Annotation Semantique de Documents Semi-Structurés pour la recherche d'information

Mouhamadou Thiam

Laboratoire de Recherche en Informatique — U.M.R CNRS 8623 — Université de Paris-Sud

Annotation Sémantique de Documents Semi-structurés pour la Recherche d'Information

THÈSE EN CO-TUTELLE

présentée et soutenue publiquement le 09 Décembre 2010

pour l'obtention du

Doctorat des Universités de Paris-Sud et Gaston Berger

Spécialité : informatique

par

Mouhamadou THIAM

Composition du jury

Rapporteurs :	Nathalie Aussenac-Gilles	Directrice de Recherche – IRIT – Toulouse
	Fabien Gandon	Chercheur Senior – Edelweiss Research team INRIA – Sophia Antipolis
Examinateurs :	Cheikh Talibouya Diop	Maître de conférences HDR – LANI – UGB
	Amel Bouzeghoub	Professeur – TELECOM & Management – Sudparis
Directeurs de thèse :	Moussa LO	Maître de conférences HDR – LANI – UGB
	Chantal Reynaud	Professeur – LRI – Université Paris-Sud
Co-directrices :	Nacéra Bennacer	Maître de conférences – E3S– SUPELEC
	Nathalie Pernelle	Maître de conférences – LRI – Univ. Paris-Sud

Université Paris-Sud, 91405 Orsay Cedex, France

Mis en page avec la classe thloria.

Je dédie cette thèse
à ma mère Bigué Kébé,
à mon père El Hadji Yague,
à mes nièces Fatou Kiné *et* BB Khady,
et à mon futur petit bout-de-bois-de-Dieu...

Remerciements

J'aimerais remercier toutes les personnes qui m'ont assisté durant mes travaux de recherche et d'écriture de cette thèse :

Amel Bouzeghoub pour l'intérêt qu'elle a porté à cette thèse en acceptant de présider le jury de ma soutenance.

Nathalie Aussenac-Gilles et *Fabien Gandon* qui me font l'honneur d'être rapporteurs de cette thèse, mais également pour leur amabilité.

Nacéra Bennacer et *Nathalie Pernelle*, mes co-directrices de thèse, pour leurs idées et critiques constructives, leurs remarques toujours pertinentes, leur relecture rigoureuse de ce mémoire et nos discussions animées autour de ce travail thèse.

Moussa Lo et *Chantal Reynaud*, mes directeurs de thèse, pour leur disponibilité, leurs critiques et leurs conseils toujours opportuns.

Cheikh Talibouya Diop pour l'intérêt qu'il a porté à cette thèse en acceptant de faire partie de mon jury.

Toute l'équipe *IASI*, qui m'a accueilli, me procurant ainsi un cadre de travail et de réflexion propice à la réalisation de cette thèse, grâce notamment aux divers projets et conférences auxquels j'ai pu participer activement.

Tous les membres du département informatique de *Supelec*, et notamment *Yolaine Bourda*, *Evelyne Faivre*, tous les doctorants, pour les nombreuses heures passées ensemble, pour leur aide précieuse et pour leur bonne humeur revigorante.

Ma famille, et plus particulièrement mes parents, qui ont su m'encourager jour après jour et qui ont toujours cru en moi. Sans leur soutien constant, tant affectif que matériel, je n'aurai jamais pu accomplir mes études et envisager cette thèse.

Mon oncle *Mouhamed Kébé* pour son soutien matériel, pour tous les livres qu'il a achetés depuis mes premiers pas à l'école.

L'ensemble de mes amis, plus particulièrement *Momath Kébé*, pour ces derniers longs mois d'absence durant ces années de thèse.

Les familles de *Abou* et *Thioub* de Kaolack, spécialement à *Yaye Diale Touré*, pour leur accueil et assistance, mais aussi la famille *Diassé* de Ndar, *Madjiguène* et ses enfants.

Et enfin, mon chouchou, ma femme, mon épaule, mon cerveau aussi parfois, *Fatimata*, pour son amour, sa tendresse, sa présence et sa grande patience surtout pour ses premières *Eid* passées sans moi, sans qui je ne serai peut-être pas arrivé au bout de cette aventure.

Table des matières

Introduction générale

1 Contexte

Le web constitue une masse d'information gigantesque. Le nombre de ressources accessibles sur le web croît de façon exponentielle et le nombre d'usagers double chaque année. Selon Tim Berners-Lee, le web sémantique est un ensemble de méthodes et de technologies permettant à des agents logiciels de manipuler et de raisonner sur le contenu des ressources du Web. Cependant, le web actuel est encore loin de cet idéal.

Cette vision du Web futur dépend de la construction, de la prolifération des ontologies et de l'utilisation de métadonnées pour décrire et représenter les ressources Web. Une ontologie est définie comme *une spécification explicite d'une conceptualisation* [Gru93]. Une métadonnée est définie comme une donnée qui définit d'autres données notamment leur sémantique.

L'annotation sémantique des ressources Web à l'aide de métadonnées permet une meilleure interprétation de leur contenu. Quand ces annotations sont disponibles, il est possible de formuler des requêtes basées sur le vocabulaire des ontologies, de raisonner sur ces annotations, et de recueillir des réponses combinant des données provenant de différents documents. Les moteurs de recherche sont alors capables d'aller au-delà d'une recherche mots-clés et de répondre précisément à des requêtes complexes.

L'annotation sémantique de ressources Web est un problème qui intéresse différentes communautés. De nombreux projets existent, certains visent à construire des bases d'annotations comme DBpedia [ABK+07] qui annote sémantiquement des pages wikipédia dans le langage RDF [W3Cb]. Le projet Linked-Data est un projet plus ambitieux qui vise à décrire, à partager et à connecter différentes ressources provenant de différentes sources telles que DBpedia, DBLP et Geo-names. D'autres projets ont pour objectif de faciliter l'annotation manuelle de ressources Web comme Annotea [KK01] un standard du W3C qui vise à améliorer la collaboration sur le web en annotant les documents dans le langage RDF.

Compte tenu du nombre des ressources disponibles, l'annotation manuelle est une tâche longue et fastidieuse qui ne sied pas à l'échelle du Web. L'automatisation des techniques d'annotation est un facteur clé pour le Web du futur et son passage à l'échelle. L'une des principales difficultés rencontrées réside dans le fait que les ressources Web sont très hétérogènes aussi bien du point de vue de leur format (HTML, pdf, gif, mpeg, avi,...) et de leur structuration que du point de vue du vocabulaire utilisé. Les approches proposées dans la littérature concernent des domaines complémentaires tels que le datamining, l'intelligence artificielle, l'ingénierie des connaissances et le traitement du langage naturel.

Pour limiter les problèmes liés à l'annotation, les méthodes développées se focalisent souvent sur un domaine d'application, sur une classe de sites Web ou encore sur des entités précises comme l'annotation des noms de personnes, des adresses e-mail, ou des pays.

2 Problématique

Techniquement, une annotation sémantique consiste à assigner à une entité (une chaîne de caractères, une phrase, un paragraphe, une partie de document ou un document) une métadonnée dont la sémantique est définie dans une ontologie. Elle a comme objectif d'annoter des concepts, des instances de concepts ou leurs relations. Cette annotation peut être stockée dans le document lui-même, ou dans un autre document référençant l'entité annotée par son URI (Universal Resource Identifier). L'annotation de documents est une problématique trés liée à l'extraction d'information. La différence est que l'extraction ne conserve pas le contexte d'apparition (le document) de l'entité extraite.

Les instances de concepts de l'ontologie recherchées sont décrites de différentes façons dans les textes de documents, par un paragraphe, par une phrase, un segment de phrase, par un terme ou une entité nommée. Souvent, ces termes et ces entités nommées représentent une part importante de la sémantique du document.

Une méthode d'annotation est caractérisée par le niveau de structuration des documents qu'elle traite. Certaines approches sont spécialisées dans de l'annotation du texte libre et d'autres dans l'annotation d'éléments de structure spécifiques comme les tableaux et les listes. Cependant, les documents existants sont souvent structurellement hétérogènes et comportent des parties bien structurées et des parties textuelles.

Si les documents sont conçus par des auteurs différents et pour des besoins différents, ces termes et ces entités nommées ne sont pas décrits dans des expressions linguistiques qui se répètent. L'utilisation de méthodes basées sur des patterns ou des règles spécifiques au domaine ou au corpus traité ne permettrait pas de couvrir correctement les termes et les entités nommées de concepts ou leurs relations. Compte tenu du nombre et de l'hétérogénéité des documents, l'utilisation de méthodes exploitant une base d'exemples pré-annotés (labellisés) permettant de superviser l'apprentissage de règles d'extraction et d'annotation aura également un taux de couverture limité.

Dans ce contexte, une méthode d'annotation qui se veut automatique est confrontée au fait que les termes et les entités nommées sont difficiles à repérer et à délimiter précisément dans les parties textuelles des documents traités. L'une des solutions est alors de s'appuyer sur des bases lexicales, des bases de connaissances (ontologie peuplée d'instances) existantes ou le Web.

3 Contributions

Notre travail de thèse se situe dans le cadre du projet *SHIRI* (Système Hybride d'Intégration pour la Recherche d'Information dans les ressources hétérogènes) [1].

1. SHIRI : Projet Digiteo labs (LRI, Supélec). Lot 1 : Annotation sémantique de documents semi-structurés (*SHIRI-Extract* et *SHIRI-Annot*). Lot 2 : Interrogation des annotations (*SHIRI-Quering*).

Notre objectif est d'annoter sémantiquement une collection de documents relatifs au même domaine et délimités par des balises (XHTML, XML). Ces documents sont structurellement hétérogènes et comportent des parties bien structurées et d'autres textuelles. Notre approche se veut automatique et non supervisée.

Nous nous plaçons dans le contexte où nous disposons d'une ontologie de domaine définie par un ensemble de concepts, un ensemble de relations entre ces concepts et leurs propriétés. Cette ontologie comporte une composante lexicale où chaque concept est accompagné de plusieurs labels, d'un premier ensemble d'entités nommées et de termes du domaine. L'ensemble des labels d'un concept est un ensemble de termes considérés comme équivalents (synonymie dans le domaine) pour représenter un concept. L'ensemble des termes d'un concept est un ensemble de termes qui décrivent les instances des concepts du domaine. Les termes d'un concept sont plus spécifiques que ses labels. Ces ensembles sont validés par l'expert du domaine.

L'idée maîtresse, dans notre travail, est de repérer des termes ou entités nommées de concepts dans les nœuds de l'arbre DOM (Document Object Model) représentant la structure d'un document (HTML ou XML). Pour avoir une bonne couverture, nous extrayons les termes ou les entités nommées candidats de manière indépendante du domaine et des documents. Ces termes ou entités nommées candidats sont rapprochés des termes ou des labels de l'ontologie lexicale afin de déterminer les concepts appropriés. Vu que ces termes ou ces entités nommées ne sont pas délimités parfaitement dans le texte, nous avons choisi d'annoter les nœuds dans lesquels ils sont repérés. Ce qui permet de prendre en compte cette délimitation imprécise et de conserver les termes et les entités nommées repérés dans leur contexte structurel minimal : le nœud qui les contient. De même, nous exploitons la proximité structurelle des nœuds instances pour déduire l'existence possible d'une relation sémantique de domaine et l'annoter par une relation de voisinage, sans chercher à l'identifier précisément.

Dans ce contexte, nous avons plusieurs problèmes à résoudre liés à : (i) l'extraction des termes et entités nommées candidats, (ii) au rapprochement avec l'ontologie, (iii) au fait que la composante lexicale ne soit pas suffisamment riche et (iv) à l'annotation qui soit adaptée au contenu du nœud.

Notre approche est composée de trois modules :

- *SHIRI-Extract* : est le module qui extrait des termes et des entités nommées (EN) dans les documents, qui les associe aux concepts de l'ontologie et qui enrichit la composante lexicale de l'ontologie.

- *SHIRI-Annot* : est le module qui annote les nœuds de documents conformément au modèle d'annotation que nous avons défini.

- *SHIRI-Querying* : est le module qui permet de poser des requêtes sur la base des annotations, ces requêtes sont formulées à l'aide des métadonnées définies dans le modèle d'annotation.

L'ontologie ainsi que les annotations sont représentées avec les langages RDF(S).

Plus précisément, nos contributions dans ce travail de thèse sont les suivantes :

Extraction des termes et entités nommées candidats et alignement avec l'ontologie.

Nous utilisons des patterns syntaxiques représentant des syntagmes nominaux utilisés couramment pour l'extraction des termes. De la même façon, les entités nommées sont extraites grâce à un ensemble de patterns lexico-syntaxiques. Ces patterns sont indépendants du domaine et du corpus.

Les termes et les entités nommées candidats sont associés aux concepts de l'ontologie avec des mesures de similarité différentes parce que ces entités ne subissent pas les mêmes types de variations syntaxiques. Ces mesures exploitent la catégorie grammaticale des mots, la similarité des chaînes de caractères ainsi que la taxonomie dans laquelle ils apparaissent. Les mesures que nous utilisons pour les termes sont celles utilisées pour l'alignement de deux ontologies dans Taxomap [HZSR08]. Dans notre approche nous appelons alignement le processus qui permet d'associer un terme ou une entité nommée candidat avec les concepts de notre ontologie.

Même si la composante lexicale de l'ontologie est riche, ces mesures ne permettront pas de trouver tous les alignements. Notre approche exploite le Web pour découvrir des termes en relation d'hyponymie avec les termes candidats ou des termes décrivant les concepts auxquels les entités nommées appartiennent. Ces termes découverts permettent ainsi d'aligner d'autres termes ou entités nommées candidats.

Enrichissement de l'ontologie.

L'ontologie est enrichie lexicalement par les termes ou EN découverts dans les documents ou sur le Web. Ainsi le besoin d'accès au Web, qui peut être coûteux, est réduit au fur et à mesure que les documents, appartenant au même domaine, sont traités. Cet enrichissement peut être validé par l'expert pour s'assurer de sa qualité.

Annotation des nœuds de documents

Nous avons défini un modèle d'annotation pour représenter les résultats d'extraction et d'annotation. L'annotation d'un nœud est définie en fonction de la présence d'un ou plusieurs termes ou ENs dans ce nœud et des concepts avec lesquels ils ont été alignés. Les métadonnées du modèle d'annotation permettent de distinguer les nœuds selon que les termes ou les entités nommées agrégés dans un même nœud soient alignés avec un ou plusieurs concepts différents ou similaires sémantiquement.

Ces métadonnées permettent également d'annoter la relation de voisinage entre les nœuds instances en exploitant leur sémantique et leur proximité structurelle en termes de longueur du chemin les reliant dans l'arbre structurel représentant les documents.

Pour annoter les nœuds instances et leur relation, nous avons défini un ensemble de règles déclaratives qui permettent de transformer les triplets résultant de l'extraction en triplets d'annotations conformes au modèle d'annotation que nous avons défini.

La base d'annotations RDF ainsi construite peut être interrogée en utilisant des requêtes SPARQL.

Notre approche est implémentée, expérimentée et évaluée sur une collection de documents portant sur les appels à participation à des conférences du domaine informatique et les résultats obtenus sont très encourageants.

Les résultats obtenus en utilisant le module *SHIRI-Querying* ont bien montré l'intérêt de notre modèle d'annotation pour trier les réponses selon le niveau d'agrégation des termes et entités nommées repérés dans les nœuds. De plus, la granularité de l'annotation permet de présenter à l'utilisateur des nœuds et non pas des documents en réponses.

4 Plan du manuscrit

Ce manuscrit est organisé en cinq chapitres.

Chapitre I : Etat de l'art.

Dans ce chapitre, nous présentons les problèmes liés à l'annotation dans les documents semi-structurés et les principales orientations et techniques utilisées pour traiter ces problèmes. Ensuite, nous présentons brièvement les langages du W3C pour le Web sémantique. Nous concluons en positionnant notre approche par rapport aux approches présentées.

Chapitre II : Présentation de l'architecture de notre système.

Dans ce chapitre, nous présentons formellement l'ontologie à composante lexicale et le modèle d'annotation. Ensuite, nous présentons brièvement l'architecture de notre système, et ses différents modules. Enfin, nous illustrons à travers un scénario d'usage l'utilisation successive des modules *SHIRI-Extract*, *SHIRI-Annot* et *SHIRI-Querying*.

Chapitre III : Extraction, alignement des entités nommées et des termes.

Ce chapitre présente notre approche d'extraction et d'alignement sémantique d'entités nommées et de termes. Tout d'abord, nous présentons l'extraction des entités nommées et des termes candidats, ainsi que l'alignement local et l'alignement utilisant le Web. Nous détaillons l'algorithme *Extract-Align*, l'enrichissement lexicale de l'ontologie et les pré-annotations qui en résultent. Le déroulement de *Extract-Align* est ensuite détaillé sur un document de la collection traitée.

Chapitre IV : Annotation des documents et interrogation.

L'objectif de ce chapitre est de présenter *SHIRI-Annot* qui permet de générer les annotations des nœuds de documents conformes au modèle d'annotation à partir des pré-annotations en utilisant un ensemble de règles déclaratives que nous avons définies. Nous présentons un ensemble de fonctions de transformations du module *SHIRI-Querying*. Ces

fonctions sont définies pour construire les reformulations permettant d'atteindre tous les nœuds réponses triés selon le niveau d'agrégation des termes ou entités nommées repérés dans ces nœuds.

Chapitre V : Expérimention et évaluation des résultats.

Ce chapitre présente notre système d'extraction, d'alignement et d'annotation qui implémente l'algorithme *Extract-Align* et intègre d'autres outils existants. Nous présentons l'évaluation des résultats obtenus des expérimentations menées sur une collection de documents portant sur les appels à participation à des conférences du domaine informatique.

Chapitre 1

ÉTAT DE L'ART

Sommaire

Introduction

L'un des objectifs du web sémantique est de construire et d'utiliser des métadonnées permettant d'annoter sémantiquement les ressources afin d'améliorer la recherche d'information. Atteindre cet objectif dépend de la prolifération des ontologies, des possibilités d'automatiser le processus d'annotation et du passage à l'échelle. Par ailleurs, les ressources Web sont très hétérogènes aussi bien du point de vue de leur structuration que du point de vue du vocabulaire utilisé.

Le problème d'annotation sémantique de ressources Web intéresse des chercheurs de différentes communautés. Les approches proposées concernent des domaines complémentaires tels que le datamining, l'intelligence artificielle, l'ingénierie des connaissances et le traitement de la langue naturelle. La plupart des approches d'annotation, qui existent dans la littérature, se concentrent sur un domaine d'application, sur une classe de sites

Web ou encore sur des entités précises comme l'annotation des noms de personnes, des adresses e-mail, des pays, etc.

L'objectif de ce chapitre est de présenter les problèmes liés à l'annotation dans les documents semi-structurés et de survoler les principales orientations et techniques utilisées pour traiter ces problèmes. Dans la section 1, nous présentons un ensemble d'approches que nous avons choisies. Dans la section 2, nous présentons brièvement les langages du W3C pour le Web sémantique. Dans la section 3, nous concluons et nous positionnons notre approche par rapport aux différentes orientations.

1.1 Extraction et annotation sémantique dans les documents semi-structurés

L'annotation sémantique consiste à assigner à une entité (une chaîne de caractères, une phrase, un paragraphe, une partie de document ou un document) une métadonnée dont la sémantique est souvent définie dans un modèle. Cette métadonnée peut être stockée dans le document lui-même, ou dans un autre document référençant l'entité annotée par son URI [2]. L'annotation externe est plus intéressante car elle permet de référencer une seule fois une métadonnée ayant des entités présentes dans plusieurs documents. Dans ce cas, une annotation est un ensemble d'instances (de métadonnées) rattachées à des ressources Web. L'annotation de documents est un processus similaire au processus d'extraction d'information. La différence est que l'extraction ne conserve pas le contexte d'apparition (le document) de l'entité.

De nombreuses approches d'extraction et d'annotation de documents semi-structurés existent dans la littérature. Certaines ont pour objectif d'annoter des documents, d'autres de construire une base de connaissances (une ontologie peuplée d'instances) ou les deux à la fois. Cette extraction peut concerner les concepts, les relations ou encore les instances de concepts. Ces approches se distinguent non seulement par leurs objectifs mais aussi par les hypothèses qu'elles exploitent, et qui souvent ne sont adaptées qu'à des corpus spécifiques. Ces hypothèses peuvent concerner l'utilisation de :

- une ontologie plus ou moins peuplée (une base de connaissances) ;
- structures régulières de documents, ou des structures spécifiques comme les tableaux et les listes ;
- patterns lexico-syntaxiques dans le texte permettant de délimiter et d'extraire les entités d'information ;
- ressources externes lexicales ou le Web ;
- une base d'exemples pré-annotés (étiquetés ou labellisés) permettant de superviser l'apprentissage de règles d'extraction et d'annotation ;

2. URI : Uniform Resource Identifier

- l'intervention de l'utilisateur pour le choix des données en entrée et/ou pour valider des résultats.

