressante pour améliorer leur précision et permettre de détecter plus aisément les attributs inutiles, mais elle a un défaut majeur : elle diminue les possibilités de regroupement des objets formels. En effet un attribut formel étendu aura moins de chance de se retrouver avec des objets formels différents qu'un attribut formel simple. Une solution consiste à saturer le contexte formel avec les attributs simples, c'est-à-dire à ajouter systématiquement la forme simple de l'attribut formel à l'objet formel. Nous proposons cependant de différencier les attributs simples usuels des attributs simples de saturation, afin de permettre à l'utilisateur de se représenter les phrases dont est issu le concept formel (un exemple est donné dans le tableau 6.13, avec les attributs `conduire` et `conduire (SAT)`). En effet pour permettre une bonne compréhension, les attributs d'un concept formel doivent se référer au moins à une phrase du corpus ; sans différenciation, ce principe n'est plus respecté.

Deux niveaux d'attributs de saturation sont proposés :

Définition 17 (Attribut de saturation de niveau 1) *Un attribut de* saturation de niveau 1 *est un attribut formel issu de la simplification en une passe d'un attribut formel étendu ou élargi. Dans le cas d'un attribut formel étendu, il est constitué d'un verbe et d'une préposition. Dans le cas d'un attribut formel élargi, il est constitué uniquement du verbe. Dans tous les cas, l'attribut est suivi du suffixe* `(SAT)`.

Définition 18 (Attribut de saturation de niveau 2) *Un attribut de* saturation de niveau 2 *est un attribut formel issu de la simplification en deux passes d'un attribut formel étendu (il n'existe pas d'attribut de saturation de niveau 2 pour les attributs élargis). Il est uniquement constitué d'un verbe, suivi du suffixe* `(SAT2)`. *Un attribut formel de saturation de niveau 2 est potentiellement plus regroupant qu'un attribut de saturation de niveau 1, mais aussi potentiellement plus polysémique.*

À titre d'illustration prenons le texte présenté sur la figure 6.4. En appliquant la méthode de construction usuelle du contexte formel, nous obtenons le contexte du tableau 6.11 et le treillis de la figure 6.5.

Nous constatons que ce treillis n'est ni très structuré, ni très informatif, alors que le texte initial est plus riche. Remarquons toute suite l'erreur d'analyse syntaxique sur la phrase « Mario conduit la nuit » : Syntex considère nuit comme un complément d'objet direct, cela est un exemple des inévitables erreurs d'analyses liées aux outils. Si nous incorporons maintenant les attributs *étendus*, nous obtenons le contexte présenté dans le tableau 6.12 et le treillis de la figure 6.6. Cette fois-ci, le treillis obtenu est plus conséquent, mais manque de structure par rapport au texte.

– Jean conduit sur la route.
– Pierre conduit avec prudence.
– Henri conduit sur la chaussée.
– Albert conduit avec joie.
– Le capitaine conduit ses troupes.
– Mario conduit la nuit.
– Julien conduit avec son moniteur.

FIGURE 6.4 : Illustration des attributs formels étendus, élargis et de saturation

TABLEAU 6.11 : Contexte formel issu de la figure 6.4, attributs "normaux"

	conduire	conduire-ABLE
Albert	×	
Henri	×	
Jean	×	
Julien	×	
Mario	×	
Pierre	×	
capitaine	×	
nuit		×
troupe		×

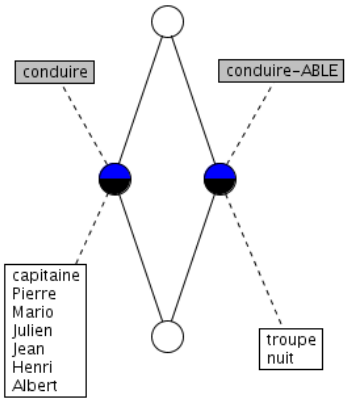

FIGURE 6.5 : Treillis issu du contexte formel 6.11

TABLEAU 6.12: Contexte formel issu de la figure 6.4, attributs étendus

	conduire	conduire avec joie	conduire avec moniteur	conduire avec prudence	conduire sur chaussée	conduire sur route	conduire-ABLE
Albert		×					
Henri					×		
Jean						×	
Julien			×				
Mario	×						
Pierre				×			
capitaine	×						
nuit							×
troupe							×

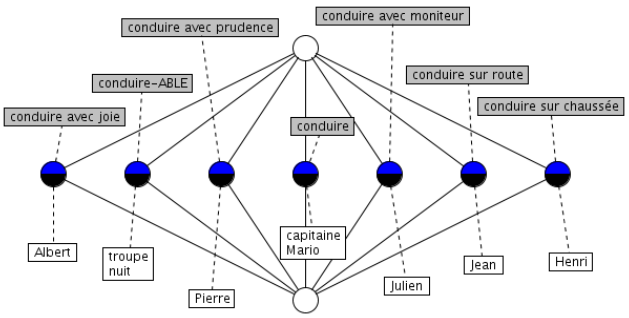

FIGURE 6.6: Treillis issu du contexte formel 6.12

TABLEAU 6.13: Contexte formel illustration des attributs étendus et de saturation

	conduire	conduire (SAT2)	conduire avec (SAT)	conduire avec joie	conduire avec moniteur	conduire avec prudence	conduire sur (SAT)	conduire sur chaussée	conduire sur route	conduire-ABLE
Albert		×	×	×						
Henri		×					×	×		
Jean		×					×		×	
Julien		×	×		×					
Mario	×									
Pierre		×	×			×				
capitaine	×									
nuit										×
troupe										×

Les deux treillis précédents sont corrects du point de vue du texte. Le treillis sur la figure 6.6 apporte plus d'informations, mais elles sont différentes de celles du treillis de la figure 6.5 : à aucun moment il n'est mentionné que tous les personnages du texte conduisent. Pour traduire ce fait en conservant les informations plus précises, le contexte du tableau 6.13 et son treillis (figure 6.7) utilisent conjointement des attributs étendus et des attributs de saturation.

Nous constatons cette fois-ci que presque tous les objets formels sont regroupés sous le concept ayant pour intension l'attribut de saturation `conduire (SAT2)`. Nous remarquons également la présence simultanée des attributs `conduire (SAT2)` et `conduire`. Il reste malgré tout un problème vis-à-vis de ce dernier : il regroupe Mario et capitaine alors que l'un conduit une voiture, l'autre conduit des troupes. Il serait intéressant de montrer à l'utilisateur qu'il existe une différence : les attributs formels élargis vont nous y aider.

Le contexte formel présenté dans le tableau 6.14 et son treillis (figure 6.8) présentent une illustration de l'intérêt des attributs formels élargis. Nous remarquons que par rapport au treillis précédent, les objets formels `Mario` et `capitaine` ne sont plus directement regroupés, mais seulement par un attribut de saturation de niveau 1, `conduire (SAT)`.

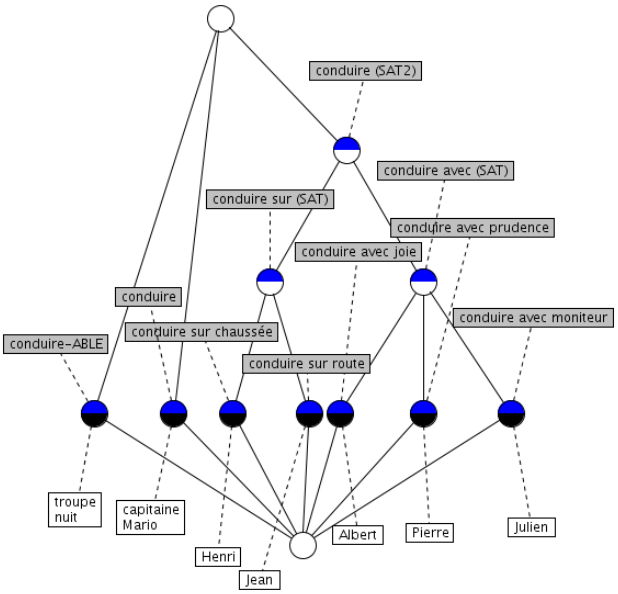

FIGURE 6.7 : Treillis issu du contexte formel 6.13

Nous notons la présence simultanée de l'attribut formel `conduire` sous forme d'attribut de saturation de niveau 1 et de niveau 2. Ceci traduit qu'il est une fois issu d'un attribut élargi (`conduire (SAT)` [15]), l'autre fois d'un attribut formel étendu (`conduire (SAT2)`).

Nous avons montré au travers de ces exemples l'intérêt que peut avoir la prise en compte des attributs étendus et élargis dans l'interprétation du treillis par l'utilisateur. Les attributs de saturation permettent de traduire les phénomènes linguistiques tout en laissant à l'utilisateur la possibilité de les repérer. En utilisant conjointement les attributs étendus, élargis et de saturation, nous proposons un treillis plus fidèle, plus proche du contenu textuel. Nous pensons donc avoir augmenté son interprétabilité. Mais d'autres pistes s'offrent à nous pour continuer à s'approcher de ce qui est exprimé dans le texte, c'est ce que nous allons voir maintenant.

Des attributs formels plus justes

Un contexte formel peut-être vu comme une représentation simplifiée du contenu sémantique du texte. Les attributs formels sont des caractéristiques, des propriétés des objets formels, issus des phrases. Il arrive parfois que les phrases soient négatives, dans

15. D'une manière générale, les attributs suffixés de (SAT) qui contiennent seulement un verbe proviennent d'un attribut formel élargi, c'est-à-dire de la prise en compte d'un COD.

TABLEAU 6.14 : Contexte formel issu du texte de la figure 6.4, attributs étendus, élargis et saturés

	conduire (SAT)	conduire (SAT2)	conduire avec (SAT)	conduire avec joie	conduire avec moniteur	conduire avec prudence	conduire nuit	conduire sur (SAT)	conduire sur chaussée	conduire sur route	conduire troupe	conduire-ABLE
Albert		×	×	×								
Henri		×						×	×			
Jean		×						×		×		
Julien		×	×		×							
Mario	×						×					
Pierre		×	×			×						
capitaine	×										×	
nuit												×
troupe												×

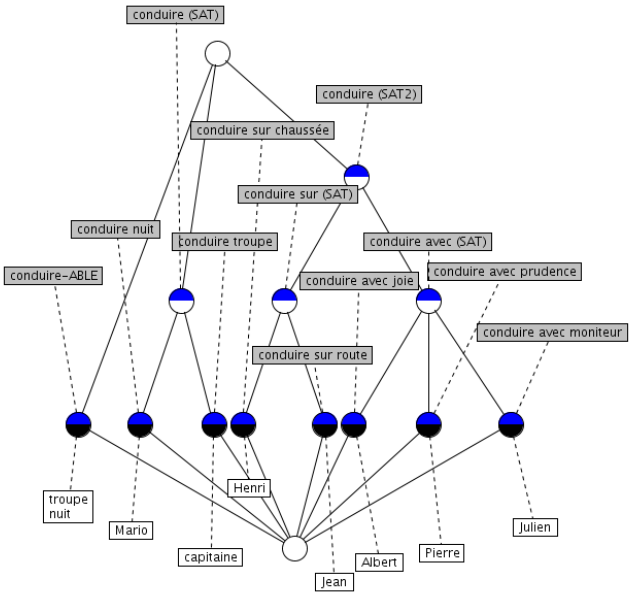

FIGURE 6.8 : Treillis issu du contexte formel 6.14

le sens où elles énoncent une caractéristique négative sur l'objet formel. Par exemple, dans le corpus de l'astronomie nous retrouvons la phrase « Les astéroïdes ne sont pas les satellites d'une planète ». En construisant le treillis sans prendre en compte le caractère négatif de cette information, nous présenterions à l'utilisateur le concept formel *(astéroïde ; être satellite de planète)*, ce qui est *faux* !

Nous proposons deux alternatives à l'utilisateur. La première consiste à systématiquement supprimer du contexte formel tous les attributs formels négatifs, la seconde consiste à les suffixer pour permettre à l'utilisateur de les repérer facilement. Le repérage systématique de la négation est un problème loin d'être trivial. Nous proposons une approche simple reposant sur une liste de marqueurs de négation : l'idée est que pour chaque verbe, nous recherchons dans les dépendances syntaxiques s'il existe un ou des adverbes négatifs reliés à ce verbe. Nous considérons les éléments suivants comme marqueurs de négation : « ne, pas, non, point, nullement, aucunement, aucun, nul, rien, jamais, pas un, personne, sans ».

Quelques cas de négation que notre système arrive à repérer correctement :

1. Les astéroïdes ne sont pas les satellites d'une planète. (phrase marquée comme négative par le système)
2. Jean ne conduit pas. (phrase marquée comme négative)
3. Pierre conduit le moins possible. (phrase *non* marquée comme négative)
4. Paul ne conduit jamais. (phrase marquée comme négative)

Certaines formes de négation sont plus problématiques d'un point de vue sémantique que d'autres, par exemple la phrase 3 est-elle à considérer comme totalement négative ? Cette notion de négation partielle se retrouve dans la phrase « En astronomie, les naines rouges sont les étoiles les moins massives ; en deçà, ce sont les naines brunes, qui ne sont pas vraiment des étoiles. ». Pour la première partie de la phrase, l'objet formel `naine rouge` sera muni de l'attribut étendu `être étoile massive`, ce qui est *faux* ! Pourtant, on ne peut pas considérer que la phrase dit qu'une naine rouge n'est pas une étoile massive, elle énonce seulement qu'elle est l'étoile la moins massive. Notre système va donc conserver l'attribut `être étoile massive`. La seconde partie de la phrase énonce une négation partielle qui sera détectée comme totale par notre système.

À notre connaissance, aucun système de l'état de l'art ne tient compte de la négation pour aider l'utilisateur dans sa tâche de COT. Nous pensons pourtant qu'elle est un élément à ne pas négliger, même s'il ne faut pas oublier que sa détection, pas totalement fiable, est susceptible d'ajouter du bruit au treillis.

6.3.2 Des objets formels étendus

Dans notre optique d'amélioration de l'interprétabilité du treillis, nous nous penchons maintenant sur les objets formels. Nous avons déjà proposé d'incorporer les termes de deux manières différentes, la première en les ajoutant préférentiellement au contexte formel (variante `termlist`), la seconde en ne conservant que ces derniers (variante `termlistonly`).

Nous envisageons maintenant une troisième voie. Si nous ne disposons pas d'une liste de candidats termes, mais que nous souhaitons tout de même proposer des objets formels plus précis que les mots, nous pouvons choisir *d'étendre* les têtes des candidats objets formels, de manière similaire à la construction des attributs étendus. Ces objets formels étendus ne sont pas tous des termes, car aucun filtrage n'est effectué sur eux et ils peuvent par conséquent contenir des éléments qui ne sont pas en rapport avec le domaine.

Définition 19 (Objet formel étendu) *Un objet formel est dit* étendu *s'il est composé d'un syntagme nominal maximal relativement au corpus.*

Les objets formels étendus peuvent être utilisés de trois façons lors de la construction du contexte formel. La première consiste à ne pas incorporer une liste de termes, mais à construire tous les objets formels sous leur forme étendue. Cette méthode n'a pas grand intérêt dans notre cas, car notre objectif est de combiner une approche terminologique avec l'ACF, mais elle permet de se passer d'un extracteur de termes. La deuxième idée consiste à incorporer une liste de termes selon la procédure habituelle et à étendre les objets formels résultants ; le résultat sera normalement identique à la première variante.

La dernière façon d'utiliser les objets formels étendus consiste à les incorporer dans la variante `termlistonly`. Cette fois-ci, le contexte formel ne contiendra pas l'intégralité des sujets / COD du texte, mais seulement les termes sous forme étendue. Quel est l'intérêt d'étendre les termes si ce sont déjà des termes maximaux ? L'intérêt existe justement parce qu'ils ne sont pas nécessairement maximaux, c'est-à-dire que l'utilisateur moyennement expert du domaine peut fabriquer, à la main, une petite liste de termes initiaux que le système va étendre avec les occurrences trouvées dans le texte. Par exemple, si l'utilisateur décide de se focaliser sur le terme *galaxie*, le système va repérer *galaxie lenticulaire*, *galaxie spirale*, etc. C'est en quelque sorte une extraction terminologique guidée par l'utilisateur et projetée sur le texte.

Cette méthode (que nous appelons `termlistonly+expand`) permet également d'envisager une construction du contexte formel en plusieurs passes. L'idée est que l'utilisateur fournit une petite liste de termes simples initiaux, le système la transforme en objets formels étendus qu'il caractérise avec des attributs formels étendus et élargis, éventuellement de saturation. Or chaque attribut formel étendu ou élargi fait apparaître un

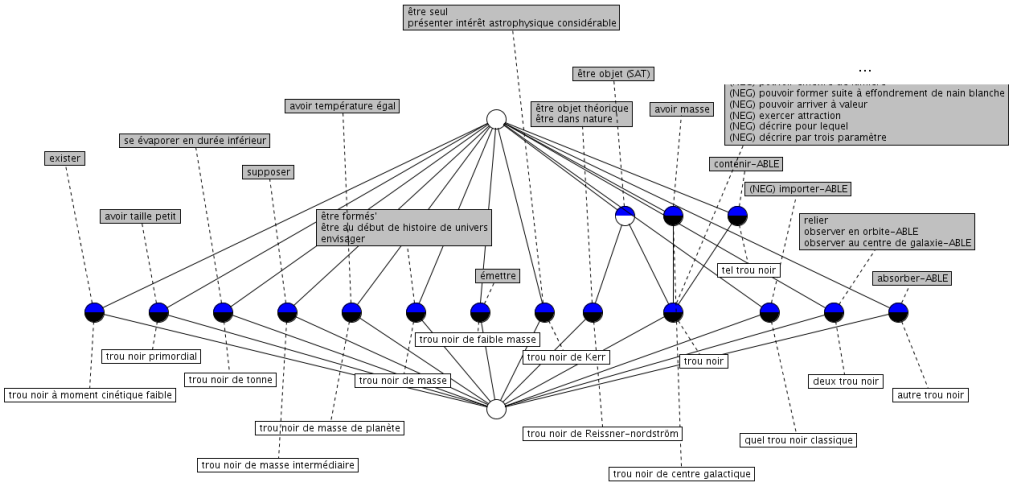

FIGURE 6.9 : Treillis sur l'astronomie, variante `termlistonly+expand`, sur le terme *trou noir*

nom commun lié au verbe (voir les définitions de la section 6.3.1). Ce nom commun ne pourrait-il pas être incorporé dans la liste des objets formels [16] ? À chaque nouvelle passe nous proposons de considérer comme liste de termes tous les noms communs retrouvés dans les attributs formels de la passe précédente.

À titre d'exemple, nous allons présenter un contexte formel issu du corpus de l'astronomie et construit selon la variante `termlistonly+expand` avec le terme `trou noir` comme entrée. Le treillis est présenté sur la figure 6.9, l'intension du concept formel d'extension `trou noir` est reportée sur la figure 6.10. Nous constatons l'impact de l'extension des objets formels, et aussi la différence qu'il peut y avoir entre cette méthode et l'incorporation de véritables termes du domaine : beaucoup de bruit est généré (« trou noir de tonne, quel trou noir classique, autre trou noir, deux trous noirs, trou noir de masse ») mais de véritables termes apparaissent (« trou noir de Reissner-nordström, trou noir de Kerr, etc. »). Nous constatons aussi sur cette figure la différence déjà évoquée dans la section 5.3.3 entre les caractéristiques génériques (`posséder entropie`) et particulières (`se trouver dans amas de sept étoiles`).

Si nous envisageons une construction du treillis en plusieurs passes, le système trouve 139 termes à incorporer dans le contexte formel, notamment *galaxie, masse, pulsar, orbite, etc.*. Le treillis résultant contient 322 concepts et nous retombons alors dans un problème de taille des données à traiter.

16. Il le serait systématiquement dans une variante de construction autre que `termlistonly`.

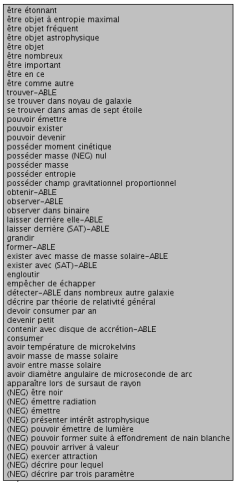

FIGURE 6.10 : Intension du concept formel `trou noir` du treillis 6.9

6.3.3 Un soupçon d'inférence

Jusqu'à maintenant, les propositions que nous avons présentées dans le but d'améliorer la précision des attributs et des objets formels étaient en fait des améliorations de la procédure de simplification du texte, c'est-à-dire que le treillis était le reflet du texte, approché avec plus ou moins de réussite, mais ne contenant pas d'éléments absents du corpus. Deux intuitions nous poussent néanmoins à proposer dans le treillis des éléments qui ne sont pas explicitement dans les textes : l'envie de saturer les objets formels comme nous saturons les attributs formels, et l'envie d'incorporer les relations de subsomption directement dans le treillis.

La saturation des objets formels

Nous proposons déjà dans la section 6.3.1 de saturer les attributs formels. Pourquoi ne pas envisager la même chose pour les objets formels ? Nous avons évoqué de manière implicite cette possibilité dans la section 5.3.3 en parlant du critère linguistique de propagation des attributs formels sur la tête des termes composés. Si nous souhaitons saturer les objets formels, deux possibilités s'offrent à nous : l'approche par emplois et l'approche par propriétés.

L'approche par emplois L'approche par emplois est l'application du critère linguistique mentionné plus haut. Si un terme[17] polylexical est sous la coupe d'un verbe en tant que sujet ou complément d'objet direct, alors sa tête l'est aussi. En termes de contexte formel, cela signifie qu'un nouvel objet formel est créé à partir de la tête du

17. ou un syntagme, la notion de spécificité au domaine n'a pas de conséquence ici.

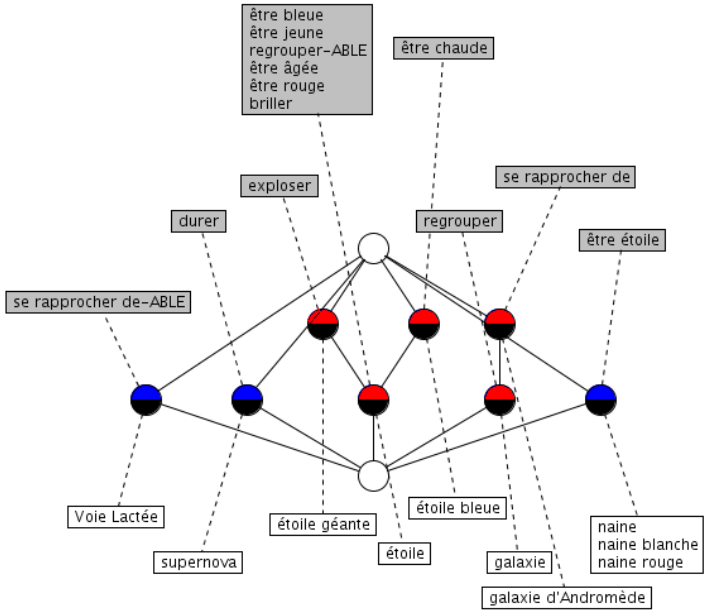

FIGURE 6.11 : Conséquences de l'approche par emplois

terme. Cet objet formel possède tous les attributs formels du terme source, en plus de tous les attributs formels des termes ayant la même tête. Or cette définition signifie que l'objet formel de saturation va entraîner la création d'un concept formel plus spécifique que ceux issus des termes composés, d'où une relation de subsomption inversée dans le treillis (voir l'exemple sur la figure 6.11). L'approche par emplois illustre le problème du passage des connaissances exprimées dans le texte à des connaissances ontologiques. Pour tenter de les traduire dans le treillis, nous proposons de raisonner plutôt en termes de propriétés ontologiques.

L'approche par propriétés Cette approche exprime des connaissances qui ne sont pas notées explicitement dans le texte. L'idée est que si deux termes composés ayant la même tête sont sous la coupe de deux verbes différents, alors la tête commune aura comme attributs formels l'intersection des attributs des termes composés. Deux cas se présentent : soit la tête commune n'existe pas dans le contexte formel, soit elle existe déjà.

Dans le premier cas, si la tête commune n'existe pas et que les termes composés n'ont aucun attribut en commun, aucun objet formel n'est créé. En revanche, si la tête commune n'existe pas, mais que les termes composés ont des attributs en commun, on

TABLEAU 6.15 : Illustration de l'approche par propriétés (1)

	tourner	être sobre	être lourd	être nocif	gpl-ABLE	chauffer	être sucré	être amer	(NEG) contenir cacao
moteur diesel		×	×	×					
moteur essence				×	×				
moteur	×					×			
chocolat au lait							×		×
chocolat noir							×	×	

crée un nouvel objet formel avec pour attributs l'intersection des attributs des termes composés de tête identique.

Si maintenant la tête des termes composés existe dans le contexte formel, elle doit donner ses attributs à ses termes composés, qui donnent également l'intersection de leurs attributs propres à la tête. Ce principe suppose que la tête est toujours employée dans son sens générique (« le moteur tourne »), et pas comme représentant elliptique d'un terme composé (« Je possède un moteur électrique. Ce moteur n'est pas nocif. »).

Les tableaux 6.15 et 6.16 viennent illustrer l'approche par propriétés. Dans le cas de `moteur diesel` et `moteur essence`, l'objet formel `moteur` existe. Il va donc donner ses attributs à `moteur diesel` et `moteur essence` et recevoir l'intersection de leurs attributs. Dans le cas du chocolat, l'objet formel `chocolat` n'existe pas. Il va donc être créé à l'aide de l'intersection des attributs de `chocolat au lait` et de `chocolat noir`. Cette intersection apporte une nouvelle information au contexte formel, car il n'existait aucun objet ayant pour attribut uniquement `être sucré`.

Pour le corpus de l'astronomie, en utilisant le contexte formel construit avec la variante `termlist` sans attributs étendus, nous trouvons 758 objets formels dont la tête est présente comme un autre objet formel et 216 objets pour lesquels un nouvel objet formel serait créé. Pour le corpus de la cuisine, il existe 594 objets formels avec tête présente (pour 96 qui n'existent pas dans le contexte formel). Le corpus du droit propose quant à lui 335 objets formels avec tête présente et 114 objets formels qui nécessitent la création d'un nouvel objet.

Cette approche de saturation des objets formels permet donc de transposer dans le contexte formel les relations d'hyperonymie qui existent dans les mots composés. Cependant ces relations sont fiables dans la mesure où le texte exprime des connaissances

TABLEAU 6.16 : Illustration de l'approche par propriétés (2)

	tourner	être sobre	être lourd	être nocif	gpl-ABLE	chauffer	être sucré	être amer	(NEG) contenir cacao
moteur diesel	×	×	×	×		×			
moteur essence	×			×	×	×			
moteur	×			×		×			
chocolat au lait							×		×
chocolat noir							×	×	
chocolat							×		

génériques sur les termes et les concepts auxquels ils renvoient, et non des connaissances particulières, spécifiques à telle ou telle instance de concept. Nous proposons donc un module interactif de saturation des objets formels, dans lequel l'utilisateur doit décider de la validité des relations proposées en consultant les phrases d'apparition.

Les relations de subsomptions explicites

La saturation des objets formels en utilisant une approche par propriétés est déjà une utilisation d'une forme de subsomption explicite ; cependant ici nous nous intéressons aux relations « à la Hearst », c'est-à-dire aux énoncés définitoires ou autres tournures caractéristiques de l'hyperonymie. Le principe est identique à celui présenté précédemment : si un concept A subsume un concept B, alors l'intension de B doit contenir au minimum l'intension de A. En termes de contexte formel, cela signifie que si le concept évoqué par un objet formel A subsume celui évoqué par un objet formel B, alors B doit posséder les attributs de A.

La détection des éléments en subsomption directe est un problème complexe, nous nous limitons pour l'instant au cas le plus simple, la forme « NOM est un NOM ». Même avec cette forme simple des problèmes peuvent survenir en cas de transitivité [18], nous choisissons ici de considérer un seul niveau de subsomption.

Si nous prenons comme exemple le texte présenté sur la figure 6.12, nous obtenons initialement le treillis de la figure 6.13. De nombreuses relations de subsomption

18. A est un B est un C.

– Riri est un chat.
– Fifi est un chat.
– Un chat est un animal à quatre pattes.
– Un chat est un animal poilu.
– Un chien est un animal poilu.
– Un chat est un animal à griffes rétractiles.
– Un chien est un animal à quatre pattes.
– Jean est heureux.
– Un animal poilu est un mammifère.
– Un animal n'est pas un végétal.
– Un mammifère est un animal.
– Un oiseau et un lézard sont des animaux.

FIGURE 6.12: Illustration des relations de subsomption explicites

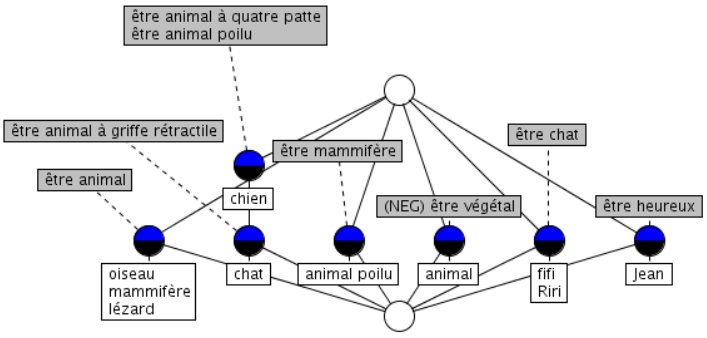

FIGURE 6.13: Treillis issu du texte présenté sur la figure 6.12

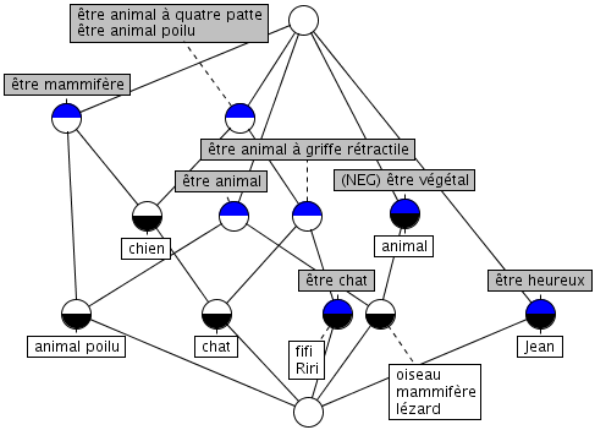

FIGURE 6.14 : Treillis issu du texte présenté sur la figure 6.12, variante avec relations explicites prises en compte

directes se retrouvent dans les attributs formels, elles sont traduites dans la figure 6.14. Les relations d'héritage que l'on aime lire sur le treillis sont correctes, même si elles paraissent curieuses (un chat est un chien) elles sont simplement l'illustration que la caractérisation des objets formels est importante. Ce treillis soulève de nombreuses interrogations. La première concerne le cas de Riri, Fifi et Jean, qui sont typiquement des instances de concepts, tandis que les autres extensions se réfèrent plutôt à des concepts. Ceci est un problème crucial de l'interprétation du treillis : nous avons tendance à vouloir associer directement les extensions à des concepts, et c'est d'autant plus vrai lorsque les extensions sont des termes.

6.4 Présentation du système

La figure 6.15 reprend la figure 5.4 en y ajoutant les modules de prédiction de performance (voir la section 6.1), de filtrage sur le contexte formel construit (voir la section 6.2) et d'ajout de connaissances implicites sur les objets formels (voir la section 6.3.3). Les flèches en pointillé représentent un traitement optionnel.

La figure 6.16 reprend quant à elle la vue détaillée de notre module de construction du contexte formel (figure 5.5) en y ajoutant (en vert) les composants de détection de la négation, d'extension / élargissement / saturation des attributs et enfin le module d'extension des objets formels.

Notre système a été implémenté en Perl, la documentation du module de construction du contexte formel est présentée dans l'annexe A.

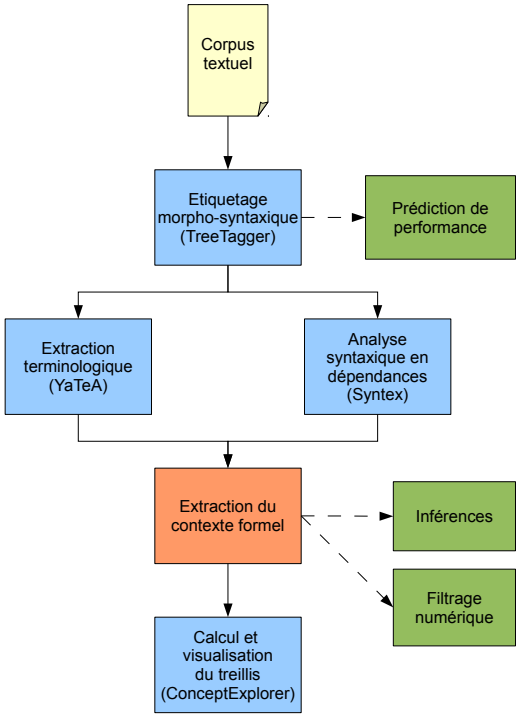

FIGURE 6.15: Vue globale du processus de fonctionnement de notre système (2)

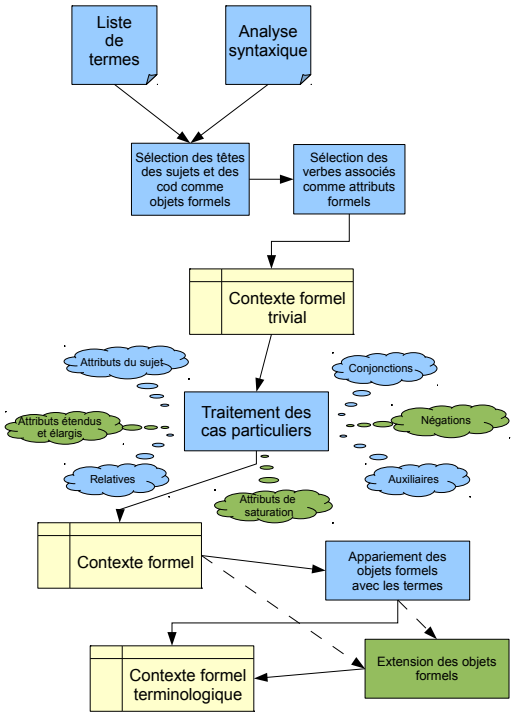

FIGURE 6.16: Le nouveau module d'extraction du contexte formel en détails

6.5 Conclusion

Dans ce chapitre nous avons présenté plusieurs solutions au problème de l'utilisation du treillis pour la COT. Nous avons tenté de résoudre le problème de la sensibilité au corpus en repérant des indicateurs de prédiction du nombre de concepts formels en corpus. Cependant, les indicateurs que nous avons exhibés ne se comportent pas de manière stable avec tous nos corpus ; peut-être qu'une caractérisation plus fine de ces corpus permettrait de mieux les cerner. Le filtrage du contexte formel par sélection des termes uniquement est une méthode simple et intuitive pour contenir le nombre de concepts formels, à condition de disposer d'une liste de termes de qualité. Les solutions que nous proposons pour améliorer la qualité du treillis mettent en avant le soin qu'il faut apporter à la représentation du texte sous la forme d'un contexte formel. Certains problèmes liés semblent toutefois épineux, notamment ceux que provoquent les emplois génériques *vs.* spécifiques des termes dans le texte. Un treillis construit uniquement sur des emplois génériques de termes serait valide pour une interprétation en ontologie mais les emplois spécifiques viennent le polluer. Il faudrait être capable de détecter ces cas à l'aide d'indicateurs linguistiques, mais la détection automatique des énoncés génériques *vs.* spécifiques est difficile et hors de portée du présent travail. Cette détection doit se faire manuellement et seule une interface de validation peut aider ce travail. Le chapitre suivant est consacré à la présentation d'expériences visant à évaluer la qualité des regroupements proposés par le treillis, et à mesurer le taux de généricité, c'est-à-dire la proportion d'apparitions génériques, des attributs formels.