Extraction de termes

Les extracteurs de termes sont des outils qui dépouillent automatiquement un corpus de textes pour extraire des termes. Un terme peut être défini comme *la réalisation linguistique d'un concept*. Ces extracteurs sont généralement utilisés pour construire une terminologie permettant d'indexer ou d'annoter des documents. Les approches d'extraction de termes appartiennent à trois catégories : les approches linguistiques, les approches statistiques et les approches mixtes. Les approches statistiques sont basées sur la redondance des mots et sur leurs probabilités de co-occurences dans les documents. Elles utilisent des mesures comme le cosinus ou encore l'information mutuelle [DES05]. Ces méthodes ont une bonne couverture et sont indépendantes des langues mais nécessitent un corpus d'une taille minimale et une bonne distribution des termes.

Les approches linguistiques se basent sur la catégorie morpho-syntaxique des mots d'un terme. Un terme doit correspondre à une séquence syntaxiquement valide [Bou]. Elles exploitent des patterns syntaxiques de formation de termes de type syntagmes nominaux ou verbaux [Bou92, AH06, Jac94]. Elles sont donc plus précises mais dépendantes de la langue.

Ces patterns peuvent être lexico-syntaxiques, dépendant ou indépendants du domaine. Les patterns de Hearst [Hea92] sont un exemple de patterns lexico-syntaxiques qui sont indépendants du domaine qui permettent de délimiter dans le texte des termes (syntagmes nominaux) et leur relation d'hyponymie. Par exemple, *works by* ***such*** *authors* ***as*** *Herrick, Goldsmith, and Shakespeare*, *... temples, treasuries,* ***and other*** *important civic buildings*, *All common-law countries,* ***including*** *Canada and England ...* sont des portions de textes répondant aux patterns de Hearst présentés dans la section 3.1.4.

Extraction d'entités nommées

L'expression *entité nommée* (EN) est apparue lors de la conférence MUC−6 (Message Understanding Conférence) [GS96]. Les catégories d'EN définies sont les personnes, les lieux et les organisations. Cette définition a été élargie aux expressions telles que les noms d'espèces (ex : *cèdre du Liban*), de maladies, ou de substances chimiques, aux expressions collectives (ex : *les Parisiens*, *les Néendertaliens*), aux expressions temporelles telles que les dates et les heures, ou aux numériques (ex : *2000 euros*).

Des travaux tels que [SC04] ont défini une hiérarchie d'entités nommées qui comporte 150 catégories telles que les aéroports, les musées ou les religions. Ainsi, certaines approches exploitent des taxonomies issues d'encyclopédies telles que Wikipédia [KT07]. D'autres travaux, dits de Reconnaissance d'Entité Nommée (REN), définissent des techniques qui combinent les ressources lexicales et l'analyse d'expressions régulières pour détecter et délimiter les EN dans le texte et leur associer une catégorie [Eva03]. Ces expressions régulières sont des patterns lexico-syntaxiques utilisant souvent la casse des

caractères. Comme pour les termes, et en particulier pour les EN, la difficulté réside dans :

- la localisation des limites des expressions désignant l'entité nommée (exemple : l'EN *le pic du Canigou* ou *le Canigou*) ;
- la surcomposition dans lesquels une EN peut en contenir une autre (exemple : L'EN *Université Gaston Berger* contient l'EN *Gaston Berger*) ;
- l'ambiguité (exemple : l'EN *Roissy* qui peut désigner l'aéroport ou la ville de Roissy) ou la coréférence (exemple : l'EN *Ford Compagny* et l'EN *Ford*).

Pour faire la jonction entre les termes, les entités nommées et les ontologies, il existe des approches qui préconisent de les représenter en utilisant une composante lexicale intégrée dans l'ontologie ([BAGC04, RTAG07], Terminae [BS99]). Ces termes et entités nommées sont alors associés aux concepts et aux relations de cet ontologie. Dans OntoPop [Ama07], l'utilisation de règles d'acquisition de connaissances permettent de passer des résultats d'un extracteur tel que IDE (Insight Discoverer Extractor) de Temis à l'ontologie.

De nombreux systèmes ont été créés pour faciliter l'extraction et l'annotation semi-automatique qui impliquent l'utilisateur et qui l'assistent grâce à l'intégration de différentes techniques d'extraction. Nous citons, à titre d'exemple Annotea [KK01], S-CREAM (Semi-automatic CREAtion of Metadata) [HSC02] et Terminae [BS99].

Dans ce qui suit, nous présentons un ensemble d'approches et de systèmes d'extraction et d'annotation de documents semi-structurés classés en fonction des éléments qu'ils exploitent dans le tableau 1.1.

1.1.1 Approches exploitant la structure des documents

L'organisation structurelle des éléments dans les documents semi-structurés peut être exploitée dans un processus d'extraction d'information. De nombreuses approches comme [VGP01], [DVN05], [BFG01b], [AKM$^+$03],[AGM03] et [HBDBH07] exploitent les éléments de structure des documents tels que les tableaux, les listes, les imbrications de ces éléments pour générer automatiquement ou semi-automatiquement des modèles ou des patterns basés sur la structure. Ces patterns permettent d'identifier des concepts, des instances de concepts et/ou des relations entre ces instances.

Lixto

Lixto [BFG01b] est un système qui permet d'encoder un ensemble de pages HTML en XML de manière semi-automatique et supervisée. Tout d'abord, l'utilisateur sélectionne un ensemble d'élements dans une page, en précisant leur ordre, le système génère un ensemble hiérarchique de patterns en analysant le contexte structurel des éléments et sa similarité avec d'autres éléments. Par exemple, un pattern $< item >$ et un sous-pattern $< price >$ expriment qu'une instance $< price >$ apparaît dans une instance de type

$< item >$. Les noms des patterns représentent des balises XML. L'ensemble des instances d'un pattern ont des niveaux de structuration différents, elles peuvent correspondre à une chaîne de caractères ou un paragraphe ou à une liste, tout dépend de la structure du document en question. Le système encode dans un langage appelé *Elog* [BFG01a] l'ensemble des patterns sous forme de règles déclaratives qu'il applique à d'autres documents HTML pour générer des annotations XML .

Appliqué à des pages de e-commerce, Lixto donne de bons résultats. Par ailleurs, Lixto suppose que l'utilisateur est un expert dans le domaine, et malgré le fait que les documents soient de structures hétérogènes, une règle n'a de couverture minimale que si une certaine régularité structurelle existe.

Ontominer

Ontominer [DVN05] est un système qui permet de construire automatiquement une taxonomie de concepts à partir d'un ensemble de pages HTML et d'annoter ces documents en XML. Il détecte et exploite la régularité structurelle des documents pour extraire les concepts, leurs relations de taxonomie et les instances. Une page est représentée par son arbre DOM (Document Object Model du W3C). L'algorithme d'extraction repose sur trois principales hypothèses :

- l'existence d'une corrélation entre la représentation structurelle et arborescente d'un document et les relations taxonomiques entre concepts définies dans l'ontologie ;
- l'ensemble des documents présente une structuration régulière ;
- l'ensemble des pages Web partage une taxonomie.

Pour identifier les concepts et leurs relations taxonomiques, Ontominer identifie les chemins des arbres DOM qui sont similaires et exploite la répétition des labels ou des labels similaires en utilisant la mesure de Jaccard [TSK05] (après lemmatisation). Un algorithme spécifique appelé *Data Table Detector* est défini pour les structures tabulaires. Ontominer procède ensuite à l'identification des instances de concepts en utilisant les chemins depuis un nœud vers une feuille. Il détermine si les segments de nœuds identifiés sont similaires ou pas en utilisant les mesures de Levenshtein [LSF07] et Jaccard. Ces segments considérés comme des instances différentes correspondent soit à des valeurs soit à des nœuds semi-structurés. Les collections traitées sur la description des hôtels et les articles de journaux répondent bien à ces hypothèses.

Dans notre approche, nous n'utilisons ni la régularité structurelle ni la nature des éléments de structure mais nous exploitons la proximité et l'imbrication des nœuds dans lesquels des instances sont repérées pour générer une relation de voisinage entre nœuds.

1.1.2 Approches exploitant des patterns syntaxiques ou lexico-syntaxique dans le texte

L'exploitation des patterns lexico-syntaxiques permet de segmenter un texte en entités et délimiter les différentes entités. Le lexique utilisé dans le pattern peut provenir d'un dictionnaire.

LP^2

LP^2 (Learning pattern) [Cir01] est un algorithme supervisé d'extraction d'instances de concepts et d'annotation en XML de pages web faiblement structurées. L'entrée de l'algorithme est un ensemble de pages annotées manuellement, la sortie est un ensemble de règles d'annotation. L'algorithme génère des règles adaptées aux textes traités et destinées à la détection d'instances de concepts. Il utilise des informations morpho-syntaxiques des mots (nom, verbe, etc., singulier ou pluriel), le typage de certains mots reconnus dans un dictionnaire (noms de villes, de pays...) et la reconnaissance des entités nommées (noms de personnes, d'organisations, dates...).

- LP^2 effectue sa phase d'apprentissage inductif (induite à partir d'exemples) des règles d'extraction de connaissances à l'aide d'un ensemble d'apprentissage décomposé en deux sous-ensembles : le premier est utilisé pour l'induction des règles qui permettent l'insertion de balises XML (certaines règles permettent l'insertion d'une balise ouvrante, d'autres d'une balise fermante), le deuxième est utilisé pour la sélection des règles les plus efficaces parmi les règles générées : l'induction de règles de correction permettant de déplacer les balises afin de corriger les erreurs et imprécisions des règles d'annotation précédentes.

- Pour chaque balise ouvrante ou fermante de l'annotation manuelle (appelée exemple), un pattern comprenant une fenêtre de n mots avant et après la balise est généré. Ce pattern est ensuite généralisé : chacun des mots du pattern peut être généralisé en le représentant par son lemme, sa classe syntaxique (nom, verbe, chiffre...), sa typographie (majuscule, minuscule...) ou sa sémantique (reconnu dans le dictionnaire ou entité nommée), ou encore par un joker autorisant n'importe quel mot. Toute combinaison de généralisation (ou mot conservé à l'identique) du pattern initial devient un pattern candidat pour devenir une règle d'annotation.

- Les patterns conservés sont uniquement ceux qui, appliqués à l'ensemble des documents annotés manuellement, permettent de couvrir un nombre minimal d'exemples, et dont le taux d'erreur (i.e. le nombre de balises insérées au mauvais endroit rapporté au nombre total de balises insérées) est inférieur à un certain seuil. Le nombre minimal d'exemples à couvrir et le seuil sont fixés par l'utilisateur.

- Ces patterns sont enregistrés dans la liste des règles d'annotation et sont appliqués à l'ensemble du corpus. Tous les exemples couverts par ces patterns sont retirés de la liste des exemples à traiter, puis le processus est répété avec l'exemple suivant,

jusqu'à ce qu'il n'y ait plus d'exemple à traiter. Ceci permet d'obtenir une liste de meilleures règles, les règles d'annotation ayant une forte précision, mais une couverture assez faible.

Pour augmenter la couverture, des règles contextuelles sont ajoutées. Elles correspondent aux patterns n'ayant pas été retenus comme meilleures règles à cause d'un taux d'erreur trop important, mais que l'on peut rendre plus précis en contraignant leur application dans une certaine fenêtre autour d'une balise insérée dans l'étape précédente par les meilleures règles.

TextRunner

TextRunner [BCS+07] est un système plus récent qui extrait des triplets de relations à partir du texte pour construire une base de connaissances. Cette extraction est automatique et indépendante du domaine, supervisée et la base d'exemples d'apprentissage est construite automatiquement.

- En utilisant un petit ensemble de documents, ce module extrait un ensemble de triplets (e_i, r_{ij}, e_j) à partir de chaque phrase où e_i et e_j sont des phrases nominales et r_{ij} est le texte entre les deux. La relation est labellisée comme positive ou négative selon que certaines contraintes comme la longueur de la chaîne, l'existence de clauses relatives, le fait que e_i ou e_j soit un pronom. Pour chaque triplet sont conservées les caractéristiques telles que le nombre de mots d'arrêt, le nombre de mots, si e_i ou e_j est un pronom, les étiquettes morpho-syntaxiques des mots. L'ensemble des exemples labellisés constitue une base pour le classifieur naïf de Bayes.

- L'extraction génère des triplets candidats à partir de chaque phrase. Chaque mot du triplet est étiqueté mopho-syntaxiquement, les relations sont détectées en examinant le texte entre les phrases nominales et normalisées en éliminant par exemple les phrases prépositionnelles qui sur-spécifient une entité ou les adverbes. Par exemple la phrase *"scientists from many universities are studying..."* est réduite à *"scientists are studying..."* et la phrase *"...definitely developped..."* est réduite à *"...developped..."*. Pour chaque phrase nominale, à chaque mot est affectée la probabilité que le mot fasse partie de l'entité, ce qui permet de sélectionner les triplets ayant une confiance suffisante. Le triplet $t(e_i, r_{ij}, e_j)$ est ensuite traité par le classifieur qui labellise t comme correct ou non. Le système utilise enfin le modèle de redondance basé sur l'information mutuelle pour valider une relation entre deux entités.

- Textrunner permet d'interroger plusieurs millions de triplets. Le triplet est ce qui va permettre de soumettre des requêtes *"intelligentes"* en comparaison à celles permises par les moteurs classiques. Par exemple, la requête constituée du triplet *"Who killed Kennedy ?"* peut être posée.

Les résultats des expériences présentés dans [BCS+07] montrent que la couverture et la qualité des relations entre instances trouvées par TextRunner sont meilleures que celles

utilisant des patterns lexico-syntaxiques présentées dans la section suivante 1.1.3.

Dans [SM07], les triplets résultant de Textrunner sont enrichis conceptuellement comme suit :

- les arguments sont alignés en utilisant les synonymes de Wordnet ;
- les relations sont alignées avec des (meta) relations prédéfinies et qui sont indépendantes du domaine comme les relations *partOf* et de causalité.

Par exemple, le triplet (*orange*, *is rich in*, *vitamin C*) est aligné à ($Fruit$, *contain*, $Vitamin$).

Textrunner et LP^2 sont toutes les deux des approches supervisées et génèrent des règles contextuelles qui sont spécifiques aux documents. Ces approches supposent donc que les documents présentent des formes grammaticales et lexicales qui se répètent. Notre approche ne génère pas des règles d'extraction spécifiques aux documents car les instances candidates sont extraites en utilisant des patterns syntaxiques indépendants du domaine.

1.1.3 Approches exploitant des ressources lexicales et/ou le web

KIM

Le système KIM [PKK$^+$04] a pour objectif d'annoter automatiquement les entités nommées dans les documents. Il se base sur les ressources suivantes :

- une ontologie généraliste appelée PROTON, qui définit des concepts, des relations, des attributs et des axiomes ;
- une base de connaissances décrivant des instances dans le vocabulaire de l'ontologie PROTON. Chaque entité y est éventuellement décrite par un ensemble de noms alternatifs appelés alias (exemple : *New York*, $N.Y$)
- des ressources lexicales permettant de reconnaître une nouvelle EN, telles que des listes de noms et de prénoms de personnes, des préfixes ou suffixes (exemple : *company, SA*), ou des devises.

L'objectif de l'approche est d'annoter sémantiquement des entités nommées trouvées dans un document qui existent déjà dans la base de connaissances ou sont nouvelles. KIM extrait les EN en utilisant les outils de la plate-forme GATE [CGW95] et annote chacune des EN extraite du texte par le concept le plus spécifique de l'ontologie auquel l'instance décrite appartient (exemple : le concept *Sea* pour l'EN *Arabian Sea*) ainsi que par un lien vers l'instance de la base de connaissances, si cette instance existe déjà. Les annotations résultats sont représentées en RDF. La base de connaissances contient environ 200 000 instances et comprend des lieux tels que les continents, les pays, les villes de plus de 100 000 habitants, ou les rivières ainsi que leurs coordonnées géographiques, les entreprises et organisations les plus connues (exemples : Apple, NATO, OPEC). Les auteurs indiquent

que ce type de base de connaissances peut toujours être peuplée avec des connaissances provenant de sources de références et que l'on ne peut extraire d'EN sur la base de l'ontologie seule.

Si l'EN extraite est nouvelle, celle-ci est conservée dans la base de connaissances et peut être utilisée pour de futures annotations sémantiques.

Armadillo

Armadillo [CCDW04] est un système d'annotation non supervisé qui peut aussi bien annoter des documents en utilisant une ontologie de domaine que créer une base de connaissances à partir de différentes entrepôts d'informations. Armadillo utilise des concepts *"ancres"*, qui servent d'accroche pour trouver les instances du concept recherché, et des concepts *"annexes"*, qui permettent de valider la classification des instances repérées.

Prenons l'exemple où des instances du concept *Professor* sont recherchées. Le concept *Person*, un super-concept de *Professor*, peut être un concept *"ancre"* qui possède des instances qui sont facilement repérables dans le texte. En revanche, le concept *Person* est ambigu. Il peut être désambiguïsé en utilisant les relations de ses instances avec des instances de concepts *"annexes"*, comme *University* ou *Article*, sur lesquelles il y a peu d'ambiguïté.

L'idée maîtresse dans Armadillo est d'utiliser tout d'abord des informations certaines obtenues sur des instances, par exemple ici, des listes d'universités. Ces instances sont alors recherchées dans les documents, les instances candidates du concept *"ancre"* (ici, des noms de personne) sont extraites (par des techniques de détection d'entités nommées). Ensuite, les relations entre les instances candidates du concept *"ancre"* et les instances des concepts *"annexes"* sont vérifiées, en utilisant :

- la redondance de l'information sur le web. Plus il y a de documents présentant une co-occurrence des instances des concepts *"ancre"* et *"annexes"*, plus la relation entre eux est probable ;

- certaines techniques, dépendantes du domaine, permettent de vérifier la nature d'une relation entre une instance du concept *"annexe"* et une instance du concept *"ancre"*. Par exemple, l'utilisation de DBLP [3] permet de trouver des instances d'*Article* et de *Person* reliées par la relation *auteur*. D'autres concepts *"annexes"* permettent d'apporter des indices supplémentaires sur la classification d'une instance du super-concept du concept recherché comme étant une instance du concept recherché. Par exemple, la recherche de titres académiques *Phd*, comme concept *"annexe"*, apporte un indice supplémentaire sur la classification d'une personne en tant que professeur. Si les différents indices recueillis sont suffisamment nombreux et/ou sûrs pour reconnaître une instance du concept recherché, cette instance est alors annotée.

Armadillo utilise différentes techniques d'extraction basées sur la structure et/ou des patterns syntaxiques (Amilcare [CDWP03]) et intègre d'autres sources comme les résul-

3. Digital Bibliography Library Project

tats de recherche mot-clé sur le web et/ou sites spécialisés. Armadillo suppose que l'on dispose de techniques permettant de reconnaître des relations entre instances. Il suppose également que les instances à extraire sont redondantes sur le web et qu'elles sont référencées par plusieurs sources.

Comme dans Armadillo, le point de départ de notre approche est une ontologie lexicale comportant des informations sûres. L'utilisation de la redondance n'est envisageable que pour certains concepts.

KnowItAll

KnowItAll [ECD+05] est un un système automatique et non supervisé d'extraction d'informations à partir du Web. Il permet d'extraire des instances de concepts donnés en entrée en utilisant des patterns lexico-syntaxiques indépendants du domaine introduits par Hearst et décrits en section 3.1.4. Ainsi, en utilisant les patterns et le concept, plusieurs requêtes sont générées et soumises à un moteur comme Google et Yahoo. Cette extraction utilise des expressions régulières basées sur les étiquettes morpho-syntaxiques pour identifier des groupes nominaux, sur la syntaxe (casse des caractères) et lexicale et est focalisée sur les entités nommées. KnowItAll peut également appliquer ces patterns sur une base d'exemples, afin de déterminer la confiance que l'on peut avoir en chacun de ces patterns, et ne retenir une instance que si elle est extraite par plusieurs patterns linguistiques dont la combinaison permet d'avoir une confiance suffisante.

Une fois ces instances extraites, la couverture de KnowItAll est améliorée en utilisant diverses techniques : apprentissage de patterns d'extraction spécifiques au domaine, extraction de sous-concepts et extraction de listes :

- L'apprentissage de patterns d'extraction spécifiques au domaine se fait en repérant dans les documents les différentes occurrences des instances extraites dans une fenêtre appelée le contexte d'apparition de l'instance. KnowItAll utilise l'information mutuelle PMI (Pointwise mutual information) entre mots des différents contextes pour trouver des discriminants pour un concept. Par exemple *stared in* peut être un pattern spécifique pour le concept *Film* qui permet de rechercher d'autres instances film.

  Les meilleurs patterns construits de ces contextes d'apparition sont utilisés comme patterns d'extraction de nouvelles instances.

- L'extraction de sous-concepts est aussi réalisée en utilisant les patterns lexico-syntaxiques de Hearst. Par exemple, l'application du pattern *such as* sur l'expression *scientists, such as mathematicians, physicists and chemists* permet d'identifier *Mathematician*, *Physicist* et *Chemist* comme des sous-concepts du concept *Scientist*. Il devient alors possible de trouver que *Leibniz* et *Lambert* sont des instances de *Scientist* grâce à la reconnaissance du sous-concept *Mathematician* dans la phrase *... formal logic in a symbolic or algebraic way were made by some of the more philosophical mathematicians, such as Leibniz and Lambert*.

La distinction entre sous-concept et instance n'est pas claire et les domaines d'application choisis (*Scientist*, *City* et *Film*) simplifient le problème, les instances sont des noms propres (techniques NER) et les sous-concepts sont des noms communs. Cette extraction peut être améliorée en utilisant des ressources lexicales comme Wordnet pour valider la relation d'hyponymie.

- L'extraction de listes : une fois que les différentes instances d'une classe ont été identifiées, il est possible de rechercher des documents dans lesquels plusieurs de ces instances apparaissent. Si ces différentes instances apparaissent sous forme de listes ou de tableaux, il est alors possible, par induction de découvrir d'autres instances du même concept.

De la même manière qu'Armadillo, KnowItAll utilise la redondance de l'information sur le Web. La principale différence est que Knowitall n'utilise pas des concepts *"ancres"* pour extraire les instances mais les patterns de Hearst.

CPankow

CPankow (Context-driven and Pattern-based Annotation through Knowledge on the Web) [CLS05] est une autre approche automatique et non supervisée permettant d'annoter des instances de concepts dans des documents. CPANKOW utilise la redondance sur le Web des instances candidates, extraites des documents, pour découvrir les concepts à associer à ces instances.

Ainsi, si l'on ne sait pas comment annoter le mot *Niger* extrait dans un document, comme dans KnowItAll, un ensemble de requêtes est généré en utilisant les patterns de Hearst et l'instance candidate (et non le concept comme dans KnowItAll). Ces requêtes sont ensuite posées à un moteur de recherche tel que Google. Par exemple pour quatre patterns, les expressions extraites des premiers documents réponses sont : *"Niger is a country"*, *"Niger is a state"*, *"Niger is a river"* and *"Niger is a region"*.

Même si l'expression indiquant que *"Niger"* est un pays est plus fréquente sur le Web, cet emploi n'est peut être pas adapté au contexte du document traité. Aussi, l'approche utilise le contexte du document pour choisir les meilleurs concepts.

Plus précisément, la méthode comporte plusieurs étapes :

- L'extraction des instances candidates à partir des documents qui utilise des patterns indépendants du domaine issus des techniques de détection d'entités nommées.

- La construction d'un ensemble de requêtes qui utilise des patterns de Hearst et une instance candidate.

- L'extraction des concepts candidats à partir des réponses fournies par le moteur de recherche qui donne une liste de concepts candidats, en tenant compte de la distribution de l'instance candidate sur le web.

- L'annotation sémantique qui consiste à sélectionner les meilleurs concepts en tenant

compte du contexte d'appartion de l'instance candidate (utilisation du TFIDF entre le document et les documents réponses).

Dans notre approche, comme dans CPankow, les instances candidates sont extraites à l'aide de patterns indépendants du domaine. En revanche, nous distinguons les patterns pour les termes candidats et les patterns pour les entités nommées candidates.

Par ailleurs, pour associer une instance candidate à un concept, soit nous utilisons directement l'ontologie lexicale soit nous utilisons le même principe que CPankow pour extraire les concepts candidats. A la différence avec CPankow, dans notre approche les concepts retenus correspondent non seulement au contexte mais doivent aussi être rattachés aux concepts de l'ontologie.

	Lixto	Ontominer	LP2	TextRunner	Armadillo	KnowItAll	C-PANKOW	KIM
Automaticité	semi	x	semi	x	semi	x	x	x
Supervision	x	-	x	x	-	-	-	-
Patterns structurels	x	forte régularité	-	-	-	possible	-	-
Patterns lexico-syntaxiques	-	-	x	x	x	x	x	x
Patterns dépendants du domaine	-	-	-	-	x	-	-	-
Ontologie	-	-	x	-	x	x	x	x
Autres ressources lexicales ou KB	-	-	-	-	x	possible	-	possible
Web	-	-	-	-	-	x	x	-
Intervention de l'utilisateur	x	-	x	-	-	-	-	-

TABLE 1.1 – Approches d'extraction et d'annotation

1.2 Langages de représentation et d'interrogation de ressources Web

Dans cette section nous présentons brièvement les langages de représentation de ressources RDF, RDFS, OWL et SKOS recommandés par le W3C et le langage d'interrogation SPARQL pour le Web Sémantique.

1.2.1 Pile du Web sémantique

D'après la vision de Tim Berners-Lee du Web sémantique, l'information pourra être accessible et compréhensible non seulement par les humains mais aussi par les machines. Dans cette vision, le Web sémantique peut être considéré comme une pile de langages représentée dans la figure 1.1. Cette pile distingue principalement des langages dédiés à la structuration des ressources Web (XML, XML Schema) et la représentation de la sémantique qui peut être associée à ces ressources décrites par des ontologies et des règles logiques. La pile est structurée sur trois niveaux principaux [BL98] :

- ***Niveau d'adressage et de nommage***. Ce niveau est représenté par le standard d'adressage des ressources du Web URI (Universal Resource Identifier) et la norme Unicode pour le codage des caractères.

- ***Niveau syntaxique***. Ce niveau syntaxique est représenté par la définition des espaces de noms qui permettent d'identifier les ressources du Web, le langage XML, XML schéma et le langage de requêtes XML Query.

- ***Niveau Sémantique***. Ce niveau est représenté d'une part par les langages de représentation d'ontologies RDF/RDFS et OWL, et d'autre part par les langages de règles, de logique, de preuves et de confiance (trust).

Les langages du Web doivent permettre de :

- décrire les données, les schémas et leur sémantique (RDF/S et OWL) ;

- échanger des métadonnées et des schémas (eXtensible Markup Language / Schema ou XML/S) ;

- interroger les documents par les annotations (SPARQL).

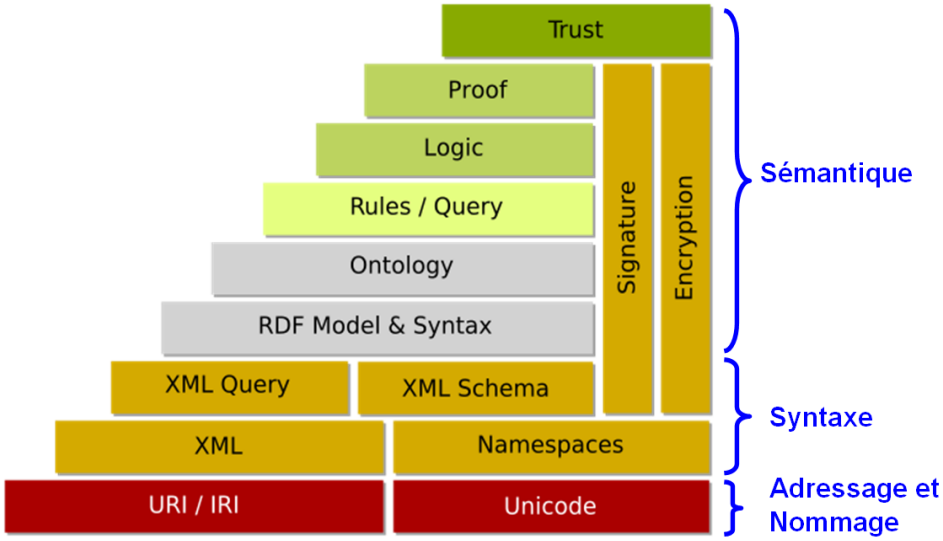


FIGURE 1.1 – Piles du Web Sémantique (Tim Berners-Lee [BL98])

1.2.2 Ressource Description Framework / Schema (RDF/RDFS)

Le langage RDF est un pilier pour les langages du Web Sémantique [W3Cb]. RDF permet d'annoter des ressources[4] du Web avec des métadonnées qui peuvent être définies en utilisant les schémas RDFS. Le modèle de données RDF est constitué d'un triplet composé trois éléments :

- *Sujet* : il représente la *ressource* à décrire. Il est nécessairement identifié par une URI.
- *Prédicat ou Propriété* : il s'agit d'une propriété utilisée pour caractériser et décrire le sujet. Un prédicat est nécessairement identifié par une URI.
- *Objet* : représente une donnée de type primitif (numérique, littéral, etc.) ou une autre ressource identifiée par une URI.

Un ensemble de déclarations RDF peut être représenté en utilisant une notation N-TRIPLE (*Sujet*, *Prédicat*, *Objet*), ou la syntaxe RDF/XML ou encore les graphes RDF.

La notation XML/RDF ci-dessous décrit une ressource (*http ://www.example.org/staffid/85740*) par la propriété *http ://www.example.org/terms/address* et cette dernière est décrite par les propriétés *http ://www.example.org/terms/city*, *http ://www.example.org/terms/street*, *http ://www.example.org/terms/state* et *http ://www.example.org/terms/postalCode*.

4. [*WikiPedia*] Toute chose ou entité susceptible d'être identifiée, nommée, manipulée à travers ses représentations, par quelque moyen que ce soit, sur le Web en général ou dans n'importe quel système d'information utilisant les technologies du Web.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF
      xmlns:cls="http://www.example.org/#"
      xmlns:prp="http://www.example.org/terms/#">

<rdf:Description rdf:about="cls:staffid/85740">
      <prp:address rdf:resource="cls:addressid/85740"/>
</rdf:Description>
<rdf:Description rdf:about="cls:addressid/85740">
      <prp:city>Bedford</prp:city>
      <prp:street>1501 Grant Avenue</prp:street>
      <prp:state>Massachausetts</prp:state>
      <prp:postalCode>01730</prp:postalCode>
</rdf:Description>

</rdf:RDF>
```

La figure 1.2 décrit le même exemple en utilisant les graphes RDF.

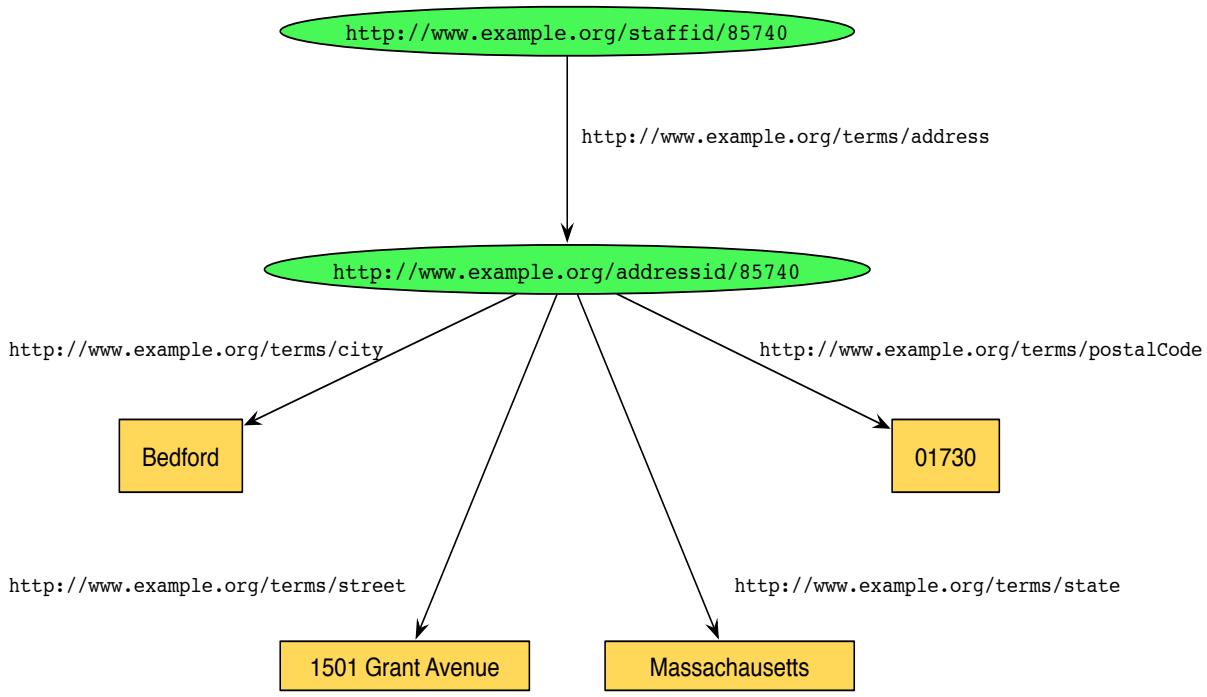


FIGURE 1.2 – Exemple de graphe RDF, W3C [W3Cb]

RDF Schema [W3Cc] permet de définir en RDF un schéma pour les métadonnées annotant les ressources. Plus précisément RDFS permet de définir :

- un ensemble de concepts ou classes (*rdfs :Class*) et leur hiérarchie (*rdfs :subClassOf*) ;
- le domaine (*rdfs :domain*) et le co-domaine (*rdfs :range*) d'une propriété ;
- une hiérarchie de propriétés (*rdfs :subPropertyOf*) ;

Il permet aussi d'utiliser des propriétés prédéfinies comme *rdfs :label*, *rdfs :seeAlso*, *rdfs :comment*, etc.

Prenons l'exemple de la figure 1.3 où nous avons une propriété *address* qui représente une adresse, on peut déclarer que cette propriété s'applique à une ressource de type *Staff* et qu'une valeur de cette propriété référence une ressource de type *Adresse*. De la même manière les propriétés *street* et *postalCode* sont définies et s'appliquent à des ressources de type *Adresse* et leurs valeurs sont de type *Literal*.

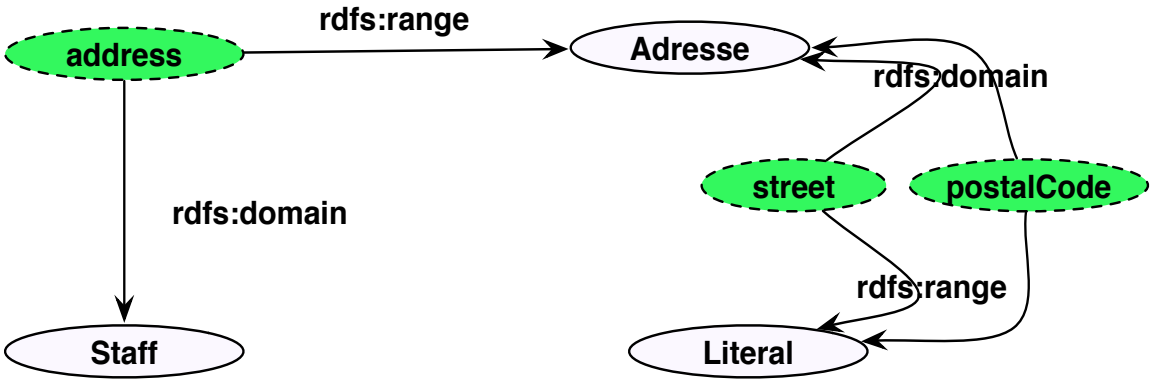


FIGURE 1.3 – Exemple de schéma RDF

La sémantique en logique du premier ordre des déclarations RDFS de subsomption entre classes, de subsomption entre propriétés et de signature des propriétés est présentée dans le tableau 1.2.2.

Constructeurs	**N-TRIPLE**	**Sémantique LP**
Subsomption entre Classes	C_1 *rdfs :subClassOf* C_2	$\forall X(C_1(X) \Rightarrow C_2(X))$
Subsomption entre Propriétés	P_1 rdfs :subPropertyOf P_2	$\forall X\ \forall Y(P_1(X,Y) \Rightarrow P_2(X,Y))$
Typage du domaine d'une propriété	P rdfs :domain C	$\forall X\ \forall Y(P(X,Y) \Rightarrow C(X))$
Typage du co-domaine d'une propriété	P rdfs :range C	$\forall X\ \forall Y(P(X,Y) \Rightarrow C(Y))$

TABLE 1.2 – Sémantique des constructeurs RDFS en logique du premier ordre

1.2.3 Ontology Web Langage (OWL)

Le niveau d'expressivité du langage RDFS est insuffisant pour représenter des ontologies riches et complexes. OWL étend le langage RDFS pour exprimer des contraintes sur des classes et des propriétés. Par exemple on peut exprimer :

- des contraintes de cardinalité sur les valeurs ou le type d'une propriété : *owl :minCardinality*, *owl :maxCardinality* ;
- des propriétés fonctionnelles ou inverses fonctionnelles : *owl :FunctionalProperty* et *InverseFunctionalProperty*, symétriques *owl :SymmetricProperty*, transitives *TransitiveProperty* ;
- des contraintes de cohérence entre classes : équivalence *owl :equivalentClass*, disjonction *owl :disjointWith* ;
- des contraintes de mise en correspondance entre instances *owl :sameAs* et *owl :differentFrom* ;
- de nouveaux constructeurs à partir de classes existantes par intersection *owl :intersectionOf*, union *owl :unionOf* ou complément *owl :complementOf*.

Selon son niveau d'expressivité, OWL est décomposé en trois familles de langages :

- Le langage *OWL Lite* permet de définir une hiérarchie de classes et des contraintes simples. Par exemple, bien que le langage gère les contraintes de cardinalité, il n'admet que les valeurs de cardinalité 0 ou 1. Les constructeurs tels que *owl :unionOf*, *owl :intersectionOf* ne font pas partie de *OWL Lite*.
- Le langage *OWL DL*, dont le nom est hérité des logiques de description, est l'un des nombreux sous-ensembles décidables de *OWL*. Il Le niveau d'expressivité de *OWL DL* garantit la complétude et la décidabilité des algorithmes de raisonnement. *OWL DL* étend *OWL Lite*, il permet par exemple l'utilisation des contraintes de cardinalité autre que 0 et 1 et de constructeurs comme *owl :disjointWith*. Par ailleurs *OWL DL* nécéssite une séparation entre classes et instances, entre *owl :DataTypeProperty* et *owl :ObjectProperty*. De même, des contraintes ne peuvent pas être définies sur des propriétés transitives ou leurs inverses.
- Le langage OWL Full est destiné aux utilisateurs voulant une expressivité maximale. Le raisonnement dans ce cas est incomplet ou non décidable.

1.2.4 Simple Knowledge Organisation Systems (SKOS)

SKOS est un modèle de données permettant la représentation standard et la publication de vocabulaires structurés tels les thésaurus et les taxonomies. SKOS est construit sur

la base du langage RDF. Il s'agit d'une recommandation du W3C depuis août 2009. Il peut être utilisé seul ou combiné avec des langages tels que OWL(Ontology Web language).

SKOS définit une classe *Concept* de type *OWL :Class* à laquelle peuvent se rattacher de nombreuses propriétés. Un concept SKOS peut être décrit par :

- des littéraux appelés labels en distinguant le label préféré (*skos :prefLabel*) par langue des autres labels dits alternatifs (*skos :altLabel*) ;
  Exemple

```
ex:rocks rdf:type skos:Concept;
         skos:prefLabel "rocks"@en;
         skos:altLabel "basalt"@en;
         skos:altLabel "granite"@en;
         skos:altLabel "slate"@en.
```

- des identifiants spécifiques aux applications (*skos :notation*) ;
  Exemple

```
<Potassium> skos:notation "K"^^<ChemicalSymbolNotation> .
```

- des propriétés de documentation comme des notes et des définitions (*skos :note*, *skos :definition*, ...) ;
  Exemple

```
<Protein> rdf:type owl:Class ;
  skos:definition "A physical entity consisting of a sequ-
  ence of amino-acids; a protein monomer; a single polypept-
  ide chain. An example is the EGFR protein."@en .
```

- des relations avec d'autres concepts exprimant par exemple la généricité (*skos :broader* ou *skos :broaderTransitive*) ou la spécificité (skos :narrower ou skos :narrowerTransitive) d'un concept par rapport à un autre concept comme pour les termes d'un thésaurus.
  Exemple

```
ex:mammals skos:broader ex:animals.
ex:animals skos:narrower ex:mammals.
ex:renaissance skos:related ex:humanism.
```

Cette liste n'est pas exhaustive, mais elle présente les principaux éléments de *SKOS* et en particulier les propriétés $prefLabel$ et $altLabel$ que notre ontologie utilise.

1.2.5 Langage d'interrogation SPARQL

SPARQL est un langage de requête recommandé par le W3C depuis le 15 Janvier 2008. SPARQL est capable d'interroger des informations représentées par des graphe RDF.

Une requête SPARQL, dans sa forme élémentaire, est constituée d'un ensemble de patrons de triplets appelé un patron de graphe élémentaire (basic graph pattern). Les patrons de triplets sont comme les triplets RDF sauf qu'un sujet, un prédicat et un objet peuvent être des variables.

L'exemple ci-dessous montre une requête SPARQL permettant de rechercher le titre d'un livre dans le graphe de données RDF fourni. La requête comprend deux clauses : la clause *SELECT* identifie les variables qui doivent apparaître dans les résultats, et la clause *WHERE* définit le patron de graphe élémentaire à mettre en correspondance avec le graphe RDF de données. Le patron de graphe élémentaire dans cet exemple se compose d'un seul patron de triplet avec une seule variable (?title) en position d'objet.

- Données :

```
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
@prefix :     <http://example.org/book/> .
@prefix ns:   <http://example.org/ns#> .

:book1  dc:title  "SPARQL Tutorial" .
:book1  ns:price  42 .
:book2  dc:title  "The Semantic Web" .
:book2  ns:price  23 .
```

- Requête :

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1>
        <http://purl.org/dc/elements/1.1/title> ?title .
}
```

- Résultats :

```
"SPARQL Tutorial"
```

Les clauses *FILTER* du *SPARQL* permettent de restreindre les résultats à celles que l'expression de filtre évalue à vrai. La fonction de filtre comme regex peut tester des littéraux RDF. L'exemple ci-dessous montre une requête SPARQL permettant de rechercher le titre d'un livre commençant par la chaîne de caractères *"SPARQL"* dans le graphe de données RDF ci-dessus.