Chapitre 7

Nouvelles expérimentations, vers un treillis de meilleure qualité

Sommaire

Dans le chapitre précédent, nous avons proposé d'améliorer le processus de construction du contexte formel, afin d'obtenir un treillis qui soit un reflet plus fidèle et plus intelligible des connaissances contenues dans les textes. Ce chapitre présente les nouvelles expérimentations que nous avons menées pour mesurer la qualité d'un treillis. Ces expérimentations portent sur le domaine de l'astronomie ; elles sont centrées autour de deux treillis que nous présentons dans une première section. Dans la section suivante, nous évaluons la qualité informative des éléments intensionnels des treillis, pour poursuivre en mesurant leur taux de généricité. Nous terminons ce chapitre en comparant les regroupements proposés par l'ACF avec une ontologie de référence.

7.1 Présentation des treillis

Nous avons construit deux treillis sur le domaine de l'astronomie. Le premier treillis est construit à l'aide de la méthode `termlistonly` munie d'une partie des améliorations présentées dans le chapitre précédent (la saturation, l'extension des attributs formels et la négation). Nous n'avons pas utilisé les modules de filtrage numérique et d'inférence afin de rester le plus fidèle possible au corpus. Comme liste de termes, nous avons utilisé

TABLEAU 7.1 : Caractéristiques du treillis

Nombre d'objets	24
Nombre d'attributs	720
Dont négatifs	*54*
Dont saturation	*89*
Nombre de concepts	104
Niveau 6 (+spécifique)	*24*
Niveau 5	*44*
Niveau 4	*22*
Niveau 3	*7*
Niveau 2	*4*
Niveau 1 (+général)	*3*

les titres des articles Wikipedia utilisés pour constituer le corpus de l'astronomie (voir la section 5.1.3), à l'exception de « évolution des étoiles », « naissance des étoiles » et « Soleil », qui n'évoquent pas directement des concepts [1]. Nous obtenons donc en entrée de notre système une liste de 24 termes. Le treillis, présenté sur la figure 7.1, comporte 104 concepts, 24 objets et 720 attributs, 6 niveaux de profondeur. Le tableau 7.1 montre la répartition des concepts formels selon leur position dans le treillis, sans compter le *top* ni le *bottom*. Les attributs *négatifs* sont issus d'une tournure négative de phrase, les attributs de saturation sont ceux qui sont utilisés pour ajouter de la structure au treillis ajoutant à l'objet la réduction en forme verbale + préposition des attributs usuels (voir la section 6.3.1).

Le second treillis est construit en utilisant l'extension des objets formels (voir la section 6.3.2) : par exemple l'objet `galaxie` sera étendu en `galaxie naine`. Combiner la variante `termlistonly` avec l'option `expand` implique que seuls les termes fournis en entrée du système seront étendus. Le tableau 7.2 montre l'influence quantitative de l'utilisation des objets formels étendus sur le treillis. Les attributs formels qui composent le contexte formel sont les mêmes que ceux du premier treillis, c'est leur répartition vis-à-vis des objets qui change. Nous constatons à la lecture de cette table que l'incorporation des objets formels étendus multiplie par 6,5 le nombre d'objets formels et par plus de 2 le nombre de concepts formels [2]. Les nombres entre parenthèses après les nombres de concepts formels indiquent combien de ces derniers possèdent une extension propre [3].

1. Ce choix est évidemment discutable. Les deux premiers termes évoquent des processus et ne sont pas sur le même plan que les autres termes retenus. « Soleil » évoque quant à lui une instance de concept.
2. Pour des raisons de lisibilité nous ne pouvons pas présenter ce treillis dans la globalité.
3. Dans le treillis précédent, les seuls objets avec extension propre sont ceux du niveau le plus spécifique.

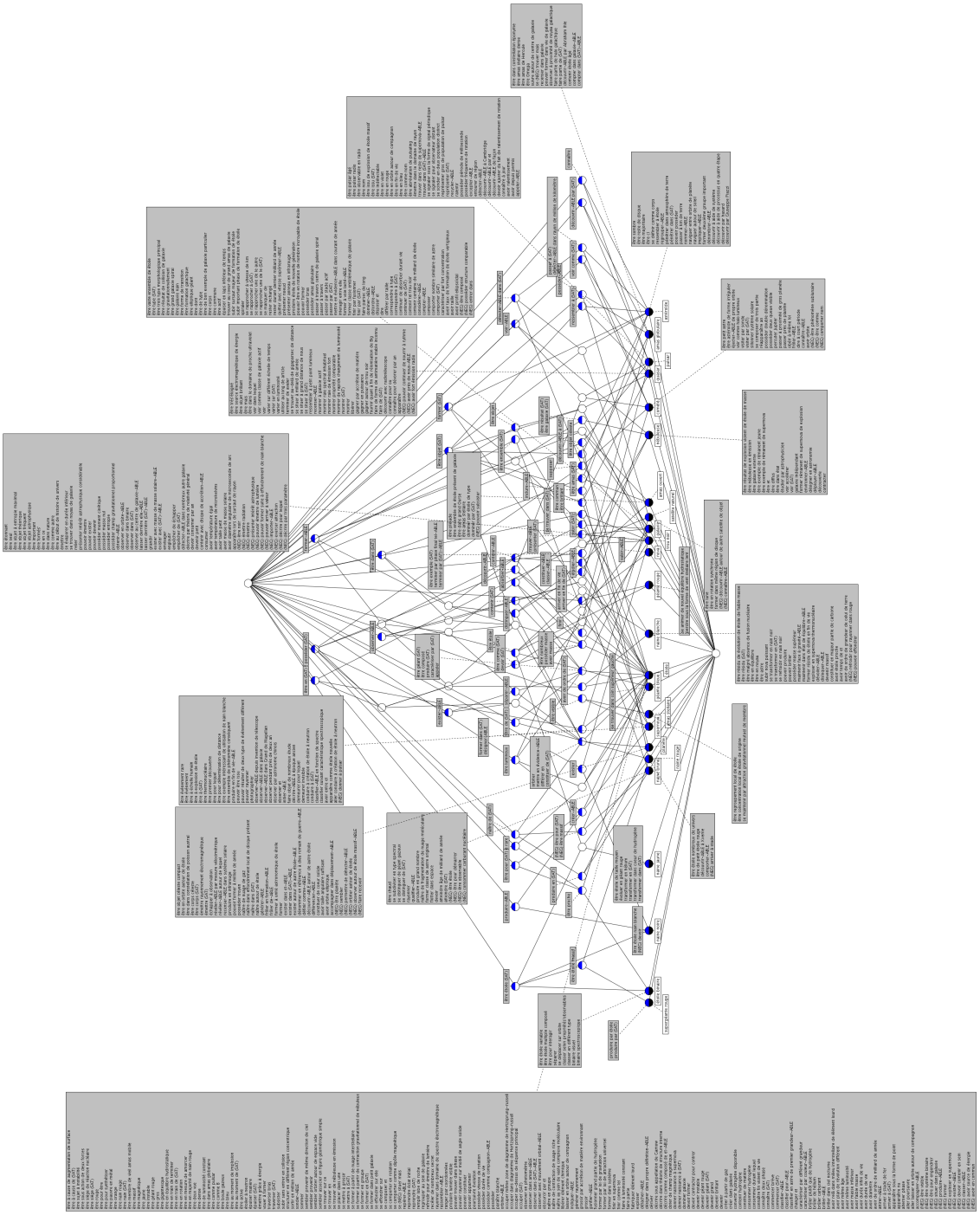

FIGURE 7.1 : Premier treillis

TABLEAU 7.2: Influence sur le treillis de l'utilisation des objets formels étendus

	treillis 1	treillis 2
Objets formels	24	154
Attributs formels	720	720
Concepts formels (dont extensions propres)	104(24)	239(150)
Niveau 6 (+spécifique)	*24*(24)	*130*(130)
Niveau 5	*44*(0)	*73*(10)
Niveau 4	*22*(0)	*23*(6)
Niveau 3	*7*(0)	*8*(3)
Niveau 2	*4*(0)	*2*(0)
Niveau 1 (+général)	*3*(0)	*3*(1)

Nous pouvons remarquer que certains concepts qui ne sont pas les plus spécifiques possèdent une extension propre. La somme des concepts ayant une extension propre peut être inférieure au nombre d'objets formels, car certains concepts ont plusieurs objets comme extension propre.

Le treillis construit avec les objets formels étendus contient plus d'informations que le premier treillis, mais il en devient plus difficile à lire, en raison de la quantité de concepts formels. Il serait intéressant de mettre en parallèle les deux treillis pour pouvoir identifier sur le premier les attributs formels qui se retrouvent dans des objets formels étendus du deuxième.

7.2 La pertinence des éléments du treillis vis-à-vis du domaine

Dans cette section nous mesurons la pertinence des éléments du treillis (les attributs formels et les concepts formels) vis-à-vis du domaine étudié. Tout d'abord, définissons ce que nous appelons pertinence.

Définition 20 (Attribut pertinent pour le domaine) *Un attribut formel est dit* pertinent pour le domaine *s'il évoque une propriété, une notion spécifique au domaine.*

Par exemple, `briller` est un attribut pertinent pour le domaine de l'astronomie, tandis que `subir` ne l'est pas. Cette notion de pertinence est totalement dépendante des connaissances de l'expert. Un attribut pertinent pour le domaine ne veut pas dire qu'il est correct vis-à-vis de l'objet formel sur lequel il est appliqué. Par exemple, l'attribut `être géante rouge` n'est pas correct s'il est appliqué à l'objet formel `naine rouge`.

Par extension, nous définissons la pertinence d'un concept formel :

Définition 21 (Concept formel pertinent pour le domaine) *Un concept formel est dit* pertinent *si au moins l'un des attributs de son intension l'est.*

De manière analogue à la définition d'un attribut pertinent, le fait qu'un concept formel soit pertinent ne signifie en aucun cas que son intension caractérise de manière correcte son extension. Cela indique simplement que le concept formel est susceptible d'apporter des informations sur le domaine, qu'il n'est pas uniquement constitué de bruit. Par exemple, le concept (`nébuleuse, galaxie`; `être résultat (SAT), être galaxie (SAT)`) est pertinent grâce à son attribut `être galaxie (SAT)`, mais le regroupement n'est pas correct, car une nébuleuse n'est typiquement pas une galaxie. Nous avons manuellement évalué la pertinence de chacun des 640 attributs propres des deux treillis, en utilisant nos connaissances du domaine.

7.2.1 Pour le treillis 1

Le tableau 7.3 montre pour chaque objet formel issu de la liste de termes la proportion d'attributs pertinents par rapport à l'ensemble de ses attributs. La précision moyenne de 42,66% indique qu'un peu moins de la moitié des attributs propres du treillis est pertinente par rapport au domaine. Certains objets comme `amas stellaire` possèdent uniquement des attributs pertinents (`être regroupement local d'étoiles`, `être concentration locale d'étoiles` et `se maintenir par attraction gravitationnelle mutuelle des membres`). D'autres, comme `géante bleue` ou `nova`, n'ont aucun attribut pertinent : le seul attribut propre de `géante bleue` est `se trouver dans coin supérieur gauche` tandis que `nova` est caractérisée par `subir-ABLE`. Les attributs non pertinents regroupent à la fois les erreurs d'analyse du système (`être pour être`) et les éléments que nous jugeons trop flous (`être à cause de augmentation de surface`, `terminer de avaler`) ou trop génériques (`découvrir par hasard`).

Dans le tableau 7.4 nous analysons la pertinence pour le domaine des attributs de regroupement (par opposition aux attributs propres). Ce tableau présente les attributs en fonction de la hauteur du concept formel sur lequel ils portent, avec 1 correspondant au niveau le plus général dans le treillis. Nous remarquons que plus nous remontons dans la hiérarchie du treillis, plus le nombre d'attributs pertinents diminue. Au premier niveau de regroupement nous retrouvons des attributs pertinents comme `être objet céleste`, `être étoile massive`. Certains attributs de saturation ou négatifs se retrouvent aussi comme attributs pertinents (`(NEG) être massif`, `être amas (SAT)`). Au niveau le plus général les attributs sont `être en (SAT)`, `posséder (SAT)` et `former-ABLE`, aucun n'est pertinent [4].

4. Ou alors pour une ontologie d'un niveau plus haut, par opposition à une ontologie de domaine.

TABLEAU 7.3: Pertinence des attributs propres

Objet	**Attributs pertinents**	**Attributs totaux**	**Précision**
étoile	81	195	41,54%
naine noire	1	2	50,00%
étoile binaire	5	9	55,56%
naine jaune	4	6	66,67%
naine brune	9	17	52,94%
naine rouge	2	5	40,00%
planète	15	40	37,50%
supernova	11	35	31,43%
amas stellaire	3	3	100,00%
géante bleue	0	1	0,00%
supergéante rouge	1	2	50,00%
géante rouge	2	2	100,00%
naine blanche	14	26	53,85%
nova	0	1	0,00%
trou noir	26	61	42,62%
galaxie	27	70	38,57%
satellite naturel	2	5	40,00%
amas ouvert	2	7	28,57%
nébuleuse	6	18	33,33%
comète	11	22	50,00%
quasar	21	40	52,50%
pulsar	17	37	45,95%
amas globulaire	6	15	40,00%
astéroïde	7	21	33,33%
Moyenne			**42,66%**

TABLEAU 7.4: Pertinence des attributs de regroupement pour le domaine

Niveau	**Attr. Pertinents**	**Attr. totaux**	**Précision**
5	19	42	45,24%
4	10	20	50,00%
3	1	3	33,33%
2	1	2	50,00%
1	0	3	0,00%
Moyenne			**44,28%**

TABLEAU 7.5 : Pertinence des concepts

Niveau	Concepts pertinents	Total	Précision
6	24	24	100,00%
5	33	44	75,00%
4	13	22	59,09%
3	3	7	42,86%
2	1	4	25,00%
1	0	3	0,00%
Moyenne			**71,15%**

Maintenant que nous avons mesuré la proportion d'attributs pertinents, nous proposons pour terminer cette expérience d'évaluer la quantité de *concepts formels pertinents* dans le treillis. Le tableau 7.5 montre la pertinence des concepts formels. Par exemple, le concept (`étoile, quasar` ; `être commun, être brillant`) est pertinent.

Nous remarquons que comme pour la pertinence des attributs de regroupement, la proportion de concepts formels pertinents diminue au fur et à mesure que l'on monte dans le treillis. Nous avons mesuré la pertinence des éléments du treillis vis-à-vis du domaine. Pour l'astronomie, nous constatons qu'environ la moitié des attributs formels sont utiles, ainsi que 70% des concepts formels.

7.2.2 Pour le treillis 2

Les attributs formels sont par construction identiques à ceux de l'expérience précédente ; c'est leur répartition au sein des concepts formels qui change. Le tableau 7.6 présente pour chaque concept formel, dont l'extension propre appartient à la liste des 24 termes initiaux de l'expérience précédente [5], le nombre d'attributs propres et la proportion d'attributs propres pertinents pour le domaine. Nous constatons à la lecture de ce tableau que l'utilisation des objets formels étendus n'a quasiment pas d'influence sur la proportion d'attributs pertinents dans les objets formels non étendus. En revanche, certains objets formels comme `galaxie` ou `étoile` ont vu leur nombre d'attributs fortement diminuer, ce sont ceux qui ont été les plus étendus.

Prenons comme exemple le terme *étoile*. Il a été étendu en *étoile avec masse suffisante, étoile binaire, étoile binaire spectroscopique, étoile bleue, étoile brillante, étoile chaude, étoile chaude magnétique, étoile comme soleil, étoile de Barnard, étoile d'amas, étoile de compagnon, étoile de constellation, étoile de célèbre astérisme, étoile de faible masse, étoile de forte masse, étoile de galaxie, étoile de masse, étoile de nouvelle génération, étoile de première génération, étoile de type, étoile de type particulier, étoile de type solaire, étoile dense, étoile double, étoile en fin de vie, étoile en formation, étoile filante,*

5. Nous n'y avons pas reporté l'intégralité des 130 objets formels étendus supplémentaires.

TABLEAU 7.6: Pertinence des attributs propres, treillis avec objets formels étendus

Objet	**Attr. Pertinents**	**Attr. Totaux**	**Précision**
étoile	59	121	48,76%
naine noire	1	2	50,00%
étoile binaire	4	8	50,00%
naine jaune	4	6	66,67%
naine brune	9	15	60,00%
naine rouge	2	5	40,00%
planète	15	36	41,67%
supernova	9	25	36,00%
amas stellaire	3	3	100,00%
géante bleue	0	1	0,00%
supergéante rouge	1	2	50,00%
géante rouge	2	2	100,00%
naine blanche	13	24	54,17%
nova	0	1	0,00%
trou noir	23	49	46,94%
galaxie	13	40	32,50%
satellite naturel	1	3	33,33%
amas ouvert	2	7	28,57%
nébuleuse	0	3	0,00%
comète	10	19	52,63%
quasar	20	34	58,82%
pulsar	13	29	44,83%
amas globulaire	4	11	36,36%
astéroïde	7	18	38,89%
Moyenne des non étendus			**46,34%**
galaxie lenticulaire	1	3	33,33%
étoile massive	9	20	45,00%
...	...	...	...

étoile froide, étoile jeune, étoile jeune dans constellation d'Orion, étoile lumineuse, étoile massive, étoile naine, étoile naine blanche, étoile ordinaire, étoile pauvre, étoile petite, étoile proche du centre galactique, étoile riche, étoile variable, étoile voisine, étoile à neutrons, étoile âgée.