- Requête :

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
SELECT  ?title
WHERE   { ?x dc:title ?title
          FILTER regex(?title , "^SPARQL")
        }
```

- Résultats de la requête :

```
"SPARQL Tutorial"
```

Une requête SPARQL peut être complexe et comporter des clauses comme la clause *OPTIONAL* qui définit un patron de graphe optionnel par rapport au patron de graphe élémentaire. Plusieurs patrons peuvent être optionnels dans une requête.

Le moteur de recherche sémantique CORESE [CDKFZ04, CDKFZG05] a ajouté des extensions tels que le constructeur de patterns *UNION* et des clauses telles que *GROUP BY* et *ORDER BY*. Ces extensions seront dans SPARQL 1.1 Query Language[5].

Il existe des outils permettant d'interroger une base d'annotations RDF utilisant SPARQL. CORESE utilise les graphes conceptuels et propose différentes approximations utilisant soit la distance ontologique basée sur la relation de subsomption, soit la proximité contextuelle, soit sur la longueur du chemin séparant les éléments recherchés. CORESE propose un langage de règles permettant d'inférer de nouveaux faits.

Jena [McB01] est une autre API Java pour la construction d'applications pour le Web Sémantique. Il fournit un environnement de programmation pour RDF, RDFS, OWL et SPARQL. L'API de Jena inclut aussi un moteur d'inférence basé sur des règles.

Conclusion de l'état de l'art

Dans notre travail de thèse, notre objectif est d'annoter sémantiquement une collection de documents relatifs au même domaine et délimités par des balises (HTML, XML). Ces documents sont structurellement hétérogènes comportant des parties bien structurées et d'autres textuelles. Notre approche se veut automatique pour le passage à l'échelle et non supervisée en raison du nombre et de l'hétérogénéïté des documents.

Les instances de concepts étant noyées dans le texte de documents plus ou moins structurés sont difficiles à repérer et à délimiter précisément dans une approche complètement automatique. Ces instances ne sont pas représentées avec une structure régulière dans les documents et ne sont pas non plus décrites avec des expressions linguistiques qui se répètent. L'utilisation de méthodes d'extraction automatique de patterns spécifiques au domaine ou au corpus traité, qu'ils soient structurels ou lexico-syntaxiques, ne permettra pas de couvrir correctement les instances de concepts ou leurs relations.

Nous nous plaçons dans le contexte où nous disposons d'une ontologie de domaine définie par un ensemble de concepts, un ensemble de relations entre ces concepts et leurs propriétés. Cette ontologie comporte une composante lexicale, chaque concept est accompagné de plusieurs labels et d'un premier ensemble de termes ou entités nommées du domaine.

L'idée, dans notre travail, est de repérer des termes ou entités nommées de concepts dans les nœuds de l'arbre DOM (Document Object Model) représentant la structure d'un document (HTML ou XML). Pour avoir une bonne couverture, nous extrayons les termes ou les entités nommées candidats de manière indépendante du domaine et des documents. Ces termes ou entités nommées candidats sont ensuite rapprochés des termes ou des labels de l'ontologie lexicale afin de déterminer les concepts appropriés.

Dans ce contexte, nous avons plusieurs problèmes à résoudre liés à l'extraction des termes et des entités nommées candidats, au rapprochement avec l'ontologie lexicale, au fait que la composante lexicale ne soit pas suffisamment riche et aussi à la granularité de l'annotation.

Pour traiter l'extraction des termes candidats, nous utilisons des patterns syntaxiques

5. W3C Working Draft 14 October 2010, http ://www.w3.org/TR/sparql11-query/

représentant des syntagmes nominaux et pour les entités nommées nous utilisons des patterns lexico-syntaxiques indépendants du domaine.

Le rapprochement aux concepts de l'ontologie d'un terme ou d'une entité nommée candidat exploite les mesures de similarité souvent utilisées pour l'alignement d'ontologie. Ces mesures exploitent la similarité des chaînes de caractères, la catégorie grammaticale des mots et les relations taxonomiques des concepts. Par exemple, si "*European Conference*" est un terme candidat, et si "*Conference*" est un des labels ou termes du concept *Evenement*, et bien le terme "*European Conference*" peut être associé au concept *Evenement*.

Pour l'annotation, notre approche étant automatique, nous ne cherchons pas à délimiter parfaitement les instances de concepts. Ce sont les nœuds de documents, où des termes et des d'entités nommées associées aux concepts sont localisées, qui sont les instances de concepts. Cette annotation prend en compte la sémantique des termes ou des d'entités nommées et le voisinage structurel des nœuds les contenant.

En effet, Nous exploitons la structure non seulement pour délimiter les instances mais aussi pour détecter les éventuelles relations du domaine reliant ces instances.

Nous utilisons le Web pour chercher d'autres termes candidats ayant des relations d'hyponymie (is-a ou instanceOf) avec le terme ou l'EN candidat. De ce point de vue, la redondance de ce terme ou de l'entité nommée candidat sur le Web est primordiale pour renforcer l'apprentissage de nouveaux termes qui permettraient d'annoter les documents et aussi d'enrichir lexicalement l'ontologie.

Chapitre 2

PRÉSENTATION DE L'ARCHITECTURE DE NOTRE SYSTÈME

Sommaire

Introduction

Dans notre travail de thèse, nous avons défini une approche d'annotation de documents semi-structurés automatique, non supervisée et guidée par une ontologie à composante lexicale [TPB08, TBPL09, TBP07] et [MBPT10b, MBPT10a].

Étant donnée l'hétérogénéité des documents, les instances de concepts sont difficilement localisables de manière complète et précise dans une approche complètement automatique. Notre approche repère les termes et les entités nommées de concepts dans les nœuds de l'arbre DOM (Document Object Model) représentant la structure d'un document (HTML ou XML) et ce sont les nœuds qui sont annotés en tant qu'instances de

concepts. Nous exploitons, ensuite, la proximité des nœuds annotés pour déduire l'existence possible d'une relation sémantique de domaine et les annoter par une relation de voisinage, sans chercher à l'identifier précisément.

Nous avons défini une architecture utilisant les langages du W3C RDF(S) et SPARQL composée principalement 3 modules : *SHIRI-Extract*, *SHIRI-Annot* et *SHIRI-Querying*.

- *SHIRI-Extract* : est le module qui extrait des termes et des entités nommées (EN) dans les documents, qui les associe aux concepts de l'ontologie et qui enrichit la composante lexicale de l'ontologie. Ce module s'appuie sur des patterns syntaxiques et lexico-syntaxiques indépendants du domaine permettant d'extraire des entités nommées et des termes de type groupe nominal. Il utilise un algorithme appelé *Extract-Align* que nous avons défini permettant d'enrichir l'ontologie chaque fois qu'un terme ou une entité nommée candidat a pu être aligné, c'est à dire, associé à un concept de l'ontologie. L'alignement se fait soit directement avec l'ontologie soit en utilisant le Web pour découvrir de nouveaux termes.

- *SHIRI-Annot* : est le module qui annote les nœuds de documents conformément au modèle d'annotation que nous avons défini. Ce modèle est généré automatiquement à partir de l'ontologie pour représenter les résultats d'extraction et d'annotation. Pour annoter les nœuds de documents, nous avons défini un ensemble de règles exprimées en logique du premier ordre.

- *SHIRI-Querying* : est le module qui permet de poser des requêtes sur la base des annotations, ces requêtes sont formulées à l'aide des métadonnées définies dans le modèle d'annotation.

Ce chapitre est organisé comme suit. D'abord, dans la section 1, nous présentons formellement l'ontologie à composante lexicale. Dans la section 2, nous présentons le modèle d'annotation que nous avons défini pour représenter l'ontologie et les annotations relatives aux documents. Dans la section 3, nous présentons l'architecture de notre système et nous détaillons les entrées et les sorties des modules *SHIRI-Extract*, *SHIRI-Annot*. Dans la section 4, nous détaillons un scénario d'usage du système.

2.1 Ontologie à composante lexicale

2.1.1 Définitions

Une ontologie est une formalisation explicite d'une conceptualisation commune et partagée [Gru93].

Dans notre travail, l'ontologie que l'on considère est définie par un ensemble de concepts, de relations entre ces concepts et de propriétés. Elle est à composante lexicale [RTAG07], chaque concept est caractérisé par un ensemble de labels et un ensemble de termes. Ce lexique est enrichi par le module *SHIRI-Extract*. Plus formellement, soit $\mathcal{O}(\mathcal{C}_\mathcal{O}, \mathcal{R}_\mathcal{O}, \preceq$

, $\mathcal{S}_\mathcal{O}, \mathcal{X}_\mathcal{O}, \mathcal{L}^{ex}_\mathcal{O}$) l'ontologie de domaine où :

- $\mathcal{C}_\mathcal{O}$ est l'ensemble des concepts,
- $\mathcal{R}_\mathcal{O}$ est l'ensemble des relations entre les instances de concepts,
- $\preceq$ désigne la relation de subsomption entre les concepts et entre les relations,
- $\mathcal{S}_\mathcal{O}$ définit le domaine et le co-domaine (signature) de chaque relation,
- $\mathcal{X}_\mathcal{O}$ est l'ensemble d'axiomes du domaine définis sur les concepts et les relations,
- $\mathcal{L}^{ex}_\mathcal{O}(\mathcal{L}, \mathcal{T}, prefLabel, altLabel, hasTerm, hasTermNe)$ définit :
  - $\mathcal{L}$ est l'ensemble des labels de concepts. L'ensemble des labels d'un concept est un ensemble de termes considérés comme équivalents (synonymie dans le domaine) pour représenter un concept.
  - $\mathcal{T}$ est l'ensemble des termes et entités nommées décrivant les instances des concepts du domaine. L'ensemble des termes d'un concept est un ensemble de termes plus spécifiques que les labels.
  - Chaque concept $c \in \mathcal{C}_\mathcal{O}$ est lié à un label préféré $l \in \mathcal{L}$ via la propriété $prefLabel$ [6] et aux labels alternatifs $l' \in \mathcal{L}$ via la propriété $altLabel$[7].
  - Chaque concept $c \in \mathcal{C}_\mathcal{O}$ est relié à un terme via la propriété $hasTerm$ et à une entité nommée via la propriété $hasTermNe$. En effet, un terme candidat peut être extrait soit par un pattern spécifique pour les entités nommées soit par un pattern pour les termes.

L'ontologie que notre système exploite pour l'annotation ne décrit que les relations qui ont comme co-domaine des concepts et non des littéraux. Si l'ontologie fournie par l'expert comporte de telles relations (attributs), nous transformons ce littéral en concept car notre approche ne cherche à repérer que des instances de concepts décrits par des labels, des termes ou entités nommées.

Dans notre approche, les ensembles $\mathcal{L}$ et $\mathcal{T}$ sont respectivement initialisés par un ensemble de labels et un ensemble de termes choisis par l'expert du domaine en utilisant, par exemple, WordNet.

L'ensemble $\mathcal{T}$ est automatiquement enrichi par le module *SHIRI-Extract*. L'ensemble $\mathcal{L}$ n'est enrichi que manuellement par l'expert qui valide les termes en modifiant ou en supprimant certains ou en les associant à un concept plus spécifique ou plus général. Il peut également décider que certains termes deviennent des labels.

6. Propriétés définies dans SKOS : Simple Knowledge Organization System

2.1.2 Exemple

Les figures 2.1 et 2.2 décrivent le graphe RDF(*Resource Description Framework* [W3Cb]) de l'extrait d'une ontologie de domaine portant sur les appels à participation dans des conférences dédiées au domaine informatique. Elle comprend des concepts comme *topic* (thème), et des concepts comme *event* (évènement). Ces concepts sont reliés par des propriétés telle que *hasTopic* dont le domaine est *CallForPaper* (appel à participation) et le co-domaine est *Topic*.

Les labels et les termes sélectionnés initialement par l'expert pour le concept *Topic* sont : {*topic*} pour le label préféré, {*field*, *area*, *theme*} pour les alternatives et {*communications protocols*, *data encryption*, *information*, *object-oriented programming language*} pour les termes (figure 2.2).

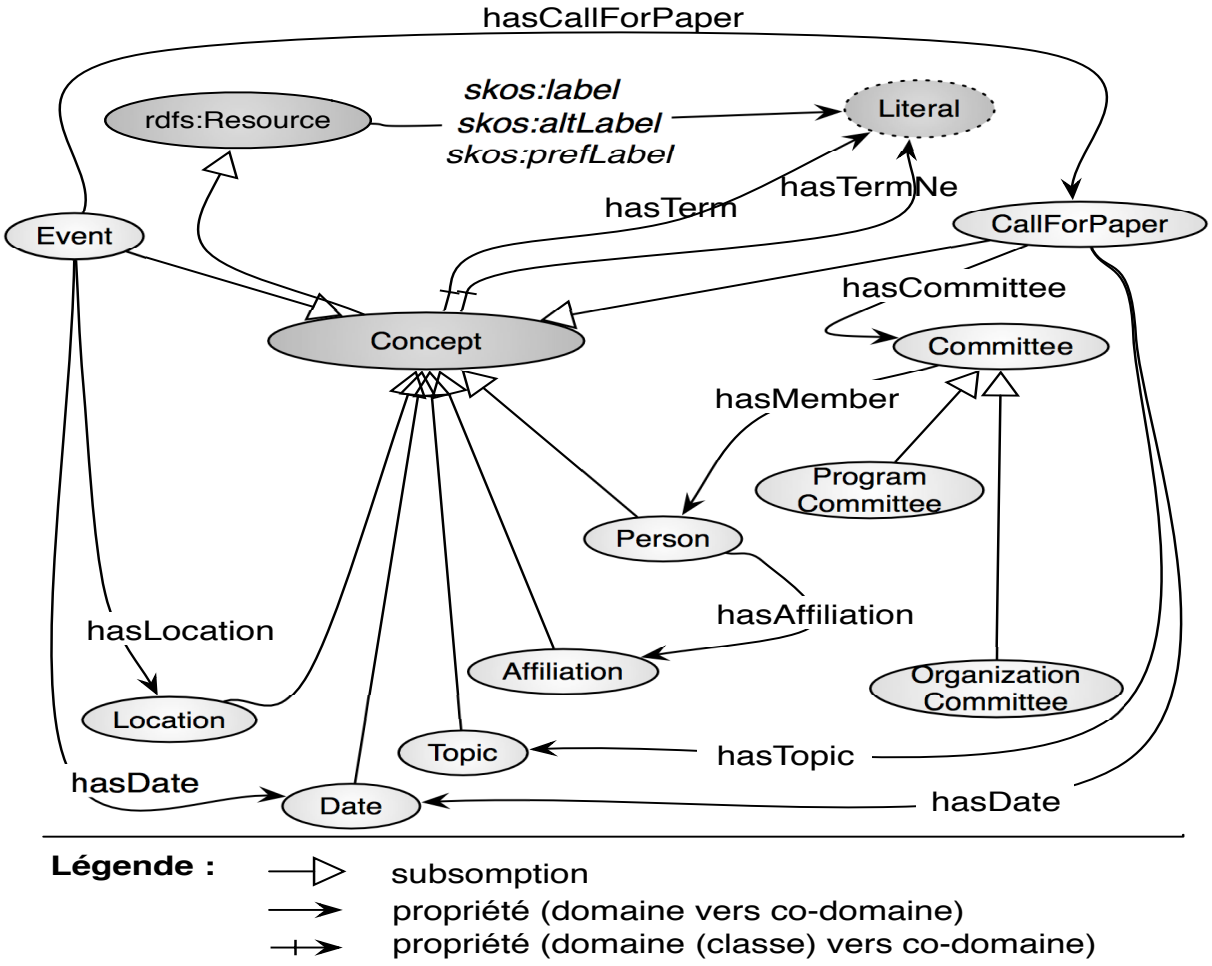


FIGURE 2.1 – Extrait du graphe RDF de l'ontologie d'appel à participation à des conférences

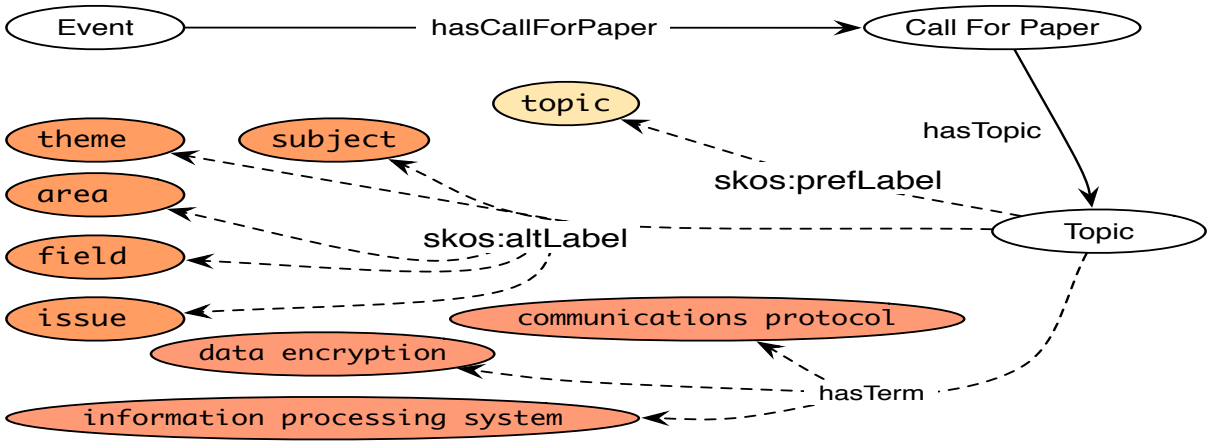


FIGURE 2.2 – Exemples de labels et de termes du concept *Topic*

2.2 Modèle d'annotation

Dans notre travail, les métadonnées d'annotation sont définies dans le modèle d'annotation décrit dans la figure 2.3.

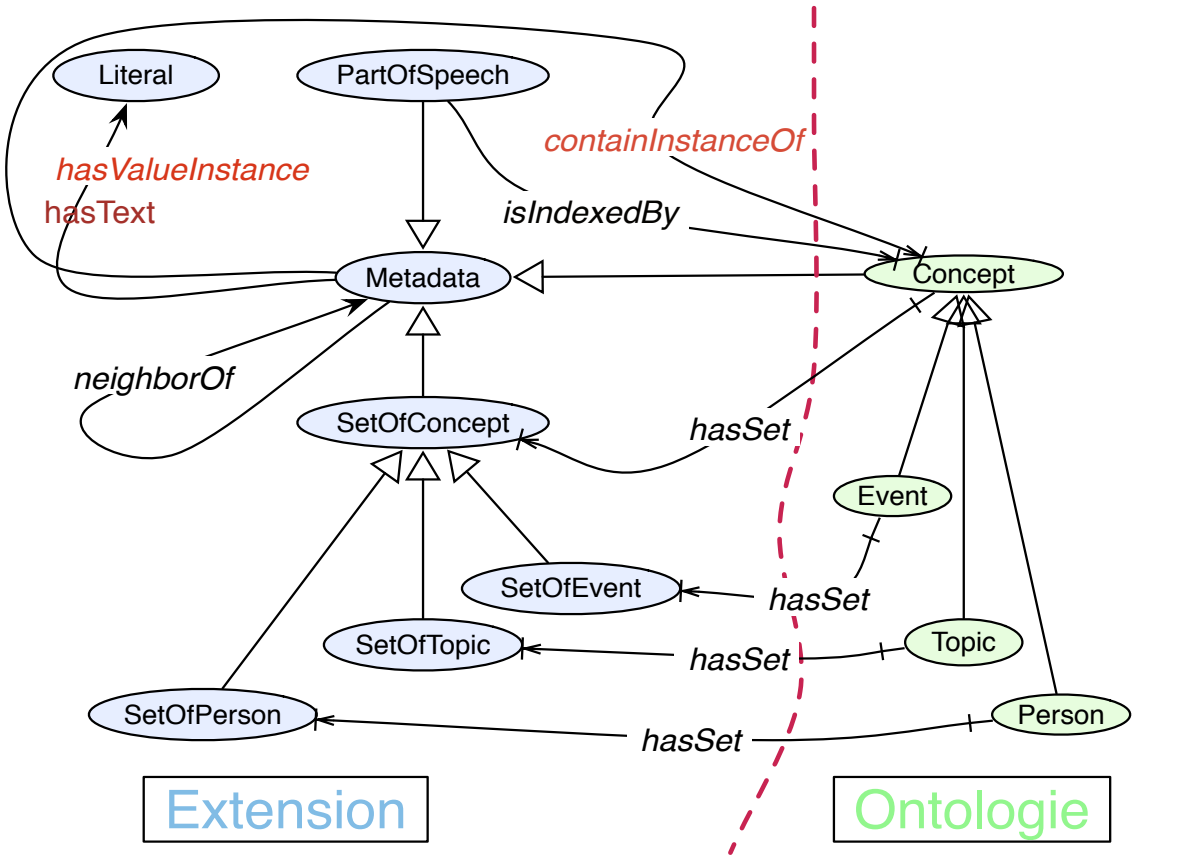
Rappelons qu'une métadonnée est définie comme une donnée définissant une autre donnée, pouvant être associée à des éléments définis dans une ontologie. Dans notre cas, en plus des métadonnées dont la sémantique est définie dans l'ontologie, nous avons défini un ensemble de métadonnées spécifiques pour le module d'annotation *SHIRI-Annot*. Celles-ci expriment l'agrégation et la proximité des instances repérées dans les documents. L'ensemble des concepts et relations du domaine et d'agrégation sont réunis dans le modèle d'annotation. Ces métadonnées sont générées automatiquement à partir de l'ontologie du domaine.

Plus formellement, le quintuplet $\mathcal{A}(\mathcal{C}_\mathcal{A}, \mathcal{R}_\mathcal{A}, \preceq, \mathcal{S}_\mathcal{A}, \mathcal{X}_\mathcal{A}, \mathcal{L}_\mathcal{O}^{ex})$ désigne le modèle d'annotation où :

- $\mathcal{C}_\mathcal{A} = \mathcal{C}_\mathcal{O} \cup \mathcal{C}_\mathcal{S}$, $\mathcal{C}_\mathcal{S}$ est l'ensemble des concepts définis pour la tâche d'annotation,
- $\mathcal{R}_\mathcal{A} = \mathcal{R}_\mathcal{O} \cup \mathcal{R}_\mathcal{S}$, $\mathcal{R}_\mathcal{S}$ est l'ensemble des propriétés définies pour la tâche d'annotation,
- $\mathcal{X}_\mathcal{A} = \mathcal{X}_\mathcal{O} \cup \mathcal{X}_\mathcal{S}$, $\mathcal{X}_\mathcal{S}$ est un ensemble de règles décrites en logique du premier ordre et définies pour le module *SHIRI-Extract*.

Nous avons défini dans l'ensemble $\mathcal{C}_\mathcal{S}$ les concepts $Metadata$, $Concept$, $PartOfSpeech$, $SetOfConcept$ et tous les concepts $SetOfc$ générés pour tout $c \in \mathcal{C}_\mathcal{O}$ comme des concepts descendants de $SetOfConcept$.

$Metadata$ est défini comme un super-concept direct des concepts $Concept$, $SetOfConcept$ et $PartOfSpeech$. Tous les concepts racines $\in \mathcal{C}_\mathcal{O}$ de l'ontologie sont des sous-concepts directs de $Concept$.



Légende :

- subsomption
- propriété (domaine vers co-domaine)
- propriété (domaine (classe) vers co-domaine (classe))
- propriété (domaine vers co-domaine (classe))

FIGURE 2.3 – Modèle d'Annotation

Génération des concepts *SetOfc* subsumés par *SetOfConcept* : La génération des concepts *SetOfc* est réalisée comme suit :

- Pour chaque concept ontologique $c \in \mathcal{C}_\mathcal{O}$ est généré le concept $SetOfc$.
- Tout concept $SetOfc$ généré, tel que c est sous-concept direct de $Concept$, est sous-concept direct de $SetOfConcept$.
- Tout concept $SetOfc'$ généré, tel qu'il $\exists c \in \mathcal{C}_\mathcal{O}$ et c' est sous-concept direct de c, est sous-concept direct de $SetOfc$.
- La propriété *hasSet* dont le domaine est la classe du concept *Concept* et le co-domaine est la classe du concept *SetOfConcept* est instanciée pour chaque c et $SetOfc$ correspondant.

L'exemple représenté dans la figure 2.4 illustre la génération de $SetOfc$.

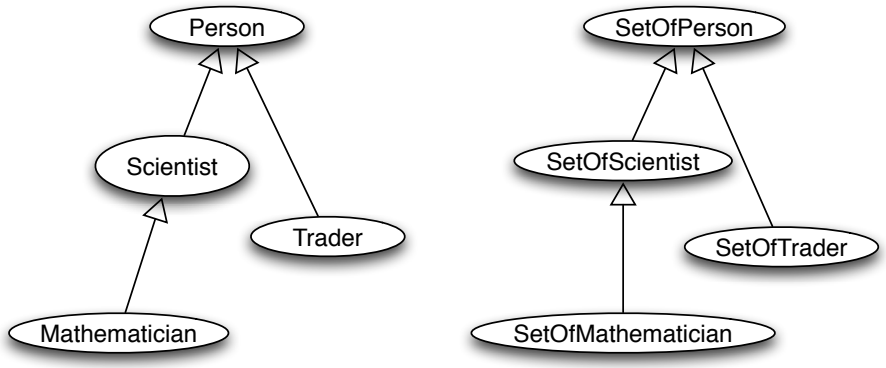


FIGURE 2.4 – Exemple de génération des concepts *SetOfc*

Granularité de l'annotation (Nœud) : La structure d'un document est représentée par l'arbre DOM (Document Object Model, une recommandation du W3C). Chaque nœud de l'arbre représente un élément de structure (une balise), et possède un nœud fils contenant son texte. Les nœuds sont reliés par des liens père/fils qui repésentent leur imbrication arborescente. La figure 2.5 décrit un exemple d'une repésentation DOM d'un document HTML.

Dans notre approche, la granularité d'annotation est le nœud. En d'autres termes, une instance d'un concept défini dans le modèle d'annotation est un nœud (cf. 2.3) identifié par son URI (Universal Resource Identifier). Les nœuds annotés sont tous des instances du concept $Metadata$. De plus, une instance est soit une instance d'un concept ontologique $c \in \mathcal{C}_\mathcal{O}$ ou d'agrégation $c \in \mathcal{C}_\mathcal{S}$.

La propriété *hasText* associe à un nœud son texte.

Dans ce qui suit, nous définissons les éléments des ensembles de $\mathcal{C}_\mathcal{S}$ et de $\mathcal{R}_\mathcal{S}$.

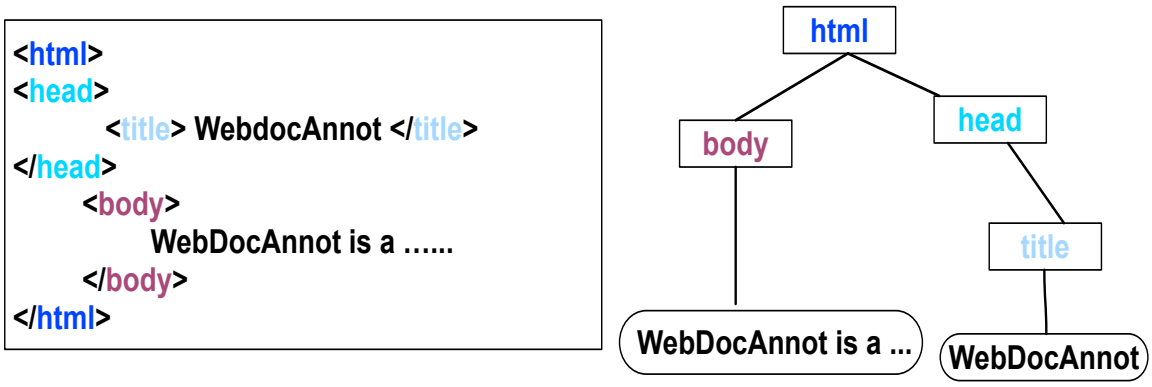


FIGURE 2.5 – Exemples de document DOM

2.2.1 Le concept *PartOfSpeech*

Comme souligné plus haut, les termes et les ENs relatifs aux concepts ontologiques sont souvent noyés dans les parties non structurées et sont ainsi difficiles à repérer. Un nœud de document peut comporter des termes et des ENs de concepts différents. Le concept $PartOfSpeech$ est défini pour annoter un nœud de document dont le texte contient plusieurs termes ou ENs alignés avec des concepts différents. Une instance de type $PartOfSpeech$ est reliée via la propriété $isIndexedBy$ aux concepts des termes ou ENs repérés dans ce nœud.

Prenons l'exemple d'un nœud contenant le texte *"ECAI 2008, the 18th conference in this series, is jointly organized by the European coordinating Committee on Artificial Intelligence the university of Patras and the Hellenic Artificial Intelligence Society"* où des termes ou EN alignés avec les concepts *Date* (date), *Topic* (thème), *Event* (événement) et *Location* (lieu) sont localisés. Ce nœud est donc annoté comme instance du concept $PartOfSpeech$ et est reliée via la propriété $isIndexedBy$ aux concepts *Date*, *Topic*, *Event* et *Location*.

2.2.2 Le concept *SetOfConcept*

Avant de définir les métadonnées *SetOfConcept* et ses concepts descendants *SetOfc*, nous définissons les concepts comparables.

Concepts comparables : C un est ensemble de concepts comparables si $\forall c_i \in C$ et $\forall c_j \in C$, $subClassOf(c_i, c_j) \vee subClassOf(c_j, c_i)$. Un exemple de concepts comparables est représenté dans la figure 2.6.

Un nœud de document, comportant des termes ou ENs de concepts comparables appartenant à un ensemble C de concepts comparables, est annoté comme une instance de *SetOfc* tel que c est le concept ancêtre des concepts de C.

Par exemple, le concept ontologique *Topic* défini dans 2.1 est relié via la propriété *hasSet* au concept $SetOfTopic$ qui est un sous-concept direct de $SetOfConcept$. Le nœud

contenant le texte *"searching, querying, visualizing, navigating and browsing the semantic web"*, où plusieurs termes du concept $Topic$ sont localisées, est annoté par $SetOfTopic$.

Un autre exemple de nœud où plusieurs entités nommées de concepts comparables *scientist* et son super-concept $Person$ sont repérés dans le texte, ce nœud est annoté en tant qu'instance de $SetOfPerson$.

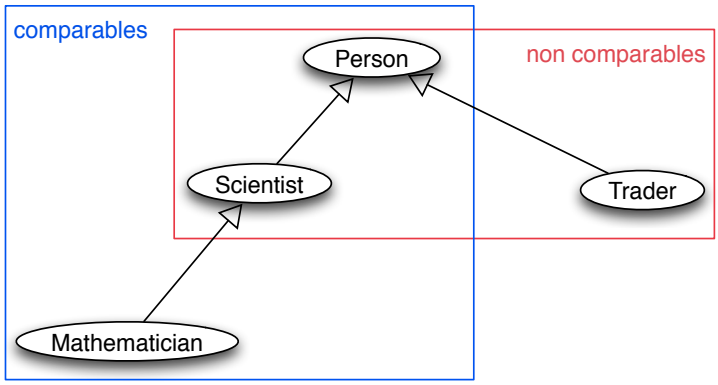


FIGURE 2.6 – Exemples de concepts comparables et non comparables

2.2.3 La propriété *neighborOf*

Vu l'hétérogénéité des documents, l'extraction des relations du domaine peut s'avérer encore plus difficile que pour les instances de concepts. Le principe de notre approche est d'utiliser les nœuds annotés, leur sémantique et leur proximité structurelle pour déduire une relation entre instances, plus précisément, entre nœuds.

Le voisinage structurel est représenté par le chemin de liens entre nœuds de l'arbre structurel DOM du document.
La relation $neigbhorOf$ est instanciée pour deux nœuds de type $Metadata$ reliés par un chemin structurel de longueur ne dépassant pas une valeur déterminée d. Les deux nœuds doivent contenir respectivement des instances de concepts c_i et $c_j \in \mathcal{C}_\mathcal{O}$ tels que $\exists r \in \mathcal{R}_\mathcal{O}$ où $c_i \in domaine(r)$ et $c_j \in co\text{-}domaine(r)$. Cette relation est symétrique.

La relation $neighborOf$ est instanciée quand la distance est nulle. Dans ce cas, les instances se trouvent dans le même nœud et ce nœud est de type $PartOfSpeech$.

2.2.4 Les métadonnées de pré-annotation

Nous avons également défini deux propriétés de pré-annotation pour représenter les résultats obtenus par le module *SHIRI-Extract*.

- La propriété $containInstanceOf$ est définie entre un nœud et un concept $c_i \in \mathcal{C}_\mathcal{O}$, pour exprimer le type du terme ou de l'entité nommée localisée dans ce nœud.

- De la même manière, la relation *hasValueInstance* est définie entre un nœud et un littéral pour conserver le terme ou l'entité nommée (littéral) localisé dans ce nœud.

2.3 Architecture de notre système

Dans cette section, nous présentons les modules définis dans notre système, à savoir, la constitution du corpus, l'extraction *SHIRI-Extract* et l'annotation *SHIRI-Annot* (figure 2.7). Notre système utilise les langages du W3C [W3Ca] : RDF/RDFS pour la représentation du modèle d'annotation et les documents annotés et SPARQL pour leur interrogation.

2.3.1 Constitution du corpus

La constitution du corpus se fait en deux étapes : (i) chargement du corpus portant sur le domaine d'intérêt et (ii) pré-traitement du corpus.

Chargement du corpus

Le chargement du corpus peut se faire automatiquement en utilisant un moteur de recherche comme Google ou Yahoo. Pour ce faire nous identifions un ensemble de mots clé appelés graines (seed en anglais). Ces mots sont combinés pour former des n-uplets qui sont fournis au moteur de recherche pour obtenir une liste d'URLs de documents à télécharger. Ces mots clé sont formés à partir des labels de concepts et éventuellement des termes déjà alignés aux concepts de l'ontologie. Le chargement automatique des documents à partir des sites ainsi identifiés se fait via un *crawler* (un robot d'indexation) comme montré dans la figure 2.7.

Pré-traitement du corpus

Les documents chargés sont traités pour obtenir un ensemble de documents XHTML bien formés grâce à des outils comme JTidy, TidyHtml et des APIs comme SAX (Simple API for XML) et DOM (Document Object Model). Chaque document est ensuite représenté par un ensemble de mots numérotés par ordre d'occurrence dans le document. Les caractères de ponctuation sont bien entendu conservés pour la tâche d'extraction, et les balises remplacées par des points.

2.3.2 Processus d'extraction *SHIRI-Extract*

SHIRI-Extract s'applique à un ensemble de documents après pré-traitement. Il s'appuie sur des patterns indépendants du domaine en distinguant :

- les patterns syntaxiques de type syntagmes nominaux couramment utilisés dans l'extraction d'information pour extraire les termes candidats,

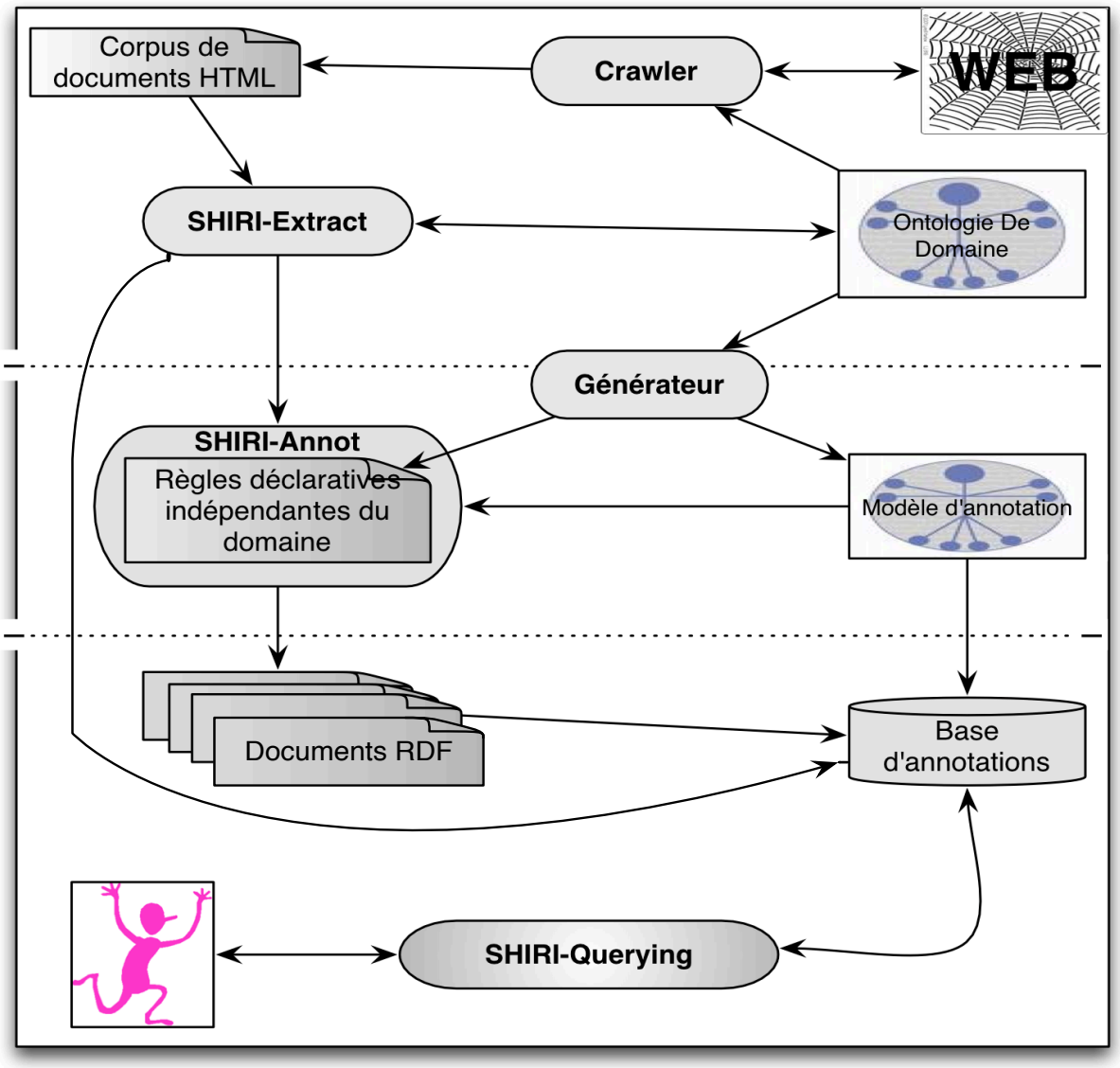


FIGURE 2.7 – Principaux modules de notre système

- et lexico-syntaxiques utilisés dans les techniques de détection d'entités nommées pour extraire les entités nommées candidates.

Le module *SHIRI-Extract* utilise un algorithme appelé *Extract-Align* qui enrichit lexicalement l'ontologie de manière incrémentale par chaque terme ou EN candidat aligné (associé un concept défini dans l'ontologie). L'alignement utilise soit directement l'ontologie lexicale augmentée de nouveaux termes alignés soit en passant par le Web pour découvrir de nouveaux termes. La sortie de ce module est un ensemble de triplets RDF :

1. les triplets qui enrichissent l'ontologie : ($term$, $hasTerm$, $uri(Concept)$) et ($termNe$, $hasTermNe$, $uri(Concept)$)

2. les triplets de pré-annotation qui précisent quels concepts de termes ou ENs ont été repérés et dans quels nœuds des documents. ($uri(Node)$, $containInstanceOf$, $uri(Concept)$) et ($uri(Node)$, $hasValueInstance$, $term$ ou $termNe$)

2.3.3 Processus d'annotation *SHIRI-Annot*

Le module *SHIRI-Annot* a pour objectif d'annoter les documents en utilisant les triplets des relations $containInstanceOf$ et $hasValueInstance$ résultant de *SHIRI-Extract*. Pour ce faire nous avons défini un ensemble de règles en logique du premier ordre permettant de générer les annotations conformes au modèle d'annotation :

1. annoter les nœuds en tenant compte à la fois des concepts associés à ces nœuds via la propriété $containInstanceOf$ dans le module d'extraction et de leur agrégation structurelle dans les documents. Les triplets que nous obtenons en sortie sont : ($uri(Node)$, $type$, $uri(c)$), ($uri(Node)$, $type$, $uri(PartOfSpeech)$), ($uri(NodePos)$, $isIndexedBy$, $uri(c)$), ($uri(Node)$, $type$, $uri(SetOfc)$)

2. inférer des relations entre nœuds en tenant compte à la fois des types de termes et ENs localisés dans ces nœuds et de leur voisinage structurel. Ce voisinage structurel est représenté par un chemin de longueur fixée entre nœuds dans l'arbre structurel DOM du document. Les triplets que nous obtenons en sortie sont : ($uri(Node)$, $neighborOf$, $uri(Node)$)

SHIRI-Querying est le module qui traite la requête de l'utilisateur exprimée avec les concepts et les relations de l'ontologie. L'objectif est de reformuler automatiquement cette requête initiale en exploitant le modèle d'annotation afin d'atteindre d'autres nœuds réponses et de trier ces nœuds en fonction de leur sémantique.

Dans notre travail, nous avons défini un ensemble de fonctions de transformations qui permettent de construire des requêtes reformulées. Ce module est détaillé dans [MBPT10a].