Dans cette liste, outre les erreurs d'analyse (`étoile comme soleil`, `étoile de masse`), nous pouvons remarquer des expansions qui énoncent d'autres concepts au sens ontologique (étoile double, étoile à neutrons, étoile filante), d'autres qui pourraient se référer à des fils du concept étoile (étoile bleue, étoile âgée). D'autres encore dénotent clairement des instances du concept d'étoile (étoile jeune dans constellation d'Orion, étoile proche du centre galactique).

À cause de la trop grande quantité de concepts formels du treillis, nous n'avons pas pu mener à bien la mesure du nombre de concepts formels pertinents. Ces expériences nous enseignent que malgré la nature encyclopédique du corpus, les treillis qui en sont issus ne comportent qu'une moitié d'attributs formels pertinents pour le domaine. L'incorporation des objets formels étendus permet une redistribution homogène des attributs formels, dans le sens où elle ne modifie pas la proportion d'attributs pertinents dans les objets formels initiaux. Nous pensons que les deux treillis sont utiles : le premier garde une taille raisonnable pour être lisible, tandis que le second apporte plus de précision sur les objets formels. Il serait intéressant de différencier visuellement, sur le premier treillis, les attributs formels qui se retrouvent comme attributs propres d'objets formels étendus du second treillis.

7.3 Mesure du taux de généricité du contexte formel

Dans cette expérience nous allons mesurer le *taux de généricité* du premier treillis (celui composé des 24 objets formels). Par taux de généricité, nous entendons la proportion des attributs du contexte formel qui sont des attributs génériques vis-à-vis des objets formels ayant une extension propre.

Définition 22 (Attribut générique) *Un attribut formel est dit* générique *vis-à-vis d'un objet formel s'il traduit une propriété du* concept *évoqué par cet objet.*

Définition 23 (Attribut spécifique) *Un attribut formel est dit* spécifique *vis-à-vis d'un objet formel s'il traduit une propriété d'un* sous-concept *ou d'une* instance *du concept évoqué par cet objet.*

Par exemple, en astronomie, l'attribut `explose` est un attribut spécifique vis-à-vis de l'objet formel `étoile` car toutes les étoiles n'explosent pas, tandis que c'est un attribut générique vis-à-vis de l'objet formel `supernova` car une supernova explose systématiquement.

TABLEAU 7.7 : Attributs propres génériques ou spécifiques

Objet	**gén.**	**% gén. / pert.**	**% gén. / total**	**spé.**	**% spé. / pert.**	**% spé. / total**
étoile	41	50,62%	21,03%	40	49,38%	20,51%
naine noire	1	100,00%	50,00%	0	0,00%	0,00%
étoile binaire	2	40,00%	22,22%	3	60,00%	33,33%
naine jaune	4	100,00%	66,67%	0	0,00%	0,00%
naine brune	7	77,78%	41,18%	2	22,22%	11,76%
naine rouge	2	100,00%	40,00%	0	0,00%	0,00%
planète	10	66,67%	25,00%	5	33,33%	12,50%
supernova	6	54,55%	17,14%	5	45,45%	14,29%
amas stellaire	3	100,00%	100,00%	0	0,00%	0,00%
géante bleue	0	0,00%	0,00%	0	0,00%	0,00%
supergéante rouge	1	100,00%	50,00%	0	0,00%	0,00%
géante rouge	1	50,00%	50,00%	1	50,00%	50,00%
naine blanche	10	71,43%	38,46%	4	28,57%	15,38%
nova	0	0,00%	0,00%	0	0,00%	0,00%
trou noir	12	46,15%	19,67%	14	53,85%	22,95%
galaxie	6	22,22%	8,57%	21	77,78%	30,00%
satellite naturel	1	50,00%	20,00%	1	50,00%	20,00%
amas ouvert	2	100,00%	28,57%	0	0,00%	0,00%
nébuleuse	2	33,33%	11,11%	4	66,67%	22,22%
comète	3	27,27%	13,64%	8	72,73%	36,36%
quasar	6	28,57%	15,00%	15	71,43%	37,50%
pulsar	6	35,29%	16,22%	11	64,71%	29,73%
amas globulaire	3	50,00%	20,00%	3	50,00%	20,00%
astéroïde	4	57,14%	19,05%	3	42,86%	14,29%
		48,71%	**20,78%**		**51,28%**	**21,87%**

Certaines phrases expriment des connaissances sur les concepts, comme dans *les étoiles sont le siège de réactions nucléaires*, elles donnent un attribut générique. D'autres en revanche expriment des connaissances sur des éléments qui pourraient être des sous-concepts ou des instances (*Les étoiles de ces amas sont généralement des géantes rouges*[6]), elles produiront un attribut spécifique.

Le tableau 7.7 présente pour chaque concept qui possède une extension propre le nombre d'attributs propres *pertinents* génériques (colonne `gén.`) et le nombre d'attributs propres *pertinents* spécifiques (colonne `spé.`). Les colonnes `% gén. / pert.` et `% gén. / total` présentent la proportion d'attributs génériques/spécifiques parmi

6. La spécificité est marquée à la fois par l'adverbe *généralement* et par le complément du nom *de ces amas*.

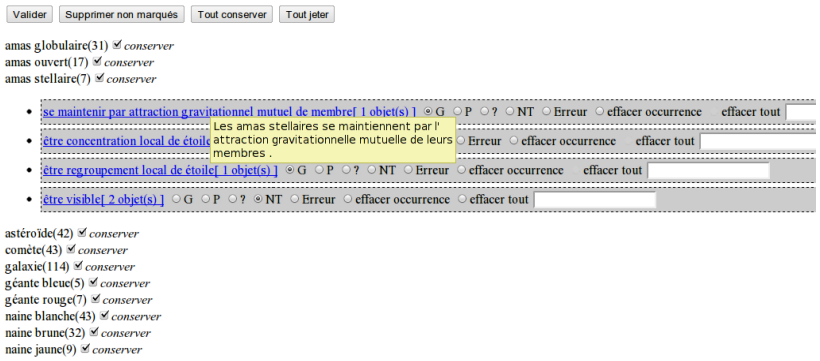

FIGURE 7.2 : L'interface d'évaluation de la généricité des attributs formels

les attributs pertinents et le total des attributs, respectivement. Par exemple, l'objet formel `comète` possède trois attributs pertinents génériques : `être petit astre`, `être astéroïde de forme irrégulière` et `se composer de trois parties`. Comme attributs spécifiques de `comète`, nous pouvons mentionner `percuter Jupiter`, `posséder deux queues visibles`.

La distinction entre génériques et spécifiques est effectuée manuellement en utilisant nos connaissances sur le domaine. Nous avons évalué chacun des 273 couples *(objet formel, attribut pertinent)*, en nous référant au texte en cas de doute. Cette distinction est parfois difficile à établir, par exemple `exploser en supernova thermonucléaire` appliqué à `naine blanche` est un attribut spécifique, mais cela suppose de savoir *a priori* que toutes les naines blanches ne terminent pas leur vie de cette manière. Cette expérience montre que le corpus de l'astronomie est composé majoritairement d'attributs génériques, ce qui est cohérent au vu de sa nature encyclopédique.

Durant notre évaluation, que nous avons réalisée sur papier, nous est venue l'idée d'une interface d'aide. Nous avons développé cette interface sous la forme d'un programme qui transforme un contexte formel en une page web dynamique. À partir de cette page, dont une capture d'écran est présentée sur la figure 7.2, l'utilisateur peut décider pour chaque objet formel si l'attribut formel associé (les attributs propres portent l'indication « 1 objet(s) » entre crochets) est *générique, spécifique, indécidable* (le « ? »), *non traité* (le « NT »), *issu d'une erreur d'analyse mais à conserver*, ou *à supprimer définitivement*. Une zone de texte libre permet d'inscrire un commentaire sur la décision. L'utilisateur a accès, en passant son curseur sur l'attribut, aux phrases desquelles il est issu. Un clic sur un attribut montre à l'utilisateur tous les objets formels qui le possèdent. La sauvegarde du travail en cours s'effectue à l'aide de la fonction d'enregistrement du navigateur. Une fois le contexte formel traité, l'utilisateur valide et un script se charge de construire six contextes formels qui correspondent aux différents choix pour chaque attribut (générique, spécifique, indécidable, non traité, erreur, supprimer). Le résultat est présenté sur la figure 7.3 : l'utilisateur peut directement incorporer, par copier-coller, les contextes formels dans ConceptExplorer.

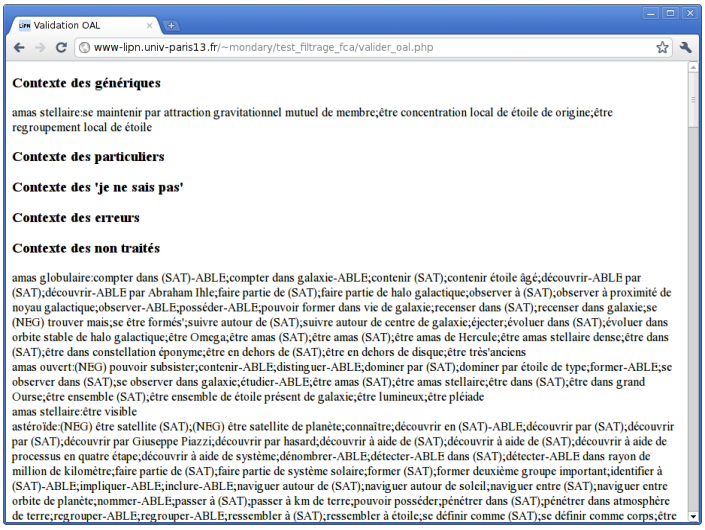

FIGURE 7.3 : Les contextes formels validés

La distinction entre attribut générique et attribut spécifique est **cruciale** dans le cadre de l'interprétation d'un treillis en ontologie. En effet l'ACF est conçue pour ne fonctionner qu'avec des attributs génériques sur les objets, car l'intension du concept formel est formée de la *conjonction* des attributs partagés. Dans le cadre des attributs spécifiques, la conjonction des attributs se trouve être sémantiquement erronée (une étoile ne peut pas `être âgée` et `être jeune` simultanément), nous avons affaire à une disjonction des propriétés.

Les entités nommées fonctionnent différemment des termes. Comme elles font généralement référence de manière stable à un objet bien précis du domaine, leurs attributs ont plus de chances de s'interpréter comme des propriétés ontologiques, en tout cas dans un état du monde considéré comme stable. La conjonction des attributs est dans ce cas valide (« Mars est rouge. Mars est tellurique »), on ne retrouvera pas deux propriétés contradictoires, comme dans le cas d'un terme (« L'étoile est rouge. Une étoile est bleue. »). C'est la raison pour laquelle le système présenté dans (Bendaoud *et al.*, 2007) fonctionne bien : en travaillant sur un catalogue d'objets célestes et des comptes rendus d'observation, les objets formels qui composent le treillis sont des entités nommées. Il n'y a pas d'attributs spécifiques qui viennent polluer les regroupements.

7.4 Évaluation de la qualité des regroupements

Dans cette partie, nous présentons une expérience pour évaluer qualitativement les regroupements que propose l'ACF, sur le domaine de l'astronomie. Cette expérience, qui doit permettre de cerner les parties utiles du treillis de concepts, se déroule en deux étapes. Dans un premier temps, nous construisons une ontologie de référence à

partir de textes, selon une approche terminologique. Ensuite, nous confrontons cette ontologie avec un treillis de concepts issu du même corpus.

7.4.1 Description de l'ontologie de référence

À l'aide du corpus de l'astronomie présenté dans la section 5.1.3 nous avons construit une ontologie de référence en utilisant une approche terminologique (Terminae). Nous n'avons pas trouvé d'ontologie des objets célestes en français, nous avons donc choisi de la construire en utilisant le corpus et notre propre expertise du domaine. En construisant une ontologie à partir du corpus, nous avons l'assurance qu'elle y est adaptée, c'est-à-dire qu'elle ne mentionne pas d'éléments du domaine qui ne se trouvent pas dans le corpus. L'ontologie, dont la partie hiérarchique est présentée sur la figure 7.4, vise à représenter les objets célestes usuels, en se concentrant plus précisément sur l'évolution des étoiles. Nous avons construit cette ontologie dans le cadre d'une tâche (fictive) de prédiction et de caractérisation, elle doit permettre de répondre aux questions telles que « Quel est le devenir probable de cette étoile ? », « À quelle catégorie appartient cet objet céleste ? ».

Nous sommes partis des titres de la plupart des articles du corpus pour construire les concepts principaux, que nous avons ensuite structurés avec l'aide du corpus. Nous avons érigé les termes des titres en concepts que nous avons ensuite spécialisés et généralisés. Notre ontologie est constituée de 73 concepts. Dans la suite, nous nous concentrons uniquement sur la partie hiérarchique de l'ontologie, qui nous intéresse pour l'évaluation de l'ACF. Nous avons validé *a posteriori* notre ontologie à l'aide de l'ontologie formelle « Ontology of Astronomical Object Types, v1.20 [7] ». Cette ontologie couvre bien plus de notions que notre corpus, et ne possède pas de pendant terminologique. Nous avons vérifié manuellement que les relations de subsomption de notre ontologie se retrouvaient dans la référence. Dans la suite de cette section, nous présentons quelques concepts de notre ontologie.

Concepts de structuration

Nous avons choisi d'exprimer quatre concepts au premier niveau pour structurer notre ontologie :

objetCéleste représente tous les objets du ciel, aussi bien les objets discrets (les astres) que les regroupements (par exemple les galaxies) et les objets diffus (les nébuleuses).

7. http://www.ivoa.net/Documents/latest/AstrObjectOntology.html

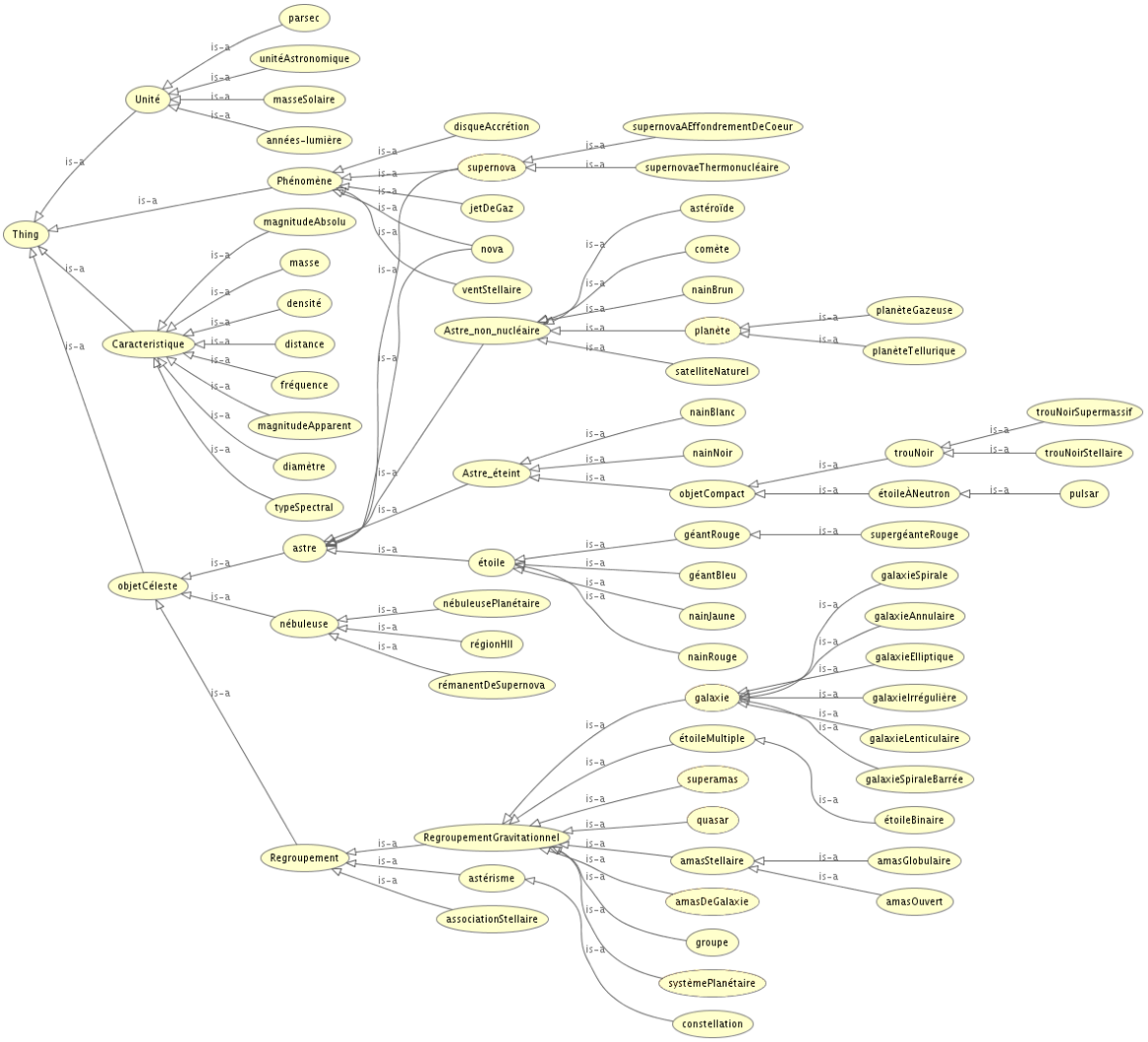

FIGURE 7.4 : Partie hiérarchique de l'ontologie de référence.

Caractéristique exprime les différentes grandeurs qui permettent de caractériser une étoile. La plupart de ces éléments auraient pu être exprimés sous forme de *data properties* mais nous avons choisi de les représenter sous forme de concepts afin de mettre davantage en avant ce qui caractérise une étoile : en effet la donnée de ces grandeurs peut suffire à la caractériser totalement.