```
<h2>
31 May - 4 June 2009, Heraklion, Greece
</h2>
<p>
The 6th Annual European Semantic Web
Conference (ESWC2009) will present the latest
results in research and applications of Semantic
Web technologies. ESWC2009 will also feature
a tutorial program, system descriptions and demos,
a poster track, a PhD Symposium and a
 number of collocated workshops.
</p>

Document 1
```

```
<h1>
12th International Conference on Data Warehousing and Knowledge Discovery
</h1>
<h3>
Bilbao, Spain<br><br /><br>August 30 - September 2, 2010<br />
</h3>
<ul>
<li> Data mining techniques: clustering, classification, association rules,
decision trees, etc.</li>
<li>Semantic web intelligence</li>
<li> Analytics for social networks</li>
</ul>
<h1>Programm Committee</h1>
Alberto Abello, Universitat Politecnica de Catalunya<br/>
Ira Assent, Aalborg University<br/>
Elena Baralis, Politecnico di Torino<br/>

Document 2
```

FIGURE 2.8 – Exemples de documents à traiter

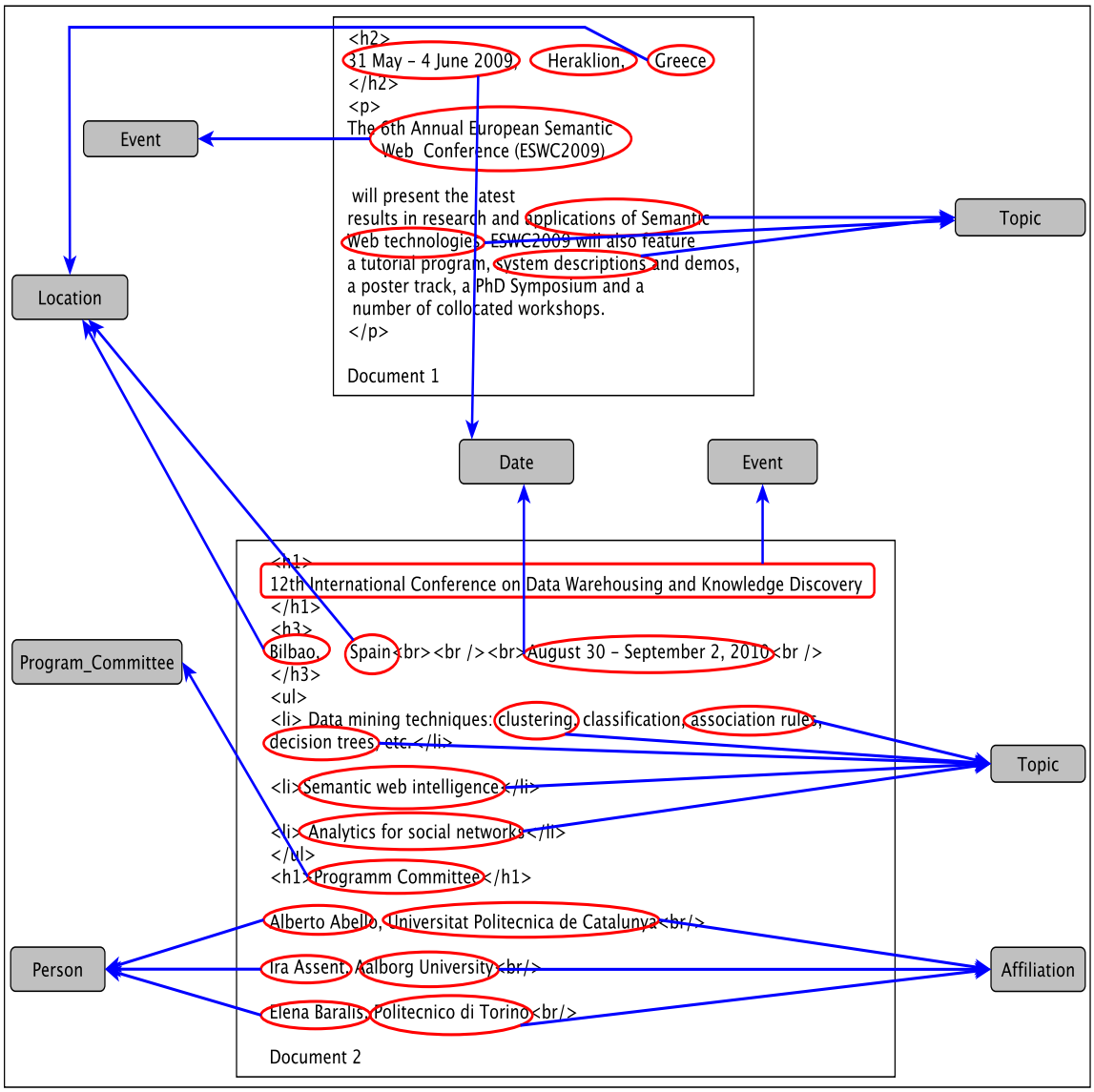


FIGURE 2.9 – Extraction et alignement des termes et entités nommées

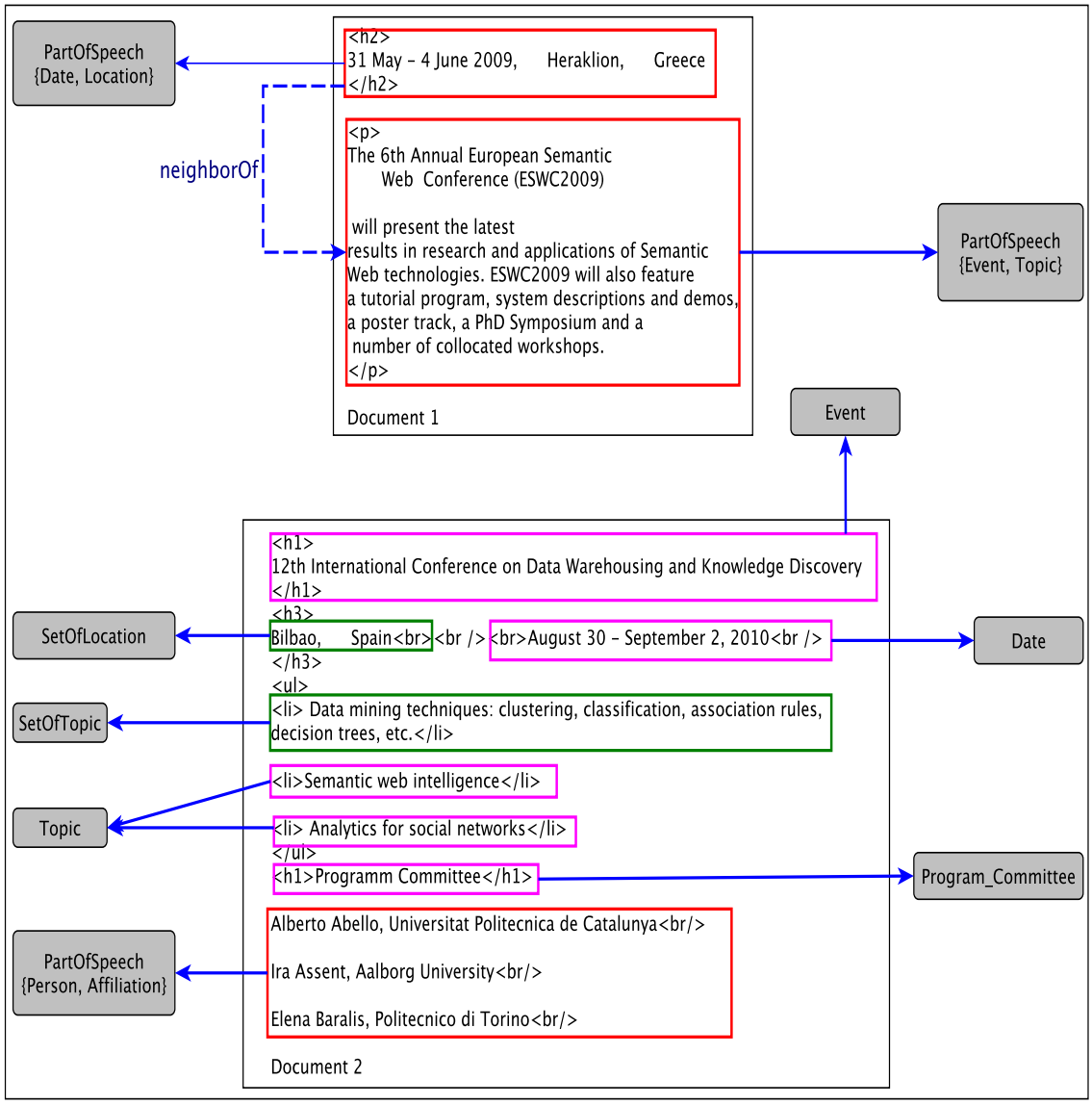


FIGURE 2.10 – Annotation des nœuds de documents

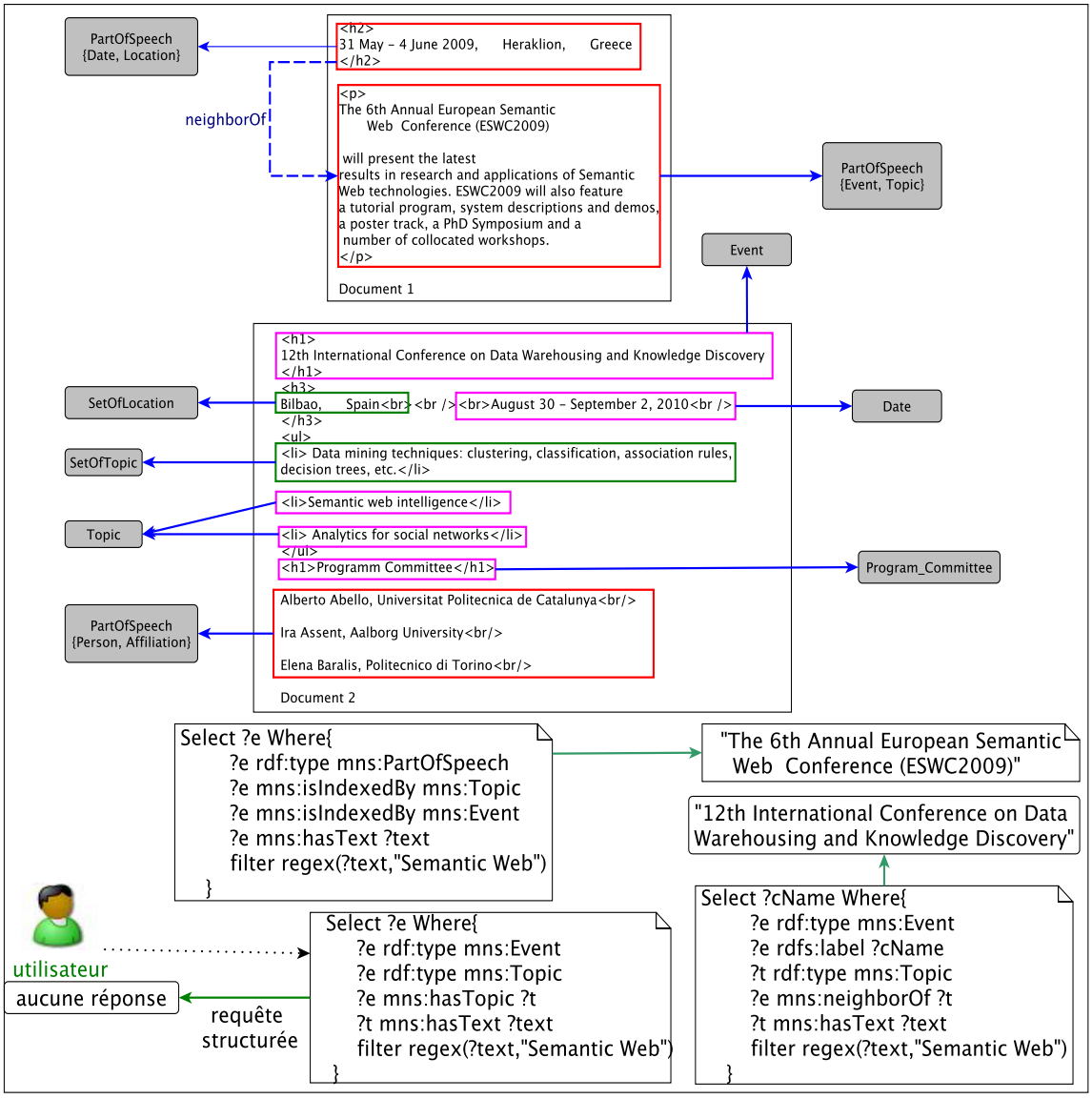


FIGURE 2.11 – Interrogation des annotations

2.4 Scénario d'usage

La figure 2.8 montre 2 exemples de documents semi-structurés (Document 1 et Document 2) traités successivement par *SHIRI-Extract* et *SHIRI-Annot*.

SHIRI-Extract extrait les termes et les entités nommées et les aligne avec les concepts de l'ontologie. La figure 2.9 montre les résulats de ce module. Par exemple, dans le document 1, on a pu extraire et aligner les entités nommées *Greece* et *Heraklion* au concept *Location* et *applications of Semantic Web technologies* au concept *Topic*. Dans le document 2, le terme *Semantic web intelligence* extrait est aligné au concept *Topic* et le terme *Politecnico di Torino* est aligné au concept *Affiliation*.

Les triplets générés à ce stade pré-annotent les documents en instanciant la relation $containInstanceOf$ entre les nœuds et les concepts des termes et ENs localisés dans ces nœuds et la relation $hasValueInstance$ entre ces mêmes nœuds et les littéraux des termes et entités nommées.

La figure 2.10 montre les résultats de *SHIRI-Annot* qui annote les nœuds en utilisant les pré-annotations et l'agrégation des instances et leur voisinage structurel. Par exemple, dans le document 2, la métadonnée $SetOfTopic$ est utilisée pour annoter le nœud qui contient plusieurs termes alignés au concept $Topic$.

Dans le document 1, les nœuds contenant plusieurs termes ou ENs alignés avec différents concepts comme $Person$ et $Event$ ou *date* et $Location$ sont annotés comme des instances de *PartOfSpeech*.

La relation $neighborOf$ est instanciée entre nœuds voisins (distance de 2 dans la figure 2.10) contenant des termes et ENs de concepts domaine et co-domaine d'une relation définie dans l'ontologie. Par exemple, dans le document 1, $neighborOf$ est instanciée entre deux nœuds $PartOfSpeech$ voisins. Le premier contient des termes et ENs alignés avec les concepts $Event$ et $Topic$. Le second contient des ENs alignés avec les concepts $Location$ et $Date$.

Enfin, la figure 2.11 montre une requête utilisateur exprimée par les concepts et les relations de l'ontologie. Cette requête cherche des conférences qui ont pour thème *Semantic Web*. La requête telle qu'elle est posée n'a aucune réponse. Les différentes reformulations permettent d'atteindre les nœuds agrégés $PartOfSpeech$ et $SetOfConcept$ et les nœuds reliés par $neighborOf$.

Conclusion

Dans ce chapitre, nous avons présenté notre architecture modulaire qui permet de traiter séparément l'extraction et l'annotation. *SHIRI-Extract* extrait des termes et ENs candidats de manière indépendante du domaine et les aligne ensuite avec l'ontologie.

SHIRI-Annot exploite la structure et la sémantique associée aux nœuds pour annoter les documents. *SHIRI-Extract* est détaillé dans le chapitre 3. *SHIRI-Annot* est détaillé dans le chapitre 4.

Chapitre 3

EXTRACTION, ALIGNEMENT DES ENTITÉS NOMMÉES ET DES TERMES

Sommaire

Introduction

L'automatisation de l'annotation de documents hétérogènes peut exploiter une technique de reconnaissance d'entités nommées qui a pour but de localiser et de classer des éléments du texte dans des catégories pré-définies telles que les noms de personnes, d'organisations, de lieux ou les dates.

Elle peut également se baser sur la recherche de termes associés à des concepts du domaine et qui ne sont pas des entités nommées.

Nous proposons une approche appelée *SHIRI-Extract* qui permet d'extraire des entités nommées et des termes dans les documents et de les rapprocher à des concepts d'une ontologie de domaine. Cette approche utilise des patterns lexico-syntaxiques indépendants du domaine. Elle vise à s'affranchir de l'existence de ressources lexicales en utilisant le Web : l'alignement des termes candidats se fait soit localement avec l'ontologie, soit en utilisant des termes provenant du Web. L'ontologie est ensuite enrichie par les termes extraits découverts dans les documents ou sur le Web. Ainsi le besoin d'accès au Web est réduit au fur et à mesure que les documents, appartenant au même domaine, sont traités.

Dans ce chapitre, nous présentons brièvement *SHIRI-Extract*, ensuite nous présentons l'algorithme *Extract-Align* qui extrait des entités nommées et des termes et qui les aligne avec des concepts de l'ontologie.

3.1 Présentation de SHIRI-Extract

Dans cette section, nous présentons tout d'abord les principales étapes de l'extraction et de l'alignement. Nous détaillons ensuite comment l'extraction des termes et des entités nommées est effectuée, puis comment ces termes candidats sont alignés avec l'ontologie. Enfin, nous montrons comment l'ontologie est enrichie.

3.1.1 Objectifs généraux

SHIRI-Extract correspond à la phase de localisation et d'alignement de *SHIRI*. L'objectif de *SHIRI-Extract* est de générer des triplets RDF qui associent aux nœuds de documents les concepts du domaine décrits par les termes ou les entités nommées qui ont été localisés dans ces nœuds. Les triplets générés utilisent les attributs *hasValueInstance* et *containInstanceOf* définis dans le modèle d'annotation.
Par exemple, la figure 3.2 montre les triplets qui permettent d'associer le terme *Porto* au nœud *[http ://www..../cfp.html/ body/.../p]* qui le contient :

(http ://www..../cfp.html/body/.../p, hasValueInstance, "Porto") ;
(http ://www..../cfp.html/body/.../p, containInstanceOf, uri(Location)).

Les principales étapes de *SHIRI-Extract* sont les suivantes (cf. figure 3.1) :

- Les documents XHTML relatifs au domaine sont nettoyés et passés à l'outil d'extraction.

- L'approche d'extraction commence par étiqueter morpho-syntaxiquement les documents avant de leur appliquer un ensemble de patterns pour obtenir les termes ou les EN candidats. Nous appelons *terme candidat* (ou *EN candidate*) un terme (ou une entité nommée) extrait et non encore aligné avec un concept de l'ontologie. Nous distinguons deux types de patterns :

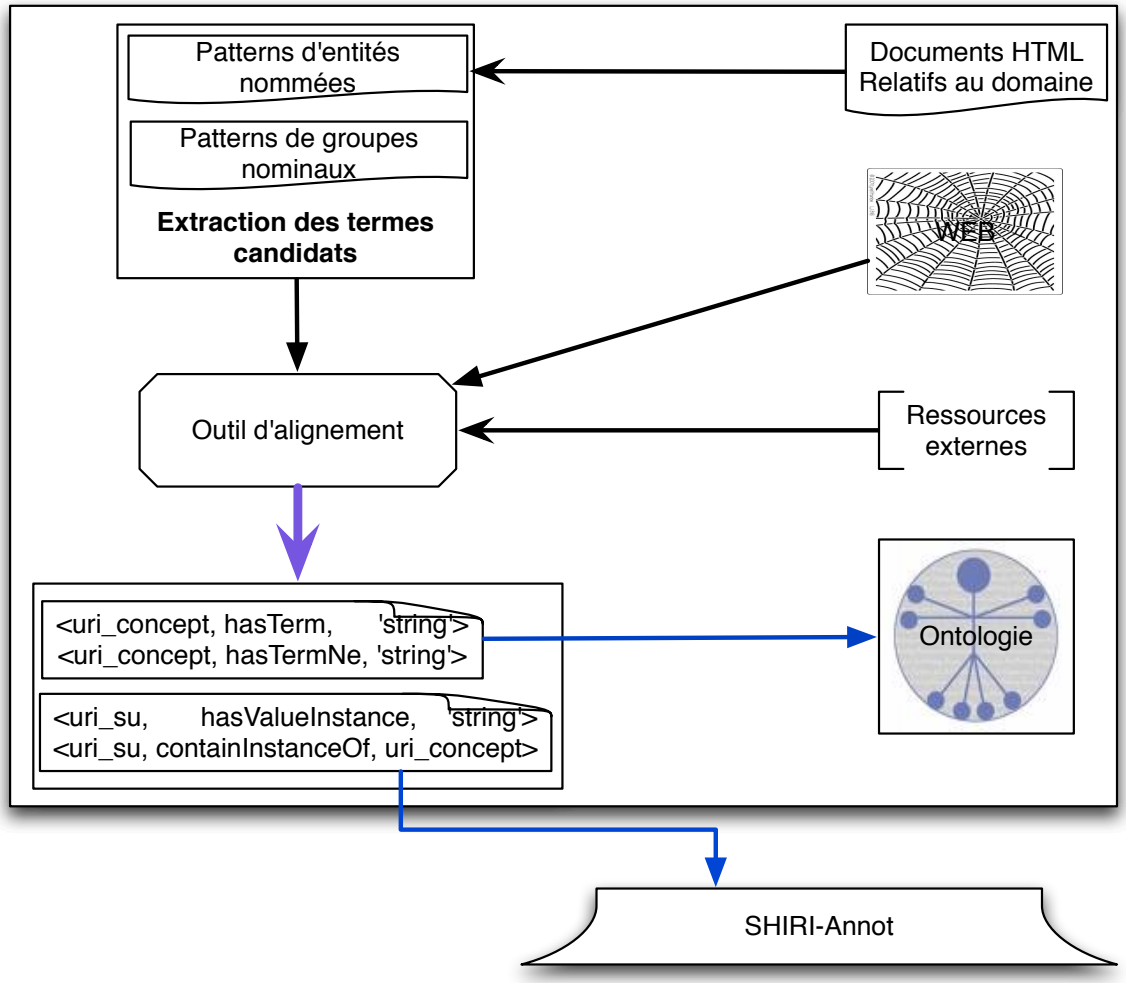


FIGURE 3.1 – Architecture de SHIRI-Extract

1. des patterns spécifiques à la recherche d'entités nommées ;

2. des patterns permettant de découvrir des termes.

Les deux types d'extraction sont réalisés indépendamment l'un de l'autre sur l'ensemble des documents.

- Les termes ou EN candidats obtenus sont alignés avec les concepts de l'ontologie du domaine. Cet alignement est particulier à chaque type de pattern. Il est effectué soit directement avec les termes, les entités nommées ou les labels des concepts de l'ontologie, soit indirectement en utilisant une ressource externe ou via le Web. Une ressource externe peut être terminologique comme WordNet ou une base de connaissances d'entités nommées comme celle utilisée dans KIM [PKK+04] ou un outil d'extraction d'EN appartenant à des catégories que l'on peut directement associer à des concepts de l'ontologie de domaine comme [P.07].

- L'ontologie est régulièrement enrichie avec les termes candidats alignés qui peuvent ainsi être exploités dans des traitements ultérieurs. L'approche génère un ensemble de faits construits à partir des propriétés *hasTermNe* et *hasTerm* définies dans l'ontologie. Par exemple, si "*Porto*" n'existe pas dans les entités nommées de l'ontologie et que l'on a découvert qu'il s'agit d'une instance du concept *Location* (Lieu), le fait *(uri(Location), hasTermNe, "Porto")* est généré.

Nous avons défini un algorithme de localisation et d'alignement appelé *Extract-Align*. Il est décrit en détail dans la section 3.2.

3.1.2 Extraction des termes candidats

Dans notre approche, pour extraire les termes candidats dans les documents, nous avons appliqué une méthode basée sur l'utilisation d'un ensemble de patterns syntaxiques ou lexico-syntaxiques. Dans cette section, nous présentons les patterns utilisés pour les termes et pour les entités nommées.

Extraction des termes

Nous avons choisi d'utiliser des patterns syntaxiques indépendants du domaine qui permettent de localiser certains groupes nominaux susceptibles de représenter des termes.

Comme nous souhaitons distinguer les entités nommées des termes, nous avons supprimés les noms propres des seize patterns définis dans [Arp95] .

1. (NN|NNS) & (IN|CC) & (NN|NNS) & (IN|CC) & (NN|NNS) & (IN|CC) & (NN|NNS)
2. (NN|NNS) & (IN|CC) & (NN|NNS) & (IN|CC) & (NN|NNS)
3. VBG & (NN|NNS) & (IN|CC) & (NN|NNS) & (NN|NNS)

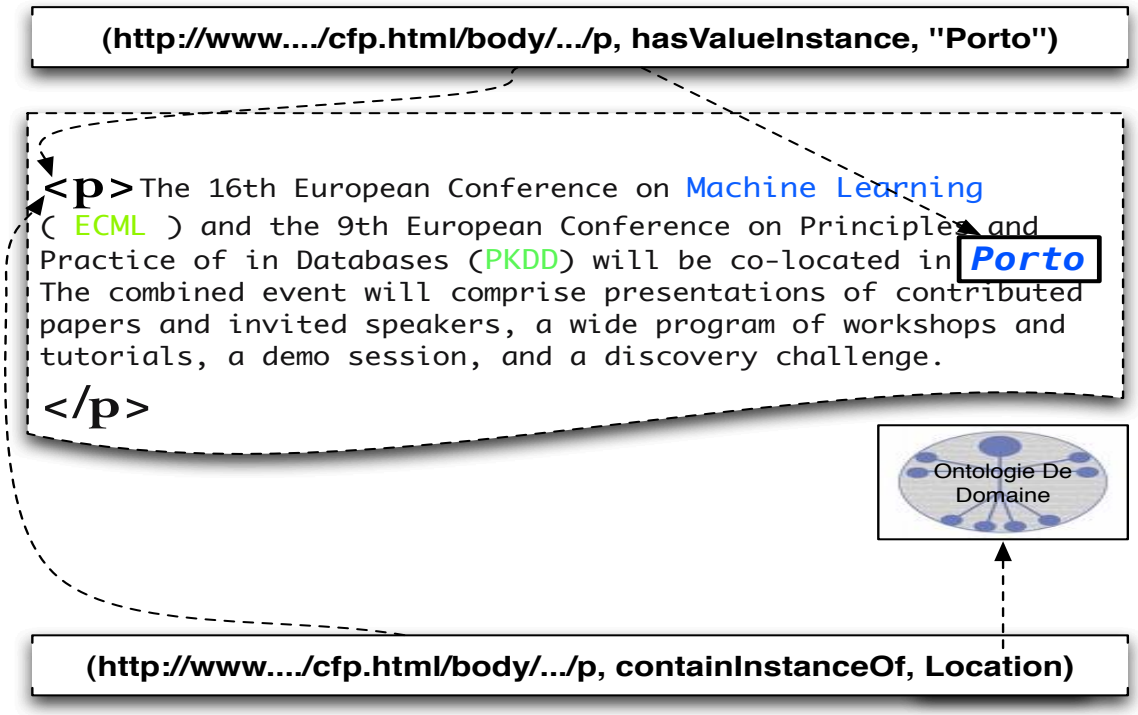


FIGURE 3.2 – Exemple d'instanciation des relations *hasValueInstance* et *containInstanceOf*

Symbole/Tag	Signification	Exemples
?	0 ou 1 fois	
+	Répétition 1 ou N fois	
*	Répétition 0 ou N fois	
&	Concaténation	A & B : A suivi de B
|	Ou/Choix	A | B : A ou B
NN	Nom commun	conference
NP	Nom propre	Moussa
CC	Conjonction de coordination	and, but, or, nor, yet
IN	Conjonction de subordination	without
JJ	Adjectif	great
FW	Mot étranger	bête, noire, persona, non, grata
DT	Déterminant	the, a(n), no, every
UH	Interjection	OH
POS	Forme possessive	's, '
NNS	Nom commun au pluriel	conferences
NPS	Nom propre au pluriel	Jacques
VBG	Verbe au gérondif / participe présent	saving, presenting
VBN	Verbe au participe passé	presented
\w	tout caractère	
[a-z]	tout caractère minuscule	m, k, j
[A-Z]	tout caractère majuscule	M, K, J

TABLE 3.1 – Signification des tags

4. (JJ|VBN) & (JJ|VBN) & (NN|NNS) & (IN|CC) & (NN|NNS)
5. (NN|NNS) & (IN|CC) & (JJ|VBN) & (NN|NNS)
6. (JJ|VBN) & (NN|NNS) & (IN|CC) & (NN|NNS)
7. (NN|NNS) & (IN|CC) & (NN|NNS) & (NN|NNS)
8. (NN|NNS) & (IN|CC) & (NN|NNS)
9. (JJ|VBN) & (NN|NNS) & (NN|NNS)
10. (JJ|VBN) & (JJ|VBN) & (NN|NNS)
11. (JJ|VBN) & VBG & (NN|NNS)
12. (NN|NNS) & (NN|NNS) & (NN|NNS)
13. (JJ|VBN) & (NN|NNS)
14. (NN|NNS) & (NN|NNS)
15. VBG & (NN|NNS)
16. NN|NNS

Ces patterns permettent d'obtenir des termes tels que : *workshop*, *database indexing and search*, *Semantic Web*, *combined events*, *Principles and Practice of Knowledge Discovery*.

Ces patterns ont en général un bon rappel puisqu'ils ne sont pas spécifiques à un domaine, ni contraints syntaxiquement. Par exemple tous les noms communs sont extraits et nombre d'entre eux ne sont pas des termes du domaine. Une manière de filtrer ces termes aurait été de leur appliquer des techniques statistiques permettant de retenir une majorité de termes pertinents. Dans notre approche nous avons défini une méthode d'alignement qui permet d'associer un terme à un concept du domaine et qui sert également de filtre au grand nombre de termes candidats extraits.

Extraction des Entités Nommées

Pour extraire les entités nommées, nous exploitons les patterns lexico-syntaxiques utilisés par C-Pankow [CLS05]. *pn* signifie *"pattern de nom"* et *pnp* signifie *"pattern de nom propre"*

- pn = DT ? & JJ ? & (NN(S) ?)+
- pn^2 = pn & CC (and|or) & pn
- pn^+ = pn & (, pn)+ & CC (and|or) & pn
- $pn^{(+)}$ = pn & (, pn)+
- pnp = DT ? & (JJ [A-Z]+) ? & (((NPS|NNS|NN|NP|JJ|UH|FW) [A-Z] & \w*)+ & (DT the|IN of|POS '|-|FW de|FW la|FW los|FW las|FW del|NP de|NP del|NP la| & NP los|NP las|NN [A-Z]+|NNS [A-Z]+|NP [A-Z]+|NPS [A-Z]+)+ & (((NPS|NNS| NN|NP|JJ|UH|FW) [A-Z] & \w*)+)) ?))

Ces patterns permettent d'obtenir des termes tels que : *Porto*, *SEMMA*, *European Conference on Machine Learning*, *ECML*, *Tim Berners-Lee*.
Nous utilisons également des expressions régulières classiques pour la détection de certaines instances de concepts de l'ontologie telles que le concept *Date*.

3.1.3 Stratégie de sélection des termes candidats

Un terme candidat extrait peut être inclus dans un autre terme candidat extrait par un autre pattern pour une même occurrence. Par exemple, dans le tableau 3.2 qui montre les résultats de l'extraction de termes et leur numéro d'occurrence dans les documents, le terme candidat $distributed_{75}$ $databases_{76}$ contient le terme $databases_{76}$.

Nous avons choisi de traiter ces termes candidats du plus long au plus court, la taille d'un terme candidat étant le nombre de mots le composant. En effet, le terme le plus long est souvent plus précis ou plus significatif que les sous-termes qu'il contient. Par exemple, l'entité nommée *"Gaston Berger University"* qui est un établissement est plus significatif que "*Gaston Berger*" qui est une personne (philosophe et industriel né à Saint-Louis, Sénégal). Ce raisonnement est également valable pour les termes. Par exemple, le terme "*knowledge management systems*" obtenu avec le pattern (NN|NNS) & (NN|NNS) & (NN|NNS) est plus précis que "*systems*" obtenu avec (NN|NNS).

D'un autre côté, un terme peut être trop complexe pour être aligné avec un concept de l'ontologie. Aussi, si nous n'avons pas réussi à aligner un terme, nous tentons d'aligner ses sous-termes. C'est le cas, par exemple, du terme candidat "*Interoperability of data on the Semantic Web*" pour lequel l'alignement avec le concept *Topic* échouerait très probablement à cause de sa complexité. En particulier, l'alignement exploitant le Web comme ressource externe, détaillé dans la section suivante, risque d'échouer compte tenu de la faible fréquence de ce terme sur le Web.

Afin de garder l'information sur l'inclusion des termes, les mots qui forment les termes sont numérotés selon leur ordre d'apparition dans les documents. Cette numérotation permet de distinguer les différentes occurrences du même terme et leurs inclusions. Par exemple, si le terme "$distributed_{75}$ $databases_{76}$" a pu être aligné il sera inutile de traiter "$databases_{76}$". En revanche, si le terme "$artificial_{78}$ $intelligence_{79}$" n'a pas pu être aligné avec le concept "*Topic*", nous tentons d'aligner le terme "$Intelligence_{79}$". Le terme candidat "$databases_{86}$" qui est une autre occurrence dans le document est aussi traité.

En revanche, si nous n'avons pas réussi à aligner une entité nommée candidate, nous ne tentons pas d'aligner les entités nommées qui sont contenues dans celle-ci.

3.1.4 Alignement local ou via le web

Dans notre approche, l'alignement des termes candidats se fait soit localement avec l'ontologie, soit en utilisant une ressource lexicale externe, soit via le Web. Nous avons distingué le cas des entités nommées et le cas des termes. En effet, les éléments de l'ontologie utilisés pour l'alignement sont différents et les techniques d'alignement sont plus strictes pour les entités nommées.

Texte Original	Termes Candidats Extraits
... $Areas_{71}$ of_{72} $interest_{73}$ are_{74} $distributed_{75}$ $databases_{76}$ and_{77} $artificial_{78}$ $intelligence_{79}$. The_{80} $workshop_{81}$ $SEMMA_{82}$ $focuses_{83}$ $also_{84}$ on_{85} $databases_{86}$. $Intelligence_{87}$ $areas_{88}$ ...	... [$Areas_{71}$] of_{72} $interest_{73}$ are_{74} [$distributed_{75}$ [$databases_{76}$]] and_{77} [$artificial_{78}$ [$intelligence_{79}$]]. The_{80} [$workshop_{81}$] $SEMMA_{82}$ $focuses_{83}$ $also_{84}$ on_{85} [$databases_{86}$]. [$Intelligence_{87}$] [$areas_{88}$] ...

TABLE 3.2 – Exemples de termes candidats extraits

Alignement des entités nommées

Une entité nommée extraite est comparée aux entités nommées appartenant à l'ontologie. Il y a peu de variations syntaxiques dans les noms d'entités nommées (à l'exception des abbréviations ou des erreurs typographiques). Nous avons choisi de considérer des mesures de similarités strictes pour décider que deux entités nommées doivent être alignées (e.g. chaînes de caractères identiques, techniques probabilistes avec des seuils élevés dans l'outil externe [P.07]).

Alignement local : tout d'abord, nous tentons d'aligner l'entité nommée localement en utilisant les entités nommées présentes dans les valeurs de la propriété $hasTermNe$ de l'ontologie. Si cette tentative échoue, cet alignement peut se faire via une ressource d'entités nommées externe.

Alignement via le Web : au cas où cette dernière tentative ne permet pas de rapprocher l'entité nommée d'un concept de l'ontologie, nous recherchons des termes sur le Web qui sont susceptibles de décrire le concept auquel cette entité nommée appartient et nous tentons d'aligner ces termes avec les concepts de l'ontologie. Plus précisément, cet alignement via le web est réalisé selon les principes suivants :

Requêtes traitées par le moteur de recherche : les requêtes contenant l'entité nommée sont construites et soumises à un moteur de recherche en utilisant, comme dans [CLS05], les patterns de Hearst [Hea92] suivants :

$hearst_1$ = np1 ,? & DT such & (IN | RB | CS) as & (pnp3|pnp2|pnp1)

$hearst_2$ = pnp4 & CC (or | and) & JJ other & np1

$hearst_3$ = np1 & ,? & RB (especially | including) & (pnp3|pnp2|pnp1)

$hearst_4$ = DT such & np1 & IN as & (pnp3|pnp2 |pnp1)

$apposition$ = pnp1 & , & np1

$definite_1$ = ((DT the | DT The) & (JJ & \w+)? & ((NN(S)? [a-z]+)+) & pnp1)

$definite_2$ = ((DT the | DT The) & (JJ & \w+)? & pnp1 & ((NN(S)? [A-Z]+)+))

$copula$ = pnp1 & (BEZ | BEDZ) \w+ & np1 ;

Les requêtes sont soumises au Web grâce à un plug-in adapté au moteur de recherche. Par exemple, la requête correspondant au pattern $hearst_1$ appliquée à l'entité nommée *Porto* est *"such as Porto"*. Le résultat de la requête fournit un ensemble d'extraits de documents contenant le pattern instancié. Un des extraits fournis par la requête *"such as Porto"* est :

Spel today manages a total of 16953 places in 24 car parking facilities in ***cities such as Porto****, Lisbon, Viseu and Matosinhos.*

Filtrage des extraits obtenus en réponses : les extraits sont filtrés suivant leur similarité avec le document d'où provient l'entité nommée. Les n meilleurs extraits sont retenus. La similarité est évaluée en utilisant la mesure du cosinus qui permet de comparer deux documents texte en se basant sur la fréquence des mots. Les mots vides sont supprimés avant cette comparaison grâce à une liste de mots vides.

$$cosinus(\overrightarrow{extrait}, \overrightarrow{doc}) = \frac{\overrightarrow{extrait}.\overrightarrow{doc}}{||\overrightarrow{extrait}||.||\overrightarrow{doc}||} \tag{3.1}$$

Sélection des termes trouvés sur le web : les extraits sont étiquetés morpho-syntaxiquement. Ensuite, les termes candidats sont repérés dans les patterns instanciés (*np1*) et lemmatisés. Ainsi, le terme candidat *City* est repéré dans l'exemple ci-dessus. Nous obtenons alors une liste de termes candidats pour l'entité nommée. Les termes dont la fréquence est supérieure à un seuil sont sélectionnés.

Alignement des termes trouvés sur le web : les termes sélectionnés sont utilisés pour aligner l'entité nommée en utilisant les valeurs des propriétés $hasLabel$ ou $hasTerm$ de l'ontologie. Cet alignement est réalisé suivant le principe décrit ci-dessous pour l'alignement local des termes. Par exemple, l'entité nommée *Porto* pour laquelle le Web a proposé le terme *City* est alignée avec le concept *Town* grâce à son label alternatif *City*. De même, *SEMMA* pour laquelle le Web a proposé le terme *international workshop* est alignée avec le concept *Event* grâce au terme *workshop* de ce concept.

Alignement des termes

Pour associer un terme à un concept de l'ontologie, nous utilisons l'outil d'alignement d'ontologie Taxomap [HZSR08].

Présentation de Taxomap Taxomap est un outil d'alignement de taxonomies (ensembles de concepts organisés par une relation de spécialisation). Cet outil est particulièrement adapté à l'alignement de taxonomies dissymétriques où l'une des taxonomies est peu structurée et même éventuellement limitée à une liste de concepts. L'alignement est en effet orienté de l'ontologie peu structurée, dite ontologie source, à l'ontologie bien

structurée, dite cible. Il exploite les ensembles de labels des concepts des deux taxonomies ainsi que la structure de la taxonomie cible. Cet outil d'alignement peut donc facilement être utilisé pour comparer un terme aux termes présents dans les labels de concepts d'une taxonomie en considérant qu'il s'agit d'un concept décrit par un unique label.

Taxomap recherche différents types de relations de correspondance entre concepts : des relations d'équivalence ($isEq$), de spécialisation (isA) ou de proximité sémantique ($isClose$) dont la sémantique n'est pas précisemment définie.

TreeTagger est utilisé pour rechercher la catégorie morpho-syntaxique de chacun des mots des labels de concept et pour les lemmatiser (i.e donner sa forme canonique). Les mots sont soient considérés comme des mots pleins (les noms qui n'apparaissent pas après une préposition), soient comme des modificateurs. Les mots pleins ont alors plus de poids dans les calculs de similarité entre labels. Cette similarité est calculée en utilisant une variante de la mesure de Lin [Lin98], qui peut être utilisée pour mesurer la similaritée entre deux concepts, x et y, en se basant sur le nombre de tri-grammes communs aux labels de ces deux concepts. La variante $\text{Sim}_{LinLike}$ utilisée dans Taxomap permet de donner plus de poids aux mots pleins. Elle est définie de la façon suivante :

$$Sim_{LinLike}(x,y) = \frac{2 \ * \ \sum_{t \in Inter} logP(t) \ + \ 0.5 \ * \ \sum_{t \in I'} logP(t)}{\sum_{t \in tri(x)} logP(t) \ + \ \sum_{t \in tri(y)} logP(t)} \tag{3.2}$$

où tri(x) est l'ensemble des tri-grammes de la chaîne de caractères x, P(t) représente la probabilité d'apparition du tri-gramme t dans les labels d'une des ontologies (estimée en utilisant la fréquence du tri-gramme dans l'ontologie, Inter est l'ensemble des tri-grammes communs extraits à partir des mots pleins, I' est l'ensemble des tri-grammes communs extraits à partir des modifieurs.

Différentes techniques d'alignement (T1, T2, T3, T4, T5 , T6, T7 et T8) sont appliqués de manière séquentielle, de la plus sûre, à la plus incertaine. Soit CS un concept de l'ontologie source et $CCmax$, $CC2$, $CC3$ les trois concepts de l'ontologie cible les plus similaires à CS, les techniques proposées sont les suivantes :

- T1 (Equivalence) : une relation d'équivalence $isEq$ est générée si la similarité d'un des labels du concept CS de l'ontologie source avec un des labels du concept $CCmax$ de l'ontologie cible est supérieure ou égale à un seuil.

- T2 et T3 (Inclusions) : une relation CS isA $CCmax$ est générée si un des labels de $CCmax$ est inclus dans un des labels de CS et si tous les mots inclus sont des mots pleins (technique T2). Une relation CS $isClose$ $CCmax$ est générée si un des labels de CS est inclus dans un des labels de $CCmax$.

- T4, T5 et T6 (Similarité relative) La similarité relative est le rapport entre la similarité de $CC2$ et celle de $CCmax$. Si elle est inférieure ou égale à un seuil, T4, T5 ou T6 peuvent s'appliquer. Une relation CS $isClose$ $CCmax$ est générée si la similarité de $CCmax$ est supérieure ou égale à un seuil et qu'il existe une inclusion de

labels entre CS et $CCmax$(technique T4). Par exemple, si CS a pour label *Chaîne de Montagne*, et $CCmax$ a pour label *Montagne*, CS *isClose* $CCmax$.
Une relation CS *isClose* $CCmax$ est générée si la similarité de $CCmax$ est supérieure ou égale à un certain seuil supérieur au précédent (technique T5). Une relation CS *isA* père($CCmax$) est générée si la valeur de similarité avec $CCmax$ est inférieure à un seuil mais supérieure ou égale à un deuxième seuil (technique T6).

- T7 et T8 (Raisonnement sur la structure) Une relation CS *isClose* CC est générée si la relation de spécialisation CS *isA* X existe dans l'ontologie source et si la relation de correspondance X *isA* CC a été générée (technique T7) . Une relation CS *isClose* CC est générée si CC est le concept de l'ontologie cible qui a le plus de fils qui ont des labels identiques aux labels des fils de CS. La relation CS *isA* P est générée si les trois concepts de l'ontologie cible qui sont les plus similaires à CS ont une similarité supérieure à un seuil et ont un père commun P (technique T8).

Nous alignons le terme avec tous les concepts trouvés par Taxomap que la relation de correspondance soit l'équivalence ($isEq$), la subsomption (isA) ou la proximité sémantique ($isClose$), sous réserve que la valeur de similarité soit supérieure à un seuil donné.

Lorsque nous utilisons Taxomap, nous exploitons la composante lexicale de l'ontologie de domaine (ontologie cible) en considérant les termes présents dans les valeurs de la propriété $hasTerm$ comme des labels, en plus de deux déjà présents dans les valeurs de la propriété $hasLabel$.

Alignement local : comme pour les entités nommées, nous commençons par aligner le terme localement avec l'ontologie en utilisant l'outil Taxomap.

Alignement via le web : Si aucune correspondance n'est trouvée lors de l'alignement local par Taxomap, nous soumettons le terme au Web comme nous le faisons pour les entités nommées et avec les mêmes patterns. En effet, les patterns définis par Hearst permettent de retrouver aussi bien des liens *instance-de* que des relations d'hyponymie dans un texte.

Historisation des entités nommées et termes non alignés

Si une entité nommée ou un terme a été soumis au Web et n'a pu être aligné, nous ne souhaitons pas le soumettre de nouveau si ce terme est rencontré de nouveau dans un autre document. En effet, les appels Web sont coûteux et les documents étant relatifs au même domaine, nous supposons qu'il suffit de vérifier le contexte dans lequel apparaît le terme à sa première occurrence. Aussi, nous conservons les résultats obtenus lors de l'appel Web : l'entité nommée ou le terme accompagné de l'ensemble des termes candidats trouvés sur le web (éventuellement vide). Si nous avons pu trouver des termes candidats sur le Web, comme l'ontologie évolue, il est possible qu'une future tentative d'alignement réussisse en exploitant le fichier XML contenant ces informations historisées. Si nous n'avons pas pu en découvrir, il est inutile de soumettre à nouveau les requêtes sur le Web.

3.1.5 Enrichissement de l'ontologie

L'ontologie est enrichie par les termes et les entités nommées trouvés et alignés avec des concepts de l'ontologie. Cet enrichissement permet d'améliorer le traitement des documents de deux façons différentes. Tous les termes similaires au terme traité pourront être alignés avec ce terme et cet alignement sera local, nous n'avons plus besoin de soumettre le terme au Web. Il en est de même pour les entités nommées. Aussi, comme les documents sont du même domaine et que les termes se répètent, le nombre d'appels au web devrait diminuer au fur et à mesure que l'on traite des documents et que le nombre d'alignements augmente.

Les termes étant découverts automatiquement, ils peuvent être erronés : soit la suite de mots extraite n'a pas de sens suite à une erreur d'etiquetage (*results to scientists*) ou une erreur de typographie , soit il s'agit d'une EN ou d'un terme comportant un (des) mot(s) en plus (*year conference* au lieu de *conference*) ou une EN incomplète (*international conference on field*). D'autres types d'erreurs sont introduites par l'outil d'alignement (*Conferences themes* qui est annoté avec *Event*). Ceci explique que nous n'enrichissons pas les labels des concepts mais plutôt l'ensemble des termes associés aux concepts. Cet ensemble peut être nettoyé par un expert de temps en temps. Celui-ci peut décider de supprimer ou modifier un terme, ou d'en faire un nouveau label du concept. Il peut également décider de créer de nouveaux concepts, par exemple des concepts plus spécifiques.

3.2 Algorithme d'extraction et d'alignement

3.2.1 Notations et principes algorithmiques

L'algorithme *Extract-Align* extrait des termes et des entités nommées des documents HTML fournis en entrée et les aligne avec des concepts de l'ontologie.

Extract-Align est appliqué à un ensemble de documents HTML qui appartiennent au même domaine. Il prend en entrée l'ontologie $\mathcal{O}$ de ce domaine, un ensemble de patterns d'entités nommées $\mathcal{P}_{en}$, un ensemble de patterns de termes $\mathcal{P}_t$, un entier $\mathcal{T}aille Paquet$ fixant le nombre de candidats à traiter par le Web, un ensemble $\mathcal{E}checs$ contenant les termes ou EN candidats non alignés et déjà traités par le Web. Cet ensemble permet de ne pas soumettre au Web un terme ou EN candidat déjà traité et d'exploiter pour une autre occurrence de ce candidat le résultat obtenu lors de l'appel Web.

Entrées de la fonction *Extract-Align*

1. l'ensemble $\mathcal{C}orpus$ des documents HTML à annoter ;
2. l'ontologie $\mathcal{O}$ représentée en OWL ;
3. les patterns de termes $\mathcal{P}_t$;
4. les patterns d'entités nommées $\mathcal{P}_{en}$;
5. la taille des paquets de termes $\mathcal{T}aille Paquet$;
6. l'ensemble $\mathcal{E}checs$ des termes traités et non alignés.

Sorties de la fonction *Extract-Align*

1. l'ontologie $\mathcal{O}$ éventuellement enrichie ;
2. les pré-annotations des nœuds de documents représentées en RDF ;

Principales étapes de l'algorithme

- Nous commençons par un nettoyage et un étiquetage morpho-syntaxique des documents de $\mathcal{C}orpus$. La fonction $Extract$ extrait les entités nommées candidates en utilisant les patterns lexico-syntaxiques de $\mathcal{P}_{en}$ et les garde dans l'ensemble noté $\mathcal{I}_{en}$. Elle extrait aussi les termes candidats en utilisant l'ensemble des patterns P_t. Ces termes sont sauvegardés dans I_{te} (voir tableau 3.3).

- Chaque terme ou EN candidat est identifié par une séquence de mots numérotés en fonction de leur ordre d'apparition dans le document. Un candidat de longueur k se situant à la position i dans un document est une séquence de k mots : $t_i^k = w_i w_{i+1} ... w_{i+k-1}$, où w_{i+j} désigne le mot à la position $i+j$, j allant de 0 à $k-1$. Pour chaque candidat extrait n'appartenant pas à $Echec$, la fonction $AlignTerm$ ou $AlignNe$ est appelée (voir tableaux 3.4 et 3.5) selon qu'il s'agisse d'un candidat extrait en utilisant P_{en} ou P_t. Pour chaque candidat t_i^k, les candidats inclus $t_{i'}^{k'}$ sont traités du plus long au plus court tel que $k' = k, k-1, ..., 1$ et $i \leq i' \leq i+k-1$.

- Si l'appel à $AlignTerm$ ou $AlignNe$ réussit, I_{en} et I_t sont mises à jour en supprimant l'ensemble de ses sous-termes ou des ENs qu'il contient ($t_{i'}^{k'}$).

- Si l'appel à $AlignTerm$ ou $AlignNe$ échoue, le terme ou l'EN est ajouté à *TermesASoumettreAuWeb*. Lorsque la taille de l'ensemble *TermesASoumettreAuWeb* atteint *taillePaquet* la fonction $AlignWeb$ est appelée (voir tableau 3.6). Cette fonction utilise le Web pour rechercher des termes labels candidats pour les termes candidats appartenant à *TermesASoumettreAuWeb*. Un terme candidat sera aligné grâce à l'alignement en local de ses termes labels candidats. En cas d'échec, ce terme ainsi que ses termes labels candidats trouvés par le Web sont ajoutés dans l'ensemble $\mathcal{E}checs$. Cet ensemble sera exploité pour tenter d'aligner ce même terme pour une autre occurrence. Le nombre d'appels au Web devrait diminuer avec le traitement de l'ensemble des documents à annoter : (i) d'une part, les documents portent sur le même domaine, un même terme peut donc avoir au moins une occurrence dans le corpus, (ii) d'autre part, plus l'ontologie est enrichie plus la probabilité d'aligner en local un terme augmente.

L'algorithme de la la fonction *Extract-Align* est détaillé dans le tableau 3.3. Les fonctions d'alignement des entités nommées et des termes sont respectivement présentées dans les sections 3.2.2 et 3.2.3.