Unité exprime quelques unités spécifiques au domaine de l'astronomie : `parsec`, `unitéAstronomique`, `annéeLumière`, `masseSolaire`

Phénomène représente quelques manifestations intéressantes ayant trait au cycle de vie des étoiles, comme l'explosion en `supernova`, le `vent stellaire` ou le `jet de gaz`.

Les objets célestes

Nous avons choisi de faire la distinction entre les `astres` (objets célestes discrets « atomiques »), les `nébuleuses` (nuages diffus qui ne sont ni des astres ni des regroupements) et les `regroupements` (qui sont composés d'astres et de nébuleuses).

Les objets célestes regroupés, fils du concept `Regroupement`, sont de plusieurs types :

regroupementGravitationnel Ce concept exprime les regroupements d'objets célestes liés par la force gravitationnelle, que ce soit des galaxies, des systèmes planétaires, des étoiles multiples ou des quasars.

astérisme Ce concept représente les constellations au sens large, c'est-à-dire toutes les figures remarquables dessinées par des étoiles brillantes, sans que celles-ci soient forcément dans la même constellation. Par exemple le triangle d'été (Deneb du Cygne, Vega de la Lyre, Altaïr de l'Aigle)

associationStellaire Ce concept traduit un ensemble d'étoiles non liées par l'attraction gravitationnelle, mais qui auraient une origine commune.

Voici à titre d'exemple quelques objets célestes, fils du concept `astre` :

étoile Par étoile, nous entendons un astre nucléaire typique, c'est-à-dire un astre qui produit de la lumière à l'aide de réactions de fusion thermonucléaire de l'hydrogène ou de ses dérivés. Ainsi nous considérons qu'une `étoile à neutrons` ou qu'une `naine blanche` *ne sont pas* des étoiles.

astreÉteint Ce concept représente un astre qui a terminé son cycle de fusion thermonucléaire pour dégénérer en naine ou en objet compact.

astreNonNucléaire Ce concept présente un astre qui n'a jamais entamé un cycle de fusion thermonucléaire, à cause d'une masse trop faible. Ce concept comprend les concepts de `planète`, `naine brune`...

nova Ce concept un peu particulier représente une étoile en train d'exploser. Ce concept est aussi un `Phénomène`.

Nous avons omis le concept d'étoile variable, car il est à notre avis trop complexe à définir proprement. Nous nous sommes contenté d'exprimer le regroupement `étoile multiple` comme exemple implicite d'étoile variable optique. Nous incluons aussi dans notre ontologie quelques objets célestes diffus, fils de `nébuleuse`.

Des phénomènes

Nous proposons de classer certains objets célestes comme étant à la fois des `astres` et des `phénomènes`[8]. Une `supernova` est un phénomène éphémère dû à l'effondrement d'une étoile (ou d'une naine blanche) suffisamment massive sur elle-même. Pendant son stade (temporaire) de supernova, une étoile voit sa magnitude augmenter brutalement. Selon la nature (géante ou naine blanche) et la masse de l'astre initial, la supernova évoluera en un `astre éteint` ou sera totalement désintégrée. Une `nova` est un phénomène similaire, mais moins énergétique qui résulte de l'explosion des couches de surface d'un astre sans transformation de celui-ci. Le `vent stellaire` est l'éjection des couches supérieures d'une étoile.

Des concepts primitifs qui se définissent

Nous avons spécialisé les concepts initiaux en utilisant le corpus, ce qui nous a parfois amené à transformer les concepts primitifs en concepts définis. Par exemple le concept initialement primitif de `galaxie` a été défini comme l'union disjointe des différentes formes de galaxies (spirale, elliptique, annulaire, irrégulière, lenticulaire, spirale barrée). De même, une planète est l'union disjointe de `planèteTellurique` et `planèteGazeuse`. Un `quasar` peut être défini comme un ensemble composé d'un trou noir supermassif, d'un disque d'accrétion et de jets de gaz.

7.4.2 Pour le treillis 1

Nous nous intéressons ici à l'évaluation des regroupements : est-ce que les regroupements proposés par le treillis se justifient vis-à-vis de l'ontologie de référence ? Nous allons analyser chaque branche de l'ontologie de référence (en associant les dénominations des concepts de la branche aux extensions du treillis) et mesurer la couverture des regroupements proposés par le treillis. Comme référence nous considérons les regroupements du tableau 7.8. Ces regroupements[9] sont issus du filtrage de l'ontologie de référence

8. Cela implique que ces concepts sont non-disjoints.

9. Nous notons les regroupements de références issus de l'ontologie *Rxx* et les regroupements calculés issus de l'ACF *Cxx*.

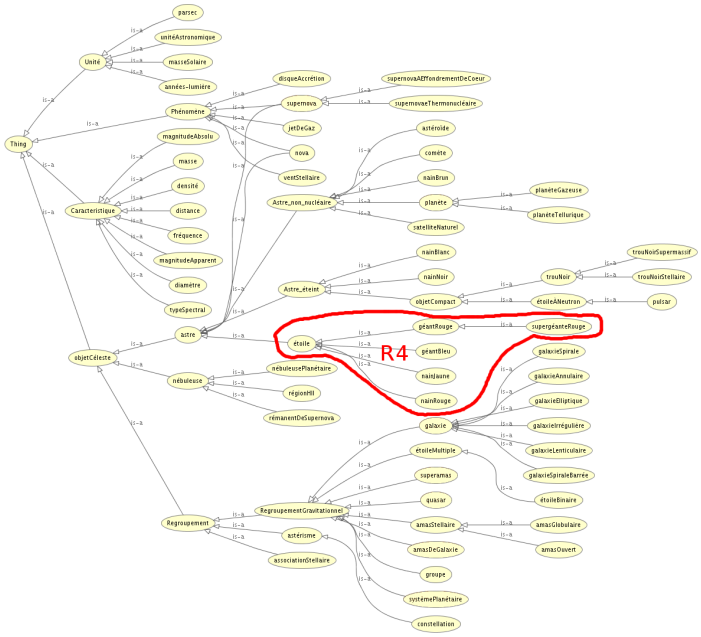

FIGURE 7.5 : Construction des regroupements de référence

TABLEAU 7.8 : Regroupements de référence

	Regroupement
R1	astéroïde, comète, naine brune, planète, satellite naturel
R2	amas stellaire, amas globulaire, amas ouvert
R3	galaxie, amas stellaire, amas globulaire, amas ouvert, quasar, étoile binaire
R4	étoile, géante rouge, géante bleue, naine jaune, naine rouge, supergéante rouge
R5	trou noir, pulsar
R6	supernova, nova
R7	naine blanche, naine noire

par les termes susceptibles d'apparaître dans le treillis. Ils correspondent à des concepts cohérents qui peuvent parfois être dénotés par leur père dans l'ontologie : par exemple *trou noir, pulsar* correspond à des *objets compacts*. La figure 7.5 montre R4.

Parmi les 104 concepts formels repérés par l'analyse de concepts formels, nous considérons uniquement les 80 dont l'extension contient au moins deux éléments. Nous avons calculé combien correspondent [10] exactement à des regroupements de référence ($C = R$), combien sont des sous-ensembles stricts des références ($C \subset R$), combien sont des sur-ensembles stricts ($C \supset R$).

Nous avons analysé combien de regroupements se rapprochent des références à une substitution près (par exemple on dira que $C1 = (a, b, c)$ et $R1 = (b, c, d)$ sont proches

10. En considérant leur extension.

TABLEAU 7.9 : Pertinence des regroupements

	Quantité	Dont pert.
$C = R$	0	0
$C \subset R$	10	9
$C \supset R$	4	0
C proche R	21	13
$C \neq R$	45	

TABLEAU 7.10 : Regroupements calculés contenus dans la référence

	Regroupement	$\subset$
C1	astéroïde, comète	R1
C2	naine brune, planète	R1
C3	astéroïde, planète	R1
C4	amas globulaire, amas ouvert	R2
C5	amas globulaire, amas ouvert	R3
C6	amas ouvert, galaxie	R3
C7	géante rouge, étoile	R4
C8	géante bleue, étoile	R4
C9	naine jaune, naine rouge	R4
C10	naine rouge, supergéante rouge	R4

à une substitution près). Nous présentons les résultats dans le tableau 7.9, dont la dernière colonne présente le nombre de regroupements dont les intensions comportent au moins un attribut pertinent au sens défini plus haut.

Nous remarquons qu'aucun regroupement calculé n'est identique à la référence et qu'une petite quantité correspond à des sous-ensembles des regroupements de référence. L'analyse des intensions pour les 4 regroupements qui englobent des concepts de la référence montre qu'aucun de ceux-là n'est issu d'attributs jugés pertinents.

Le tableau 7.10 présente les regroupements calculés, en gris les regroupements issus d'attributs non pertinents. Nous remarquons que les regroupements calculés présentés (c'est-à-dire ceux qui sont strictement inclus dans les regroupements de référence) ne contiennent que deux éléments. Nous constatons aussi que certains regroupements de référence (R5, R6, R7) ne sont jamais approchés.

Bien que tous issus d'attributs non pertinents, nous présentons dans le tableau 7.11 les regroupements calculés qui incluent la référence, avec des éléments supplémentaires. Nous remarquons que seul R5 est représenté.

TABLEAU 7.11 : Regroupements calculés incluant des regroupements de la référence

	Regroupement	⊃
C11	**pulsar**, **trou noir**, étoile	R5
C12	comète, galaxie, naine blanche, planète, **pulsar**, **trou noir**, étoile	R5
C13	naine blanche, planète, **pulsar**, satellite naturel, **trou noir**, étoile	R5
C14	naine blanche, planète, **pulsar**, **trou noir**, étoile	R5

Nous avons également analysé les regroupements proches à une substitution près, présentés dans le tableau 7.12. En gris, toujours les regroupements issus d'attributs non pertinents. Nous remarquons que tous les regroupements proches à une substitution près contiennent seulement deux éléments, ce qui indique une capacité de regroupement limitée.

TABLEAU 7.12 : Regroupements calculés proches des références

	Regroupement	Proche
C15	naine noire, **trou noir**	R5
C16	quasar, **trou noir**	R5
C17	naine rouge, **pulsar**	R5
C18	satellite naturel, **trou noir**	R5
C19	astéroïde, **pulsar**	R5
C20	géante bleue, **trou noir**	R5
C21	galaxie, **trou noir**	R5
C22	comète, **pulsar**	R5
C23	**pulsar**, étoile	R5
C24	amas globulaire, **pulsar**	R5
C25	comète, **trou noir**	R5
C26	astéroïde, **trou noir**	R5
C27	nébuleuse, **supernova**	R6
C28	galaxie, **supernova**	R6
C29	**supernova**, étoile	R6
C30	amas stellaire, **supernova**	R6
C31	**naine noire**, trou noir	R7
C32	**naine blanche**, étoile	R7
C33	galaxie, **naine blanche**	R7
C34	**naine blanche**, naine brune	R7

Dans le treillis, les 24 concepts issus des termes sont déjà situés au niveau le plus spécifique du fait de la méthode de filtrage `termlistonly` sur les objets. Nous ne pouvons donc pas obtenir de spécialisation de ces termes comme dans l'ontologie de référence.

TABLEAU 7.13 : Regroupements de référence (avec les objets formels étendus)

	Regroupement
R1	astéroïde, comète, naine brune, planète, satellite naturel, planète tellurique, planète gazeuse
R2	planète tellurique, planète gazeuse, planète
R3	amas stellaire, amas globulaire, amas ouvert
R4	galaxie, amas stellaire, amas globulaire, amas ouvert, quasar, étoile binaire, galaxie, galaxie spirale, galaxie annulaire, galaxie lenticulaire, galaxie elliptique, galaxie irrégulière, galaxie spirale barrée, amas de galaxies
R5	étoile, géante rouge, géante bleue, naine jaune, naine rouge, supergéante rouge
R6	galaxie, galaxie spirale, galaxie annulaire, galaxie elliptique, galaxie irrégulière, galaxie lenticulaire, galaxie spirale barrée
R7	trou noir, pulsar, étoile à neutron, trou noir supermassif, trou noir stellaire
R8	trou noir, trou noir supermassif, trou noir stellaire
R9	supernova, nova
R10	nova, supernova, supernova à effondrement de cœur, supernova thermonucléaire
R11	naine blanche, naine noire
R12	nébuleuse planétaire, nébuleuse

La méthode de construction du contexte formel `termlistonly+expand` peut permettre de spécialiser les concepts initiaux, c'est ce que nous allons mesurer dans la section suivante.

7.4.3 Pour le treillis 2

L'ontologie de référence comporte des concepts de spécialisation des 24 termes/concepts initiaux. Nous voulons analyser si l'ajout des objets formels étendus permet d'améliorer la qualité des regroupements.

Comme référence, nous prenons comme pour l'expérience précédente, les éléments de l'ontologie qui se retrouvent dans les extensions des objets formels (tableau 7.13).

L'ACF produit les regroupements du tableau 7.14. Comme pour l'expérience précédente, la colonne « Quantité » indique le nombre de regroupements. La colonne « Dont pert. » indique parmi ceux-ci la quantité de regroupements dus à au moins un attribut pertinent.

Si nous mettons en regard le tableau de l'expérience précédente (tableau 7.9), nous constatons qu'il n'y a toujours aucun regroupement qui est identique à la référence, qu'il n'y a plus de regroupements qui englobent la référence et que la quantité de regroupements inclus dans la référence est divisée par 2. Le tableau 7.15 montre les regroupements calculés qui sont inclus dans la référence.

TABLEAU 7.14 : Pertinence des regroupements (avec les objets formels étendus)

	Quantité	Dont pert.
$C = R$	0	0
$C \subset R$	5	4
$C \supset R$	0	0
C proche R	12	9
$C \neq R$	94	

TABLEAU 7.15 : Regroupements $\subset$ référence (objets formels étendus)

	Regroupement	$\subset$
C1	astéroïde, comète	R1
C2	astéroïde, planète	R1
C3	amas ouvert, galaxie	R4
C4	naine jaune, naine rouge	R5
C5	naine rouge, supergéante rouge	R5

TABLEAU 7.16 : Regroupements proches de la référence (objets formels étendus)

	Regroupement	Proche
C6	amas globulaire, **amas globulaire de l'hémisphère nord**, amas stellaire	R3 (**amas ouvert**)
C7	**galaxie**, supernova	R9 (**nova**)
C8	supernova, **étoile massive**	R9 (**nova**)
C9	**nébuleuse planétaire (R9)**, **supernova (R13)**	R9 (**nova**), R13 (**nébuleuse**)
C10	supernova, **supernova célèbre des temps modernes**	R9 (**nova**)
C11	supernova, **étoile**	R9 (**nova**)
C12	supernova, **étoile massive**	R9 (**nova**)
C13	**amas stellaire**, supernova	R9 (**nova**)
C14	naine noire, **trou noir à moment cinétique faible**	R12 (**naine blanche**)
C15	**galaxie**, naine blanche	R12 (**naine noire**)
C16	naine blanche, **naine brune**	R12 (**naine noire**)
C17	nébuleuse, **étoile**	R13 (**nébuleuse planétaire**)

Comme pour l'expérience précédente, nous constatons à regret que nous n'obtenons que des regroupements avec deux éléments. *A contrario*, le tableau 7.16 montre que nous obtenons un regroupement proche (C6) qui contient plus de deux éléments. En gras, l'élément qui est interverti entre le regroupement calculé et le regroupement de référence.

7.5 Conclusion

Dans ce chapitre nous avons mesuré la qualité des treillis sur le domaine de l'astronomie selon deux axes : la pertinence des attributs et des concepts formels pour le domaine, le taux de généricité des attributs propres et la qualité des regroupements vis-à-vis d'une ontologie de référence. Dans nos treillis, environ la moitié des attributs formels sont pertinents pour le domaine. Parmi ceux-ci seulement la moitié sont génériques, ce qui est peu pour un corpus de nature encyclopédique. Il serait intéressant d'évaluer de manière symétrique le domaine de la cuisine et du droit, mais dans ce dernier cas la distinction entre attributs génériques et attributs spécifiques risque d'être ardue.

L'évaluation qualitative des regroupements produits par l'ACF donne des résultats plutôt décevants. Cependant, nous pouvons les nuancer par le fait que nous avons mené une évaluation stricte, dans le sens où un regroupement partiellement complet est considéré comme faux. Nous continuons à croire en l'utilité de l'ACF pour la COT : son utilisation apporte une vision globale du corpus et amène par la force des choses l'utilisateur à s'interroger sur les notions d'emplois générique ou spécifique. Les regroupements proposés par l'ACF, même s'ils sont le plus souvent incomplets, permettent *via* l'intension des concepts formels de réfléchir sur la raison pour laquelle ils ont été effectués. Est-ce une tournure linguistique, ou bien une notion à laquelle l'expert n'avait pas pensé ?

Chapitre 8

Questions d'évaluation

Sommaire

La question de l'évaluation d'un système informatique peut s'envisager sous deux aspects, l'un considère le système comme une boîte noire et propose de l'évaluer au sein d'un processus plus complexe, comme la construction d'une ontologie, l'autre se penche sur les composantes du système.

Dans la suite de ce chapitre, nous proposons trois évaluations en rapport avec notre système. Les deux premières traitent d'une évaluation technologique : le choix de l'extracteur de termes utilisé pour débuter l'analyse terminologique et le choix de l'ACF par rapport à une méthode distributionnelle. La deuxième partie de ce chapitre propose un plan pour une évaluation orientée utilisateur.

8.1 Évaluation technologique

Par évaluation technologique, nous entendons l'évaluation des composants de notre système, à savoir un extracteur de termes, un analyseur syntaxique et une méthode de regroupement. Nous n'avons pas évalué le choix de Syntex comme analyseur syntaxique, un peu hors du cadre de cette thèse. Nous nous en tenons aux bons résultats qu'il a obtenu lors de la campagne EASY [1].