```
Extract-Align (Ensemble<Document> Corpus, Ontologie O, Ensemble<Pattern> Pt,
                              Ensemble<Pattern> Pen,
                              Entier TaillePaquet, Ensemble<Terme> Echecs)
  variables :    entier i, // le rang du terme courant
                 entier taille-courante, // la taille des termes de l'ensemble courant
                 Ensemble<Terme> Ite,  //ensemble des termes
                 Ensemble<Terme> Ien,  //ensemble des entités nommées
                 Ensemble<Terme> Ilen,  //les termes candidats de taille len
                 Ensemble<Terme> TermesASoumettreAuWeb  //les termes à soumettre au Web
  DEBUT
    Nettoyage(Corpus)
    EtiquetageMorphoSyntaxique(Corpus) //avec QTag
    Ien=Extract(Corpus, Pen)//extraction des entités nommées en utilisant les patterns dans Pen
    Ite=Extract(Corpus, Pt)//extraction des termes candidats en utilisant les patterns dans Pt
    taille-courante=tailleMaximale(Pen)
    i=0
    tant que (taille-courante >= 1) faire
      debut
        Ilen := {t dans Ien / taille(t)=taille-courante}
        pour chaque (t dans Ilen) faire
          debut
              Ilen = Ilen - t
              Si ( AlignNe(O, t, D, Echecs) == FAUX ) alors
                  debut
                      TermesASoumettreAuWeb.ajouter(t)
                      i= i+1
                  fin
              Sinon
                  debut
                      Misajour(Ien, t) //supprime les occurrences de termes
                      Misajour(Ite, t)  //incluses dans une occurrence de  t
                  fin