8.1.1 Le choix d'un extracteur de termes

Lors de nos précédentes expériences, nous avons utilisé l'extracteur de termes YaTeA. Nous pouvons nous poser la question de savoir si cet extracteur était le plus adapté à

1. Voir http://w3.erss.univ-tlse2.fr/membres/bourigault/syntex.html

Tableau 8.1 : Les candidats-termes sur le plus gros corpus

	Termes	Temps	Mémoire
Système 1	184729	45min	3935Mo
Système 2	924	135min	?
Système 3	échec	échec	échec
Système 4	59467	?	?
Système 5	10563	?	?
Système 6	91779	61min	35020Mo
Système 7	échec	échec	échec
Système 8	89883	2min	197Mo
Système 9	2129	112min	1269Mo

notre tâche. Dans le cadre du programme Quaero [2], nous avons organisé une campagne d'évaluation des extracteurs de termes. Cette évaluation repose sur une mesure de pertinence graduelle présentée dans (Nazarenko *et al.*, 2009).

L'idée est que la comparaison entre la liste de termes et la référence ne devrait pas se faire de manière booléenne, afin de diminuer l'impact des choix effectués tant sur la référence que dans la conception des extracteurs de termes. Ces choix concernent notamment la prise en compte des variantes des termes : si l'on compare une référence qui contient beaucoup de variantes avec la sortie d'un extracteur de termes qui ne retourne que les formes canoniques, ses résultats seront pénalisés bien qu'ils puissent se révéler corrects d'un point de vue sémantique.

Lors de cette campagne, nous avons testé, en collaboration avec nos partenaires du projet Quaero, des extracteurs de termes état de l'art et des prototypes de recherche qui ne sont pas encore disponibles. Au total, neuf systèmes ont été comparés sur le domaine de la pharmacologie. Nous avions à notre disposition une référence construite manuellement, indépendamment du corpus, contenant environ 5 800 termes. Le corpus, composé de brevets européens de pharmacologie (en anglais), a été découpé en trois sous-corpus de taille croissante : 500 000 mots, 1 500 000 mots et 2 500 000 mots.

Les résultats de cette campagne ne sont, au moment de la rédaction, pas encore rendus publics, nous ne les présenterons donc pas en intégralité. Le tableau 8.1 présente les données quantitatives pour chaque système (nous les avons anonymisés) sur le plus gros corpus : le nombre de candidats-termes retrouvés, le temps et la mémoire utilisés par les systèmes [3]. Sur cette table, les lignes à l'arrière-plan bleu représentent les prototypes de recherche.

Nous constatons une très grande disparité dans la quantité des termes extraits. Certains systèmes n'ont pas réussi à traiter un corpus de cette taille (2 500 000 mots).

2. http://www.quaero.org
3. lorsque disponible, mesuré sur des machines usuelles du début 2010.

TABLEAU 8.2 : Évaluation des extracteurs de termes sur le plus gros corpus

	T-precision	T-rappel	T-F-mesure
Système 1	2,81%	47,91%	5,31%
Système 2	55,55%	6,56%	11,73%
Système 4	22,61%	55,81%	32,18%
Système 5	57,21%	30,28%	39,60%
Système 6	11,53%	60,67%	19,37%
Système 8	11,31%	41,42%	17,77%
Système 9	53,57%	13,85%	22%

Le tableau 8.2 présente les résultats pour les systèmes ayant réussi à traiter le corpus. Les informations contenues dans cette table sont la précision terminologique, le rappel terminologique et la f-mesure terminologique, à un seuil déterminé. Ces mesures représentent la même chose que les mesures de précision et rappel usuelles, mais en prenant en compte une pertinence graduée et seuillée, c'est-à-dire un « taux de tolérance » vis-à-vis de la référence. Pour les définitions mathématiques de ces mesures, se reporter à (Nazarenko *et al.*, 2009).

Nous constatons sur ce tableau que certains systèmes, comme le numéro 2, choisissent de proposer peu de termes mais de qualité, tandis que d'autres, comme le premier, font le pari de proposer énormément de termes. Nous remarquons, et c'est heureux, une nette augmentation des résultats des prototypes de recherche par rapport aux systèmes plus anciens. Ceci est dû aux efforts qui sont effectués concernant le filtrage de la liste de candidats-termes.

À la question : est-ce que YaTeA était l'extracteur le plus adapté à notre tâche, nous proposons une réponse nuancée. Notre approche aurait profité d'utiliser un extracteur qui privilégie la précision au détriment du rappel, c'est-à-dire qui fournit peu de termes de grande qualité, donc en ce sens *non*, YaTeA n'est clairement pas l'extracteur le plus adapté. Cependant, il ne faut pas perdre de vue que cette évaluation a été effectuée sur des corpus de langue anglaise, et que la majorité des systèmes ne fonctionnent pas sur le français. La campagne d'évaluation n'a porté que sur un seul domaine, avec une seule référence. Il n'est donc pas recommandé, au vu des résultats, de tirer des conclusions définitives sur l'utilité ou l'inutilité de YaTeA.

La section suivante présente une autre facette de l'évaluation en tentant de déterminer si l'ACF est bien la méthode la plus adaptée pour la COT, du point de vue de la qualité des regroupements proposés.

8.1.2 Évaluation de la méthode de regroupement : approche numérique vs approche symbolique

Le cœur de notre système est constitué d'un système de regroupement symbolique, l'analyse de concepts formels. Si notre choix a été motivé dans l'état de l'art, il serait intéressant de comparer notre système avec une approche distributionnelle qui prend en compte la terminologie. Cette approche a été proposée dans (Bannour, 2010; Bannour *et al.*, 2011).

Présentation de la méthode de rapprochement distributionnelle

Cette méthode distributionnelle permet de prendre en compte les termes. Elle s'appuie sur une analyse du contexte des termes au sein de la phrase. La méthode commence par une extraction terminologique à l'aide d'un extracteur de termes. Chaque terme se voit ensuite attribué un vecteur dont les composantes sont le poids de chaque nom et de chaque verbe au sein de la phrase [4]. Dans le cadre de cette expérience, la mesure de pondération utilisée est T-TEST (Gosset, 1908). Une fois les vecteurs construits le système mesure la distance entre deux termes (la mesure de similarité utilisée est le cosinus [5]) et peut alors utiliser une méthode de regroupement (ici le clustering ascendant hiérarchique). Il est intéressant de noter que cette méthode n'utilise pas d'analyse syntaxique, mais se contente d'une analyse de surface de la phrase.

Méthodologie d'évaluation

Les deux systèmes ont été utilisés sur le même corpus de l'astronomie, en utilisant l'ontologie de référence construite pour l'expérience présentée dans la section 7.4. Comme regroupements de référence, nous reprenons ceux du chapitre précédent, pour le deuxième treillis (voir le tableau 8.3). Nous cherchons à regrouper les termes qui composent la référence. Pour l'ACF, nous allons construire un treillis en utilisant la méthode `termlistonly`, munie des attributs étendus et de saturation. Nous n'utilisons pas les objets formels étendus, car cette fonctionnalité ne s'intègre pas dans la méthode de rapprochement distributionnelle. Comme liste de termes initiale, nous utilisons les termes qui composent la référence.

4. Le terme est décomposé en mots et ceux-ci sont pris en compte dans le calcul du vecteur.
5. La justification de la combinaison mesure de pondération / mesure de similarité se trouve dans le rapport de master.

TABLEAU 8.3: Regroupements de référence

	Regroupement
R1	astéroïde, comète, naine brune, planète, satellite naturel, planète tellurique, planète gazeuse
R2	planète tellurique, planète gazeuse, planète
R3	amas stellaire, amas globulaire, amas ouvert
R4	galaxie, amas stellaire, amas globulaire, amas ouvert, quasar, étoile binaire, galaxie, galaxie spirale, galaxie annulaire, galaxie lenticulaire, galaxie elliptique, galaxie irrégulière, galaxie spirale barrée, amas de galaxies
R5	étoile, géante rouge, géante bleue, naine jaune, naine rouge, supergéante rouge
R6	galaxie, galaxie spirale, galaxie annulaire, galaxie elliptique, galaxie irrégulière, galaxie lenticulaire, galaxie spirale barrée
R7	trou noir, pulsar, étoile à neutron, trou noir supermassif, trou noir stellaire
R8	trou noir, trou noir supermassif, trou noir stellaire
R9	supernova, nova
R10	nova, supernova, supernova à effondrement de cœur, supernova thermonucléaire
R11	naine blanche, naine noire
R12	nébuleuse planétaire, nébuleuse

TABLEAU 8.4: Pertinence des regroupements, ACF et analyse distributionnelle

	ACF	Distrib.
$C = R$	0	2
$C \subset R$	8	3
$C \supset R$	0	0
C proche R	11	1
$C \neq R$	50	3

Résultats

Notre système calcule 98 concepts formels, dont 69 ayant une extension comportant au moins deux éléments. Le système de rapprochement distributionnel utilise un algorithme de clustering hiérarchique ascendant : les algorithmes de ce type ne se terminent pas, ils finissent obligatoirement par regrouper tous les éléments. Nous avons choisi de conserver le meilleur clustering, c'est-à-dire celui pour lequel à la fois $C = R$ et $C \subset R$ sont maximaux et $C \neq R$ est minimal. Les résultats sont présentés dans le tableau 8.4.

Nous constatons à la lecture de ce tableau que l'analyse distributionnelle arrive à produire deux regroupements qui sont identiques à la référence : R3 et R11. L'ACF n'arrive pas à reconstituer totalement R3, il manque toujours `amas stellaire`. Cela s'explique par le fait que l'attribut formel qui sert à regrouper `amas globulaire` et

TABLEAU 8.5 : Impact de la prise en compte des relations de subsomption explicites

	ACF	Distrib.	ACF + hyper
$C = R$	0	2	1
$C \subset R$	8	3	10
$C \supset R$	0	0	1
C proche R	11	1	7
$C \neq R$	50	3	50

`amas ouvert` est un attribut de saturation, `être amas (SAT)`, qui n'est pas partagé par `amas stellaire`.

Nous remarquons que `amas stellaire` possède l'attribut `être amas ouvert`. C'est une relation d'hyperonymie explicite ! Nous allons construire un nouveau treillis en utilisant notre module de prise en charge des subsomptions explicites (voir la section 6.3.3) afin de mesurer son efficacité. Les résultats sont présentés dans le tableau 8.5.

Nous constatons que la prise en compte des relations de subsomption explicites améliore nettement la qualité des regroupements proposés par l'ACF, qui arrive à proposer un regroupement correct. Nous remarquons cependant que la sur-généralisation ($C \supset R$) augmente, ce qui n'est pas souhaitable car cela peut induire l'utilisateur en erreur. L'analyse du regroupement sur-généralisé, constitué de `géante bleue, géante rouge, naine jaune, nova, supernova, étoile`, pourrait en partie correspondre au concept `astre` de l'ontologie, donc ce n'est pas réellement une erreur, cela peut correspondre à un choix de modélisation différent.

L'ACF est connu pour la quantité de regroupements qu'il propose. Les résultats obtenus par une méthode d'analyse distributionnelle sont pour cette expérience supérieurs en termes de regroupements exacts. Cependant, il faut remarquer que nous avons sélectionné le meilleur des regroupements au vu de la référence, ce qui est impossible en son absence. La prise en compte des relations de subsomption explicites dans l'ACF permet d'améliorer la qualité des résultats, à condition que le corpus utilisé contienne lesdites relations. Cela est vrai pour notre corpus d'astronomie à cause de sa nature encyclopédique. Nous constatons en revanche une légère tendance à la sur-généralisation, qui s'est en fait révélée suggérer une autre modélisation possible.

La comparaison des deux approches est difficile à mener, même sur un petit corpus comme celui-ci. Il faut cependant noter que les deux approches sont bruitées, même si ce n'est pas de la même manière. Toute la question est alors de savoir ce que l'ontologue peut « faire avec » ce bruit. Est-ce que les regroupements lui apportent quelque chose qui mérite qu'il passe du temps à réduire le bruit dans les résultats ?

8.2 Évaluation orientée utilisateur

L'évaluation objective, si tant est que cela puisse exister, de l'utilité de notre application pour l'ontologue est une tâche ardue que nous n'avons pas eu le temps de mener à bien durant cette thèse. Cette section est consacrée à la spécification prévisionnelle de cette tâche.

La première question qu'il faut se poser concerne ce que l'on souhaite mesurer, c'est-à-dire la notion d'*utilité pour l'ontologue*. Nous choisissons de définir l'utilité de notre système comme la combinaison entre la qualité de l'ontologie construite et le temps qu'il aura fallu pour y parvenir.

Pour mesurer la qualité de l'ontologie il est nécessaire de disposer d'une ontologie de référence que nous choisissons, et cela inclut déjà une certaine dose d'arbitraire, de considérer comme étant l'étalon de la qualité maximale. Une fois la référence établie, il est nécessaire d'établir des mesures qui vont permettre de chiffrer la proximité entre l'ontologie à évaluer et la référence. Nous proposons d'utiliser celles présentées dans la section 8.1.2. Une autre approche, étudiée dans (Ben Abbès *et al.*, 2010), permet de synthétiser la notion de regroupements proches à l'aide d'une mesure unique. La mesure du temps pris pour construire l'ontologie avec et sans notre système est une tâche qu'il est impossible d'évaluer de manière totalement objective. En effet si la même personne peut construire deux fois de suite la même ontologie il est certain que lors du deuxième essai elle ira plus vite. Il est donc nécessaire de mener deux tâches de construction avec deux groupes d'ontologues comparables, à la fois dans leur connaissance du domaine en question et dans leur expérience en construction d'ontologie ; nous pouvons envisager des étudiants de master par exemple.

Comme nous l'avons mentionné plus haut l'utilité de notre système se traduirait dans l'idéal par une augmentation de la qualité de l'ontologie produite (donc une plus grande adéquation à la référence) et une diminution du temps nécessaire pour la construire. Mais nous pouvons envisager plusieurs autres scénarios plus ou moins favorables :

- La qualité de l'ontologie produite diminue, le temps passé à la construire augmente : cela traduit que notre système dessert l'ontologue, ce qui pourrait s'expliquer soit par l'importance du bruit dans les résultats, soit par une ergonomie inadaptée.
- La qualité de l'ontologie diminue, mais le temps passé reste constant ou diminue : cela signifie que notre système induit l'ontologue en erreur en lui présentant une majorité de résultats erronés ou incomplets qu'il considère au pied de la lettre.
- La qualité de l'ontologie augmente, mais le temps passé augmente également : notre système a aidé l'ontologue à modéliser des connaissances auxquelles il n'avait pas pensé.
- La qualité de l'ontologie reste constante, mais le temps diminue : cela prouve que notre système automatise certaines parties du travail de l'ontologue, sans influencer son jugement.

De manière concrète, nous projetons de conduire, dans le cadre du programme Quaero, une nouvelle évaluation de l'acquisition d'ontologies avec l'aide d'une vingtaine d'étudiants ayant reçu une formation sur Terminae. Le domaine serait déterminé en accord avec leur taux d'expertise. Nous constituerions manuellement un corpus et une ontologie de référence (sur le domaine de l'astronomie, ces deux éléments sont déjà disponibles), puis nous répartirions les étudiants en deux groupes. Le premier groupe construirait l'ontologie en utilisant Terminae. Le second groupe aurait à sa disposition les regroupements de termes proposés par notre système. Cette liste ainsi que l'intension des concepts formels seraient intégrées dans les fiches terminologiques. Le fait d'intégrer les résultats de l'ACF directement dans l'interface de Terminae devrait permettre de minimiser le facteur ergonomique dans l'évaluation. Nous fournirions tout de même notre système en parallèle, et nous mesurerions comment les étudiants l'utilisent.

8.3 Conclusion

Dans ce chapitre nous avons présenté l'évaluation de notre système sous deux angles : l'évaluation technologique des composantes et l'évaluation orientée utilisateur. L'évaluation technologique nous apprend que YaTeA n'est pas, tout au moins dans sa version actuelle, l'extracteur de termes le plus adapté pour notre système : il produit beaucoup trop de candidats-termes, cela ne permet pas à notre variante de construction du contexte formel `termlistonly` d'exprimer tout son potentiel. Un filtrage plus poussé des résultats est d'ores et déjà intégré dans certains prototypes de recherche. Nous avons éprouvé nos choix de l'ACF et d'un contexte syntaxique en les comparant à une méthode de clustering hiérarchique munie d'un contexte de co-occurrences. Les résultats montrent que l'ACF reste inférieur en termes de qualité des regroupements. Ces résultats sont à nuancer par le problème inhérent au clustering hiérarchique, à savoir décider à quel moment l'algorithme se termine. De plus, l'ACF permet rapidement de retrouver les raisons qui ont conduit au rapprochement (l'intension). Le choix d'un contexte syntaxique apporte des informations bien plus intelligibles dans l'intension des concepts formels qu'un contexte des co-occurrences. Dans la dernière partie de ce chapitre nous avons présenté un plan d'évaluation orientée utilisateur qu'il reste à mettre en œuvre.

Chapitre 9

Conclusions et perspectives

Dans cette thèse, nous avons combiné une approche terminologique de construction d'ontologies à partir de textes avec une méthode de regroupement symbolique, l'analyse de concepts formels. Cette association a pour objectif de rendre moins fastidieuse la tâche de conceptualisation, c'est-à-dire l'identification des concepts clés d'un domaine et l'explicitation de leurs caractéristiques.

Dans notre premier système, nous combinons les termes et les mots avec un processus d'ACF reposant sur un contexte syntaxique. Cette approche est originale par rapport à l'état de l'art, même si des systèmes prenaient en compte les termes *ou* les mots. Nos premières expérimentations, sur trois corpus différents, mettent en exergue les problèmes inhérents à l'application de l'ACF sur des données textuelles : la sensibilité au corpus, la quantité de concepts formels générés et l'interprétation en ontologie.

Nous avons apporté des solutions à chacun de ces problèmes. Pour la sensibilité au corpus, nous proposons de la détecter au plus tôt dans la chaîne de traitement, à l'aide d'indicateurs de surface. Nous en avons proposé plusieurs, qui permettaient de rendre compte du contraste observé dans les résultats obtenus sur les deux premiers corpus. Nous nous sommes ensuite aperçu qu'ils ne rendaient pas bien compte du comportement de l'ACF sur un troisième corpus. Cela montre qu'il faut sans doute aller plus loin dans la caractérisation de la nature des corpus, mais nous ne pensons pas que cela mette en cause la faisabilité de la détection de la sensibilité *a priori*.

Nous avons proposé plusieurs approches de filtrage, pour contenir l'explosion de la quantité de concepts formels : au moment de la construction du contexte formel, par une sélection terminologique des éléments importants, et une fois le contexte construit. La question cruciale reste celle de la transformation du treillis en ontologie. Nous avons proposé un paramétrage de l'ACF qui permet d'obtenir un treillis reflétant plus fidèlement la connaissance contenue dans les textes.

Les expérimentations menées à l'aide de notre système amélioré ont permis d'expliquer pourquoi la traduction directe en ontologie fonctionne sur certaines expériences de l'état de l'art, mais donne de mauvais résultats avec une approche terminologique. C'est la nature des éléments composant le contexte formel qui est en cause. Si les objets du contexte formel font référence à des objets précis du domaine modélisé, comme des entités nommées, l'interprétation ontologique du treillis est valide. En revanche, lorsque les termes évoquent à la fois des objets précis et des concepts, cette interprétation introduit des incohérences, parce qu'on tend à mélanger les attributs qui traduisent des propriétés accidentelles des concepts et ceux qui traduisent des propriétés intrinsèques, plus pertinentes d'un point de vue ontologique.

Nous proposons de bien distinguer les attributs génériques et spécifiques lors de la construction du treillis. Une interface de validation vient épauler le travail manuel de séparation des attributs formels.

Nous avons proposé une évaluation technologique de notre système. Les résultats montrent que l'extracteur de termes utilisé n'est pas le plus adapté à notre tâche mais le problème de quantité de concepts formels qu'il a permis de mettre en évidence reste réel. Une première évaluation du choix de l'ACF comme méthode de regroupement a été faite par comparaison avec une approche distributionnelle adaptée à l'analyse terminologique. Les résultats sont difficiles à interpréter, mais ne semblent pas clairement favorables à l'ACF. Les résultats de l'ACF sont bruités, mais les autres approches le sont aussi : cela prouve que la COT ne peut pas s'appuyer sur une méthode entièrement automatique de regroupement, et qu'il faut intégrer une interaction avec l'utilisateur dans le processus de conceptualisation.

Nous pensons que c'est dans ce processus interactif que l'ACF combiné à une approche terminologique trouve son intérêt. Booléenne par nature, elle offre une vue compacte du corpus ; la prise en compte des termes se fait très naturellement. Les regroupements des extensions sont documentés par les propriétés intensionnelles, ce qui en facilite l'appropriation, la correction et l'interprétation par l'ontologue. L'évaluation orientée utilisateur dont nous avons jeté les bases dans le dernier chapitre doit permettre de confirmer cet intérêt de l'ACF dans le processus interactif de COT.

D'un point de vue technique, nous prévoyons d'intégrer notre système, sous forme de greffon, à la nouvelle version en cours de développement de Terminae (Biebow et Szulman, 1999; Aussenac-Gilles *et al.*, 2008) ou à Dafoe (Szulman *et al.*, 2009). À ce niveau la principale difficulté est ergonomique et consiste à déterminer comment présenter les résultats de l'ACF à l'utilisateur, pour l'aider dans son travail de conceptualisation sans le surcharger de données bruitées. À l'issue de l'évaluation orientée utilisateurs, il faudra donc mener une analyse ergonomique. Une des questions consiste à savoir si la représentation en treillis est la plus pertinente [1], ou si la consultation des regroupements suffit à l'ontologue.

D'un point de vue recherche, plusieurs pistes s'offrent à nous pour améliorer la fidélité du contexte formel à l'égard du texte. À plus court terme, il est nécessaire d'effectuer une caractérisation plus fine de la nature des corpus pour pouvoir exhiber des indicateurs fiables de prévision du nombre de concepts formels. Nos premières expériences semblent montrer que les indicateurs à utiliser sont eux-mêmes dépendants du type de corpus. Nous envisageons donc de proposer deux types d'indicateurs : le premier ensemble permettrait de caractériser le corpus et le second, dont le choix dépendrait du premier, donnerait une première estimation du nombre de concepts formels. Ces indicateurs doivent guider la constitution du corpus et le paramétrage de l'ACF.

1. Dans ce cas, de nouvelles stratégies de visualisation comme le treillis Iceberg (Stumme *et al.*, 2002) seraient à explorer.

À moyen terme, nous pourrions améliorer notre prise en charge de la négation. Actuellement, nous proposons à l'utilisateur une caractérisation booléenne des attributs, ils sont soit positifs soit négatifs. Les énoncés négatifs ne le sont pas forcément totalement. Présenter des attributs formels partiellement négatifs à l'utilisateur permettrait de mieux respecter le texte. Il est nécessaire pour cela de commencer par caractériser finement la négation et d'exhiber des marqueurs textuels.

Notre prise en charge de la subsomption explicitement mentionnée dans les textes gagnerait à être améliorée. Notre système ne traite que du cas particulier *NOM est un NOM*, or il existe d'autres tournures linguistiques qui marquent une hyperonymie (par exemple « les tulipes, les coquelicots et autres fleurs ») qui mériteraient d'être prises en compte (Morin, 1999; Séguéla et Aussenac-Gilles, 1999). Les attributs formels de saturation comportent uniquement deux niveaux : le verbe suivi de son complément prépositionnel, et le verbe seul. Nous pourrions introduire un nombre indéfini de niveaux de saturation en tenant compte des différents adjectifs qui constituent un attribut formel étendu, afin de minimiser les cas d'ambiguïté qui se manifestent sur les attributs de saturation de niveau 2.

Le plus gros travail de recherche pour améliorer la fidélité du contexte formel est la différenciation des énoncés génériques *vs.* spécifiques. Si nous arrivons à clarifier cette distinction, nous pourrons construire un treillis dont les intensions des concepts formels contiennent uniquement des propriétés sur les concepts évoqués par leur extension. Nous avons l'intuition que la traduction automatique en logique de description deviendrait alors possible. Cependant, faire cette distinction automatiquement est loin d'être trivial. Si certaines tournures comme *généralement* ou *cette étoile* indiquent clairement des énoncés génériques ou spécifiques, il existe de nombreux autres marqueurs beaucoup moins fiables. C'est souvent la combinaison de facteurs qui détermine la lecture d'un énoncé. Pour mener à bien cette tâche, une caractérisation des énoncés génériques ou spécifiques est nécessaire. Demander à l'utilisateur de la faire à la main est une possibilité, mais c'est peut-être trop exiger de lui dans le contexte de la construction d'ontologies. Nous pourrions chercher à dégrossir le travail par des méthodes de classification des énoncés, à l'aide d'apprentissage supervisé. Cette étude reste à faire.

À plus long terme, il serait intéressant de prolonger cette analyse sur une extension de l'ACF, l'analyse de concepts relationnels. Son intégration permettrait d'automatiser davantage le processus de modélisation en proposant à l'utilisateur des relations ontologiques entre les concepts. Il est probable que là aussi, il faille une analyse des corpus et un paramétrage de la méthode très précis afin d'obtenir des résultats pertinents pour la construction d'ontologies.

Annexe A

Extraction du contexte formel

Sommaire

Le contexte formel est la clé de voûte de notre système. Ce chapitre détaille notre système de construction du contexte formel à partir d'une analyse syntaxique en dépendance des textes. Cette annexe présente au lecteur la documentation du programme afin qu'il se familiarise avec l'ensemble des possibilités qui lui sont offertes.

A.1 Page du manuel de notre système

Cette section retranscrit la page du manuel correspondant à notre système. Elle a pour vocation de permettre au lecteur d'appréhender en première lecture l'ensemble des possibilités offertes par notre programme.

extraire_contexte_formel - construction de contexte formel pour l'ACF

A.1.1 SYNOPSIS

extraire_contexte_formel [options] fichier-syntex > fichier_sortie.oal

A.1.2 DESCRIPTION

Construire un contexte formel à partir des sorties de syntex en xml. Génère un fichier au format oal (chaque ligne = objet1 :attribut1 ;attribut2 ;attribut3). Le contexte formel est constitué des sujets / compléments d'objet comme objets formels et des verbes liés comme attributs. Le système gère les pronoms relatifs, les conjonctions de coordination,

les verbes d'état, les auxiliaires, les verbes avec prépositions, les verbes avec pronoms et la négation d'attributs. Le système distingue les verbes avec prépositions, pour lesquels il extrait directement le complément d'objet via l'expansion maximale de l'attribut, des verbes sans préposition qui ne font pas l'objet d'une extraction étendue, sauf avec l'option cross. Le système peut prendre en entrée une liste de termes, dans ce cas il appariera les objets formels avec ces derniers, en ne conservant que ceux-ci avec l'option termlistonly. Une liste de mots vides peut être fournie en entrée, elle s'applique sur les objets et les attributs. L'option "depth" permet de construire le treillis en plusieurs étapes, en considérant à chaque étape les attributs du sujet comme nouveaux éléments à analyser –uniquement en conjonction avec termlistonly–.

A.1.3 OPTIONS

-help

Affiche de l'aide

-depth n

Construction incrémentale du treillis en n passes, chaque passe ajoutant les nouveaux objets découverts.

SELECTION ET FILTRAGE SUR LES OBJETS

-stoplist

Fichier qui contient une liste de mots vides lemmatisés, un par ligne. Les mots vides ne seront pas pris en compte dans le contexte, qu'ils soient objet ou attribut. L'utilisation de cette option entraîne également par défaut la suppression des objets ou attributs qui contiennent un chiffre. Pour inhiber ce comportement, ajouter "-nodropnumbers"

-nodropnumbers

Empêche la suppression systématique des objets ou attributs qui contiennent des chiffres.

-namesonly

Si ce commutateur est spécifié, les sujets qui ne sont pas des noms communs ou des noms propres sont supprimés et les attributs qui ne sont pas des verbes aussi (ces derniers devraient être rares grâce au moteur de détection des propositions relatives et des conjonctions). Cela a comme conséquence que les pronoms anaphoriques (il, je...) sont ignorés.

-sujonly

N'inclure dans le contexte que les sujets.

-objonly

N'inclure dans le contexte que les COD / COI.

-termlist

Fichier qui contient une liste de termes, un par ligne sur deux colonnes séparées par ;. La première colonne est la forme fléchie et la seconde le lemme. Si ce commutateur est spécifié, le contexte sera construit en utilisant les termes lemmatisés. La détection se fait sur les formes fléchies du texte, donc bien toutes les spécifier.

-termlistonly

Si ce commutateur est spécifié, le contexte sera construit à partir des termes lemmatisés uniquement.

-expand

Tente de faire une expansion en syntagme sur les objets du contexte formel. Attention ce commutateur modifie la sémantique de termlistonly : les termes trouvés sont étendus et inclus dans le contexte. Dans ce cas, termlistonly + expand est à considérer comme des "graines" terminologiques qui permettent de contenir la taille du treillis.

FILTRAGE SPECIFIQUE AUX ATTRIBUTS

-min_attr

Nombre minimum d'attributs que doit posséder un objet pour être inclus dans le contexte, par défaut 1.

-noexpandattr

Ne tente pas d'étendre les attributs en syntagmes maximaux.

-noprep

Ne pas tenter d'inclure dans le contexte les formes étendues des prépositions des verbes (ex. "conduire => conduire sur belle chaussée")

-nopro

Ne pas tenter d'inclure dans le contexte les pronoms associés aux verbes (ex. "engager = s'engager")

-nosubjattr

Ne pas tenter d'inclure dans le contexte les attributs du sujet (ex. "Jean est un beau jeune homme" => "Jean :être beau jeune homme")

-sat

Ajoute les formes les plus génériques des attributs (sans préposition étendue et sans préposition) aux objets. Les attributs de saturation avec préposition sont suffixés de (SAT). Les attributs de saturation sans préposition sont suffixés de (SAT2). Les attributs de saturation qui ne sont pas regroupant (qui n'existent que dans un seul objet formel) sont supprimés. Si l'on utilise conjointement l'option cross, les attributs sans les objets sont suffixés de (SAT).

-negtag

Détection et marquage de la négation sur les attributs (si des adverbes de négation portent sur le verbe attribut formel ou sur un adjectif résultant de l'expansion en syntagme) et sur les objets en utilisation conjointe avec -cross. Les attributs négatifs sont préfixés de "(NEG)". Utilise comme dictionnaire les adverbes "ne", "pas", "non", "point", "nullement", "aucunement", "aucun", "nul", "rien", "jamais", "pas un", "personne", "sans".

-negdrop

Suppression de la négation. Implique -negtag.

-cross

Effectue une recherche étendue aux objets pour les sujets, afin d'avoir l'attribut le plus étendu possible, pour les verbes sans préposition et qui ne sont pas des verbes d'état. Par exemple, "Jean aime Marie" => "Jean :aimer Marie". Si on combine cette option avec l'option "sat", ajoute aussi "Jean :aimer (SAT) ;aimer Marie".

AUTRES OPTIONS

-debug

Mode verbeux pour débogger.

-allsent

Ajoute uniquement la phrase complète au contexte pour chaque attribut. Doit être mis en parallèle avec le treillis original.

A.1.4 ARGUMENTS

En argument une sortie de syntex français en XML. Tous les fichiers doivent être encodés en latin 1.

A.1.5 EXEMPLES

```
ecf.pl test.xml > test.oal
```

Construire le contexte formel constitué de tous les sujets et compléments d'objet, sous forme simple (les mots pivots) comme objets formels et de leurs groupes verbaux étendus associés et l'écrit dans test.oal.

```
ecf.pl -stoplist mots_vides.txt -termlist termes.txt test.xml
```

Construit le contexte formel en supprimant les objets formels et les attributs formels qui appartiennent exactement à une liste de mots vides. Supprime également les objets

et les attributs formels qui contiennent des chiffres. Tente d'apparier les objets formels avec les termes fournis dans termes.txt. Si l'appariement réussit, remplace les objets formels mots par le terme. Les objets formels dont l'appariement n'est pas concluant sont inclus tels quels dans le contexte.

```
ecf.pl -sat test.xml
```

Construire le contexte formel constitué de tous les sujets et compléments d'objet, sous forme simple (les mots pivots) comme objets formels et de leurs groupes verbaux étendus associés. Des attributs formels de saturation sont ajoutés lorsqu'ils apportent de l'information au contexte formel. Les attributs de saturation de premier niveau (correspondant à la forme VERBE + PREPOSITION) sont suffixés de "SAT" tandis que les attributs de saturation de second niveau, qui correspondent à la forme verbale la plus simple, sont suffixés de "SAT2".

```
ecf.pl -expand -sat -namesonly test.xml
```

Construire le contexte formel dont les objets formels sont étendus en groupes nominaux maximaux à l'aide d'une analyse syntaxique. Les attributs formels incluent les attributs de saturation. Les objets formels sont constitués uniquement de noms communs ou de noms propres, les pronoms personnels anaphoriques sont rejetés.

```
ecf.pl -termlist termes.txt -termlistonly test.xml
```

Construire le contexte formel en ne conservant comme objets formels que les termes de la liste "termes.txt.lem".

```
ecf.pl -termlist termes.txt -termlistonly -expand test.xml
```

Construire le contexte formel en ne conservant *au départ* que les objets formels qui appartiennent à la liste de termes, mais en tentant aussi de les étendre par une approche syntaxique. Le groupe nominal maximum est conservé comme objet formel.

Bibliographie

Hans Akkermans, Frank van Harmelen, Guus Schreiber et Bob Wielinga : *A formalization of knowledge-level models for knowledge acquisition*, pages 169–208. John Wiley & Sons, Inc., New York, NY, USA, 1993. ISBN 0-471-59368-0.

Sophie Aubin et Thierry Hamon : Improving term extraction with terminological resources. *Dans* Tapio Salakoski, Filip Ginter, Sampo Pyysalo et Tapio Pahikkala, éditeurs : *Advances in Natural Language Processing 5th International Conference on NLP, FinTAL 2006*, pages 380–387, Turku, Finland, 2006. Springer.

Nathalie Aussenac-Gilles, Sylvie Despres et Sylvie Szulman : The Terminae method and platform for ontology engineering from texts. *Dans Proceeding of the 2008 conference on Ontology Learning and Population : Bridging the Gap between Text and Knowledge*, pages 199–223. IOS press, 2008.

Nathalie Aussenac-Gilles et Marie-Paule Jacques : Designing and evaluating patterns for relation acquisition from texts with Caméléon. *Terminology*, 14(1):45–73, 2008.

Sondès Bannour : étude de la similarité distributionnelle entre termes et du clustering. Mémoire de D.E.A., Université de la Manouba (RIADI) / Université Paris 13 (LIPN), juin 2010.

Sondès Bannour, Laurent Audibert et Adeline Nazarenko : Mesures de similarité distributionnelle entre termes. *Dans Actes des 22èmes Journées francophones d'Ingénierie des Connaissances (IC2011)*, Chambéry, France, mai 2011.

Sarra Ben Abbès, Haïfa Zargayouna et Adeline Nazarenko : Évaluation de classes sémantiques pour la construction d'ontologies. *Dans* Sylvie DESPRÉS, éditeur : *Actes des 21èmes Journées francophones d'Ingénierie des Connaissances (IC2010)*, pages 297–308, Nîmes, France, 2010. Ecole des Mines d'Alès. URL http://hal.archives-ouvertes.fr/hal-00487726/PDF/ic2010_submission_27-1.pdf. Quaero.

Rokia Bendaoud, Mohamed Rouane Hacene, Yannick Toussaint, Bertrand Delecroix et Amédéo Napoli : Construction d'une ontologie à partir d'un corpus de textes avec l'ACF. *Dans* Frankie Trichet, éditeur : *Actes des 18èmes Journées francophones d'Ingénierie des Connaissances (IC2007)*. Cépaduès, 2007.

Rokia Bendaoud, Yannick Toussaint et Amédéo Napoli : Pactole : A methodology and a system for semi-automatically enriching an ontology from a collection of texts. *Dans* Peter W. Eklund et Ollivier Haemmerlé, éditeurs : *Proceedings of the 16th International Conference on Conceptual Structures (ICCS 2008)*, volume 5113 de *Lecture Notes in Computer Science*, pages 203–216. Springer, 2008. ISBN 978-3-540-70595-6.

Matthew Berland et Eugene Charniak : Finding parts in very large corpora. *Dans Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 57–64. Association for Computational Linguistics, 1999.

Brigitte Biebow et Sylvie Szulman : TERMINAE : A linguistics-based tool for the building of a domain ontology. *Knowledge Acquisition, Modeling and Management*, 1621:49–66, 1999.

Willem Nico Borst : *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Thèse de doctorat, Dutch research school for Information and Knowledge Systems (SIKS), Universiteit Twente, 1997.

Didier Bourigault : Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. *Dans Actes des 9emes journées sur le Traitement Automatique des Langues Naturelles*, pages 75–84, 2002.

Didier Bourigault et Cécile Fabre : Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de grammaire*, 25:131–151, 2000.

Didier Bourigault, Cécile Fabre, Cécile Frérot, Marie-Paule Jacques et Sylwia Ozdowska : Syntex, analyseur syntaxique de corpus. *Dans Actes des 12emes journées sur le Traitement Automatique des Langues Naturelles*, Dourdan, France, 2005.

Ronald J. Brachman et James G. Schmolze : An overview of the KL-ONE knowledge representation system. *Cognitive Science : A Multidisciplinary Journal*, 9 (2):171–216, 1985.

Xiaoquan Chu : *Les verbes modaux du français*. Ophrys, L'essentiel Français, 2008.

Philipp Cimiano : *Ontology Learning and Population from Text : Algorithms, Evaluation and Applications*. Springer, New York, 2006. ISBN 978-0-387-30632-2.

Philipp Cimiano, Andreas Hotho et Steffen Staab : Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research (JAIR)*, 24:305–339, août 2005.

Philipp Cimiano et Johanna Völker : Text2Onto — a framework for ontology learning and data-driven change discovery. *Dans* Andres Montoyo, Rafael Munoz et Elisabeth Metais, éditeurs : *Proceedings of the 10th International*

Conference on Applications of Natural Language to Information Systems (NLDB), volume 3513 de *Lecture Notes in Computer Science*, pages 227–238, Alicante, Spain, juin 2005. Springer.

Philipp Cimiano, Johanna Völker et Rudi Studer : Ontologies on demand ? — a description of the state-of-the-art, applications, challenges and trends for ontology learning from text. *Information, Wissenschaft und Praxis*, 57(6-7):315–320, octobre 2006.

William J. Clancey et Reed Letsinger : Neomycin : Reconfiguring a rule-based expert system for application to teaching. *Dans* Patrick J. Hayes, éditeur : *IJCAI*, pages 829–836. William Kaufmann, 1981. URL http://dblp.uni-trier.de/db/conf/ijcai/ijcai81.html#ClanceyL81.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva et Valentin Tablan : GATE : A framework and graphical development environment for robust NLP tools and applications. *Dans Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

Béatrice Daille : Conceptual structuring through term variations. *Dans* F. Bond, A. Korhonen, D. MacCarthy et A. Villacicencio, éditeurs : *Proceedings of ACL 2003 Workshop on Multiword Expressions : Analysis, Acquisition and Treatment*, pages 9–16, 2003.

Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas et Richard Harshman : Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

Maud Ehrmann : *Les entités nommées, de la linguistique au TAL : statut théorique et méthodes de désambiguïsation*. Thèse de doctorat, Université Paris 7, 2008.

David Faure et Claire Nedellec : Knowledge acquisition of predicate argument structures from technical texts using machine learning : the system asium. *Dans* D. Fensel et R. Stude, éditeurs : *Proceedings of the 11th International Conference on Knowledge Engineering and Knowledge Management*, pages 329–334. Springer-Verlag, mai 1999.

Edward A. Feigenbaum, Bruce G. Buchanan et Joshua Lederberg : On generality and problem solving : A case study using the DENDRAL program. Rapport technique, Stanford University, 1970.

Edward A. Feigenbaum et Pamela MacCorduck : *The fifth generation : artificial intelligence and Japan's computer challenge to the world*. Addison-Wesley Pub. Co.,Reading, MA, 1983.

Mariano Fernandez-López et Natalia Juristo : Methontology : from ontological art towards ontological engineering. *Dans Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, Stanford, USA, mars 1997.

Karën Fort, Maud Ehrmann et Adeline Nazarenko : Vers une méthodologie d'annotation des entités nommées en corpus ? *Dans Actes de la 16ème Conférence sur le Traitement Automatique des Langues Naturelles 2009*, Senlis, France, 2009. URL http://hal.archives-ouvertes.fr/hal-00402321/en/. Quaero.

Blaz Fortuna, Marko Grobelnik et Dunja Mladenic : Semi-automatic data driven ontology construction system. *Dans Proceedings of the 9th International multi-conference Information Society IS-2006, Ljubljana, Slovenia*, 2006.

Bernhard Ganter et Rudolf Wille : *Formal Concept Analysis*. Springer, Berlin, 1999.

William Sealy Gosset : The probable error of a mean. *Biometrika*, 1908.

Thomas R. Gruber : A translation approach to portable ontology specifications. *Knowledge acquisition*, 5:199, 1993.

Nicola Guarino : Understanding, building and using ontologies. *International Journal of Human Computer Studies*, 46:293–310, 1997.

Benoît Habert, Adeline Nazarenko et André Salem : *Les linguistiques de corpus*. Armand Colin, Paris, 1997.

Zellig Harris, Michael Gottfried, Thomas Ryckman, Paul Jr Mattick, Anne Daladier, T.N. Harris et S. Harris : *The Form of Information in Science, Analysis of Immunology Sublanguage*, volume 104 de *Boston Studies in the Philosophy of Science*. Kluwer Academic Publisher, Boston, 1989.

John Anthony Hartigan et M. A. Wong : Algorithm AS 136 : A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

Marti A. Hearst : Automatic acquisition of hyponyms from large text corpora. *Dans Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

Donald Hindle : Noun classification from predicate-argument structures. *Dans Proceedings of the Association for Computational Linguistics*, pages 268–275, 1990.

Christian Jacquemin et Didier Bourigault : *Term extraction and automatic indexing*, pages 599–615. Oxford University Press, USA, 2003.

Marie-Paule Jacques et Nathalie Aussenac-Gilles : Variabilité des performances des outils de TAL et genre textuel. *Traitement Automatique des Langues*, 47 (1):11–32, 2006.

Daniel Kayser : *La représentation des connaissances*. Hermès, 1997. ISBN 2866016475.

Guiraude Lame : *Construction d'ontologie à partir de textes. Une ontologie du droit dédiée à la recherche d'information sur le Web*. Thèse de doctorat, Ecole des Mines de Paris, Paris, France, décembre 2002.

Mariano Fernández Lopez, Asunción Gómez-Pérez, Juan Pazos Sierra et Alejandro Pazos Sierra : Building a chemical ontology using methontology and the ontology design environment. *Intelligent Systems and their Applications, IEEE*, 14(1):37–46, 2002.

Robert MacGregor : The evolving technology of classification-based knowledge representation systems. *Principles of Semantic Networks : Explorations in the representation of knowledge*, 1:385–400, 1991.

Alexander Maedche et Steffen Staab : Discovering conceptual relations from text. *Dans ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, 2000*. IOS Press, Amsterdam, 2000.

Bernardo Magnini, Emanuele Pianta, Octavian Popescu et Manuela Speranza : Ontology population from textual mentions : Task definition and benchmark. *Dans Proceedings of the 2nd Workshop on Ontology Learning and Population : Bridging the Gap between Text and Knowledge*, pages 26–32, Sydney, Australia, juillet 2006. Association for Computational Linguistics.

George A. Miller : WordNet : a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Thibault Mondary et Sylvie Després : Analyse de concepts formels pour la construction d'ontologies à partir de textes : la question du corpus. *Dans* Nicolas Laliche, Agnès Braud, Pierre Gançarski et Jean-Gabriel Ganascia, éditeurs : *Actes des ateliers Qualité des Données et des Connaissances, atelier associé à EGC09*, pages A6 :5–13, Strasbourg France, janvier 2009. URL http://hal.archives-ouvertes.fr/hal-00357119/en/.

Thibault Mondary, Sylvie Després, Adeline Nazarenko et Sylvie Szulman : Construction d'ontologies à partir de textes : la phase de conceptualisation. *Dans Actes des 19èmes Journées francophones d'Ingénierie des Connaissances (IC2008)*, pages 87–98, Nancy, France, juin 2008. URL http://hal.archives-ouvertes.fr/hal-00289613/en/.

Emmanuel Morin : Des patrons lexico-syntaxiques pour aider au dépouillement terminologique. *Traitement Automatique des Langues*, 40(1):143–166, 1999.

Charles Muller : *Principes et méthodes de statistique lexicale*. Hachette, 1977. ISBN 2010042891.

Adeline NAZARENKO : *Donner accès au contenu des documents textuels*. Habilitation à diriger des recherches (hdr), Université Paris Nord, 2004.

Adeline NAZARENKO, Haïfa ZARGAYOUNA, Olivier HAMON et Jonathan VAN PUYMBROUCK : Evaluation des outils terminologiques : enjeux, diffcultés et propositions. *Traitement Automatique des Langues*, 50(1 varia):257–281, 2009.

Daniel OBERLE, Raphael VOLZ, Boris MOTIK et Steffen STAAB : An extensible ontology software environment. *Dans* Steffen STAAB et Rudi STUDER, éditeurs : *Handbook on Ontologies*, International Handbooks on Information Systems, pages 311–333. Springer, 2004.

Charles Kay OGDEN et Ivor Armstrong RICHARDS : *The Meaning of Meaning : A Study of the Influence of Language Upon Thought and of the Science of Symbolism*. Harcourt, New York, 1923. ISBN 0156584468.

Nouha OMRANE, Adeline NAZARENKO et Sylvie SZULMAN : Combining terms and named entities for modeling domain ontologies from texts. *Dans 17th International Conference EKAW 2010, Lisbon, Portugal, Demo and Poster Abstracts*, pages 5–6, octobre 2010.

Nouha OMRANE, Adeline NAZARENKO et Sylvie SZULMAN : Les entités nommées : des clés linguistiques pour la conceptualisation. *Dans Actes des 22èmes Journées francophones d'Ingénierie des Connaissances (IC2011)*, Chambéry, France, mai 2011.

Patrick PANTEL et Marco PENNACCHIOTTI : Espresso : leveraging generic patterns for automatically harvesting semantic relations. *Dans ACL '06 : Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 113–120, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

Peter F. PATEL-SCHNEIDER, Deborah L. MCGUINNESS, Ronald J. BRACHMAN et Lori A. RESNICK : The classic knowledge representation system : Guiding principles and implementation rationale. *ACM SIGART Bulletin*, 2(3):108–113, 1991.

Nathalie PERNELLE, Marie-Christine ROUSSET, Henri SOLDANO et Véronique VENTOS : Zoom : a nested Galois lattices-based system for conceptual clustering. *Journal of Experimental and Theoritical Artificial Intelligence (JETAI)*, 14(2):157–187, septembre 2002.

Cornelius ROSSE et José L.V. MEJINO : A reference ontology for biomedical informatics : the Foundational Model of Anatomy. *Journal of biomedical informatics*, 36(6):478–500, 2003.

Helmut SCHMID : Probabilistic part-of-speech tagging using decision trees. *Dans Proceedings of International Conference on New Methods in Language Processing*, volume 12, pages 44–49. Manchester, UK, 1994.

Patrick Séguéla : *Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, mars 2000.

Patrick Séguéla et Nathalie Aussenac-Gilles : Extraction de relations sémantiques entre termes et enrichissement de modèles conceptuel. *Dans Ingénierie des Connaissances - IC'99 , Palaiseau (F), 14/06/99-18/06/99*, pages 10–20, Paris (F), juin 1999. Ecole Polytechnique.

Nigel Shadbolt, Kieron O'hara et Louise Crow : The experimental evaluation of knowledge acquisition techniques and methods : history, problems and new directions. *International journal of human-computer studies*, 51(4):729–755, 1999.

Edward H. Shortliffe, Randall Davis, Stanton G. Axline, Bruce G. Buchanan, C.Cordell Green et Stanley N. Cohen : Computer-based consultations in clinical therapeutics : explanation and rule acquisition capabilities of the MYCIN system. *Computers and Biomedical Research*, 8(4):303–320, 1975.

Barry Smith et Christopher Welty : FOIS introduction : Ontology : Towards a New Synthesis. *Dans Formal Ontology in Information Systems : Proceedings of the international conference on Formal Ontology in Information Systems- Volume 2001*. Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA, 2001.

John F. Sowa : *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley, 1984. ISBN 0-201-14472-7.

Steffen Staab et Rudi Studer, éditeurs. *Handbook on Ontologies*. Springer Verlag, 2009.

Rudi Studer, V. Richard Benjamins et Dieter Fensel : Knowledge engineering : principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.

Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier et Lotfi Lakhal : Computing iceberg concept lattices with titanic. *Data & Knowledge Engineering*, 42(2):189–222, 2002. ISSN 0169-023X.

Beth M. Sundheim et Nancy A. Chinchor : Survey of the message understanding conferences. *Dans Proceedings of the workshop on Human Language Technology*, pages 56–60. Association for Computational Linguistics, 1993.

Sylvie Szulman, Jean Charlet, Nathalie Aussenac-Gilles, Adeline Nazarenko, Éric Sardet et Valery Teguiak : Dafoe : An ontology building platform - from texts or thesauri. *Dans* Jan L. G. Dietz, éditeur : *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2009)*, pages 372–375. INSTICC Press, 2009.

Hristo TANEV et Bernardo MAGNINI : Weakly supervised approaches for ontology population. *Dans Proceeding of the 2008 conference on Ontology Learning and Population : Bridging the Gap between Text and Knowledge*, pages 129–143, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press. ISBN 978-1-58603-818-2.

Simon TONG et Daphne KOLLER : Support vector machine active learning with applications to text classification. *Dans* Pat LANGLEY, éditeur : *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 999–1006, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

Linda UPCHURCH, Gordon RUGG et Barbara KITCHENHAM : Using card sorts to elicit web page quality attributes. *IEEE Software*, 18(4):84–89, 2001.

Mike USCHOLD et Michael GRUNINGER : Ontologies and semantics for seamless connectivity. *ACM SIGMOD Record*, 33(4):64, 2004.

Gertjan VAN HEIJST, Guss SCHREIBER et Bob J. WIELINGA : Using explicit ontologies in KBS development. *International Journal of Human Computer Studies*, 46:183–292, 1997.

Johanna VÖLKER, Peter HAASE et Pascal HITZLER : Learning expressive ontologies. *Dans Proceeding of the 2008 conference on Ontology Learning and Population : Bridging the Gap between Text and Knowledge*, pages 45–69, 2008.

Yimin WANG, Johanna VÖLKER et Peter HAASE : Towards semi-automatic ontology building supported by large-scale knowledge acquisition. *Dans AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*, volume FS-06-06, pages 70–77, Arlington, VA, USA, octobre 2006. AAAI, AAAI Press.

Serhiy YEVTUSHENKO : System of data analysis "concept explorer". *Dans Proceedings of the 7th national conference on Artificial Intelligence KII-2000, Russia*, pages 127–134, 2000.

Table des matières

Table des figures

Liste des tableaux

Liste des définitions

Résumé

La construction d'ontologies est un processus fastidieux qui nécessite un travail manuel conséquent. Les textes, en tant que sources de connaissances, peuvent optimiser les recours aux experts du domaine. Le passage des textes à l'ontologie requiert un double changement de perspective. Tout d'abord du niveau du discours vers le niveau linguistique (terminologie, hyperonymie, synonymie, etc.), à l'aide d'outils de traitement automatique des langues. La conceptualisation, manuelle, permet ensuite d'entrer dans le monde des modèles. Nous étudions dans cette thèse comment une méthode de regroupement automatique, l'analyse de concepts formels (ACF), peut se combiner aux éléments du niveau linguistique afin de faciliter la tâche de conceptualisation. Nous avons mené des expérimentations sur trois domaines différents, représentés par des corpus de taille comparable. Nous montrons que, dans l'état actuel des connaissances, la construction d'ontologies à partir de textes ne peut s'effectuer de manière totalement automatique. Nous proposons plusieurs paramétrages pour s'affranchir des problèmes inhérents à l'utilisation de l'ACF sur les données textuelles, dans l'optique de fournir à l'utilisateur à la fois des regroupements pertinents et une vue fidèle sur le matériau textuel.

Mots clés : ontologie, analyse de concepts formels, termes, conceptualisation

Abstract

Build an ontology is a tedious task, which still requires a great amount of manual work. Texts, as knowledge sources, can help, but TALN tools stop at linguistic level. Manual conceptualization fill the gap between a linguistic model and a conceptual model. In this thesis we study how a symbolic clustering method, Formal Concept Analysis, can be combined with a linguistic model to help the knowledge engineer. We have experimented on three different domains represented by same-sized corpora. We show that ontology learning from texts cannot be fully automatized. We propose solutions that combine FCA and terminological analysis, to let the computer suggests useful clusters and faithful representation of texts.

Keywords : ontology, formal concept analysis, terms, conceptualization

DISCIPLINE : INFORMATIQUE

Laboratoire d'Informatique de Paris-Nord — UMR CNRS 7030
99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France