              Si ( i modulo TaillePaquet == 0 || Ilen est vide) alors
                    debut
                        WebAlign(O, TermesASoumettreAuWeb, Corpus, Echecs, Ien+Ite)
                        TermesASoumettreAuWeb = null // Réinitialisation des termes à soumettre au Web
                    fin
          fin
          taille-courante = taille-courante -1
      fin

    taille-courante=tailleMaximale(Pt)
    i=0
    tant que (taille-courante >= 1) faire
      debut
        Ilen := {t dans Ite / taille(t)=taille-courante}
        pour chaque (t dans Ilen) faire
          debut
              Ilen = Ilen - t
              Si (  AlignTerm(O, t, D, Echecs) == FAUX ) alors
                    debut
                       TermesASoumettreAuWeb.ajouter(t)
                       i= i+1
                    fin
              Sinon
                      Misajour(Ite, t) //supprime les occurrences de termes
                                              incluses  dans une occurrence du  terme aligné
              Si ( i modulo TaillePaquet == 0 || Ilen est vide) alors
                    debut
                        WebAlign(O, TermesASoumettreAuWeb, Corpus, Echecs, Ite)
                        TermesASoumettreAuWeb = null // Réinitialisation des termes à  soumettre au Web
                    fin
          fin
          taille-courante = taille-courante -1
      fin
  FIN
```

TABLE 3.3 – Algorithme d'extraction et d'alignement d'entités nommées et de termes

3.2.2 Alignement local d'une entité nommée

La fonction *AlignNe*, présentée dans le tableau 3.4 est utilisée par *Extract-Align* pour rapprocher les termes entités nommées candidats des entités nommées décrites pour chaque concept de l'ontologie. La fonction $AlignNe$ utilise des mesures de similarité appropriées à la comparaison d'entités nommées. L'égalité de chaîne de caractères est la solution la plus stricte.

Entrées de la fonction *AlignNe*

1. l'ontologie $\mathcal{O}$ représentée en OWL ;
2. l'entité nommée candidat t_i^k à aligner ;
3. le document $\mathcal{D}$ contenant t_i^k ;
4. l'ensemble des termes candidats traités et non alignés $\mathcal{E}checs$.

Sorties de la fonction *AlignNe*

1. l'ontologie $\mathcal{O}$ enrichie par des triplets $hasTermNe$;
2. les pré-annotations en RDF des nœuds de documents contenant t_i^k avec les relations $hasValueInstanceOf$ et $containInstanceOf$;
3. une valeur booléenne *vrai* si t_i^k est alignée et *faux* sinon.

Principales étapes de l'algorithme

La fonction *AlignNe* procède ainsi :

- Elle vérifie si l'entité nommée t_i^k est similaire à une entité nommée présente dans les valeurs de $hasTermNe$ de l'ontologie. Si tel est le cas, elle appelle la fonction $Misajour$, elle pré-annote le nœud contenant t_i^k par les concepts correspondants, et enrichit si nécessaire l'ontologie et retourne *vrai* ;

- Sinon elle cherche si t_i^k a été soumis au Web. Si $t_i^k \in \mathcal{E}checs$, elle récupère les termes candidats du Web de t_i^k à partir de l'ensemble $\mathcal{E}checs$.

- Si cet ensemble est non vide, elle tente d'aligner en local ces termes candidats avec les valeurs de $hasTerm$, $prefLabel$ et $altLabel$ de l'ontologie. Si cet alignement réussit, elle appelle la fonction $Misajour$, elle annote le nœud contenant t_i^k par les concepts correspondants, ajoute le(s) termes(s) du web aux valeurs $hasTerm$ de ces concepts, ajoute t_i^k aux valeurs $hasTermNe$ de ces concepts, et retourne *vrai*. Si cet alignement échoue, elle retourne *faux*.

- Si cet ensemble est vide, elle appelle $AlignWeb$.

```
Booléen AlignNe(Ontologie O, Entité Nommée t, Document D, Ensemble<Terme> Echecs)

   variables :  Ensemble<Concept> concepts

   DEBUT
         si (il-existe e appartenant-à O tel-que t similaire-à  e ) alors
            debut
                  concepts = récupérer-concepts(O,t)
                  GénérerPréAnnotation (concepts, D, t)
                  retourner VRAI
             fin
         sinon si ( t inclus-dans Echecs) alors // égal à  une entité nommée traitée et inconnue
             debut
               termesWeb = récupérer-termesWeb(Echecs, t)
               si ( termesWeb non vide ) alors // le terme a été passé au Web
                 debut
                     concepts = Aligner (termesWeb, O)
                     si ( concepts non vide ) alors
                     debut
                         GénérerPréAnnotation (concepts, D, t)
                         EnrichirOntologie (concepts, t)
                         retourner VRAI
                     fin
                 fin
             fin
         retourner FAUX

   FIN.
```

TABLE 3.4 – Fonction d'alignement local d'une entité nommée

3.2.3 Alignement local d'un terme

La fonction *AlignTerm*, présentée dans le tableau 3.5 est utilisée par *Extract-Align* pour rapprocher les termes candidats des valeurs de $hasTerm$, $prefLabel$ ou $altLabel$ pour l'ensemble des concepts de l'ontologie. *AlignTerm* utilise des mesures de similarité adaptées à la comparaison de termes ou groupes nominaux. Ces mesures s'appliquent sur les formes lemmatisées des termes et des labels.

Entrées de la fonction *AlignTerm*

1. l'ontologie $\mathcal{O}$ représentée en OWL ;
2. le terme t_i^k à aligner ;
3. le document $\mathcal{D}$ contenant t_i^k ;
4. l'ensemble des termes traités et non alignés $\mathcal{E}checs$

Sorties de la fonction *AlignTerm*

1. l'ontologie $\mathcal{O}$ enrichie
2. les annotations en RDF des nœuds de documents contenant t_i^k
3. une valeur booléenne *vrai* si t_i^k est aligné et *faux* sinon.

Principales étapes de l'algorithme

La fonction *AlignTerm* procède ainsi :

- Elle vérifie si le terme t_i^k est présent dans l'ontologie, c'est-à-dire égal à une valeur des propriétés $hasTerm$, $prefLabel$ ou $altLabel$ des concepts de l'ontologie. Si tel est le cas, elle appelle la fonction $Misajour$, elle annote le nœud contenant t_i^k par les concepts correspondants et retourne *vrai*. Sinon, la fonction vérifie si le terme peut être aligné à des concepts de l'ontologie via un lien de subsomption et leurs propriétés $hasTerm$, $prefLabel$ ou $altLabel$. Si cet alignement réussit, $AlignTerm$ annote le nœud contenant t_i^k par les concepts trouvés, ajoute t_i^k aux valeurs de $hasTerm$ de ces concepts et retourne *vrai*.

- Si l'alignement en local échoue, $AlignTerm$ cherche si t_i^k a été soumis au Web. Si $t_i^k \in \mathcal{E}checs$, elle récupère les termes du web de t_i^k à partir de l'ensemble $\mathcal{E}checs$.

- Si l'ensemble des termes du web est non vide, $AlignTerm$ tente d'aligner en local ces termes candidats avec les valeurs de $hasTerm$, $prefLabel$ et $altLabel$ de l'ontologie. Si cet alignement réussit, elle appelle la fonction $Misajour$, elle annote le nœud contenant t_i^k par les concepts correspondants, ajoute le(s) terme(s) web et t_i^k aux valeurs $hasTerm$ de ces concepts et retourne *vrai*. Si cet alignement échoue, $AlignTerm$ retourne *faux*.

- Si cet ensemble est vide, elle appelle $AlignWeb$.

```
Booléen AlignTerm(Ontologie O, Terme t, Document D, Ensemble<Terme> Echecs)

   variables :  Ensemble<Concept> concepts

   DEBUT
           si (t inclus-dans O) alors //nous cherchons dans hasLabel puis hasTerm
             debut
                   concepts = récupérer-concepts(O,t)
                   GénérerPréAnnotation (concepts, D, t)
                   retourner VRAI
             fin
           sinon si ( t inclus-dans Echecs) alors // égal à  un terme traité et inconnu
             debut
                   termesWeb = récupérer-termesWeb(Echecs, t);
                   si ( termesWeb non vide ) alors //le terme sans numéro a été passé au Web
                     debut
                           concepts = Aligner (termesWeb, O)
                           si ( concepts non vide ) alors
                             debut
                                  GénérerPréAnnotation (concepts, D, t)
                                  EnrichirOntologie (concepts, t)
                                  retourner VRAI
                             fin
                     fin
             fin
           sinon // un nouveau terme
             debut
                   //Calcul des concepts qui s'alignent avec t utilisant les termes et les labels
                   concepts = Aligner(t, O)
                   si ( concepts non vide ) alors
                       debut
                           GénérerPréAnnotation (concepts, D, t)
                           EnrichirOntologie (concepts, t)
                           retourner VRAI
                       fin
             fin

         retourner FAUX
   FIN.
```

TABLE 3.5 – Fonction d'alignement local d'un terme

3.2.4 Alignement via le Web d'un terme candidat

A chaque fois que le nombre de termes non alignés localement, avec ou sans utilisation de l'ensemble $\mathcal{E}checs$, atteint la taille fixée $\mathcal{T}aillePaquet$, *Extract-Align* fait appel à la fonction *WebAlign* définie dans le tableau 3.6 pour soumettre les termes candidats non alignés au Web. Des patterns lexico-syntaxiques de Hearst sont utilisés pour construire des requêtes contenant l'entité nommée ou le terme non aligné localement. Ces requêtes sont soumises à un moteur de recherche afin de trouver un ensemble de labels candidats qui sont utilisés pour aligner le terme candidat avec les concepts de l'ontologie. Généralement, l'alignement d'un terme long comme "*The Manhattan College Faculty*", échouera très probablement à cause du nombre d'occurrences faible de ce terme sur le Web.

Entrées de la fonction *WebAlign*

1. l'ontologie $\mathcal{O}$ représentée en OWL
2. l'ensemble des termes $\mathcal{T}ermesASoumettreAuWeb$ à soumettre au Web
3. l'ensemble des termes traités et non alignés $\mathcal{E}checs$

Sorties de la fonction *WebAlign*

1. l'ontologie $\mathcal{O}$ enrichie
2. les annotations en RDF des nœuds de documents contenant t_i^k
3. l'ensemble des candidats termes traités et non alignés ajoutés à $\mathcal{E}checs$

Principales étapes de l'algorithme *WebAlign*

La fonction *WebAlign* procède ainsi :

- La fonction cherche des labels de l'entité nommée ou du terme $t_i^k \in$ *TermesASoumettreAuWeb* sur le Web.
- Les labels trouvés pour t_i^k sont à aligner en local avec avec les valeurs de $hasTerm$, $prefLabel$ et $altLabel$ de l'ontologie.
- Si cet alignement réussit, la fonction appelle $Misajour$, elle annote le nœud contenant t_i^k, elle ajoute t_i^k et le(s) label(s) aux termes des concepts de l'ontologie ($hasTerm$). Les labels du Web étant extraits automatiquement, ils sont considérés comme des termes.
- Sinon, elle ajoute le terme et les labels trouvés à $\mathcal{E}checs$.

Les fonctions *Aligner*, *Annoter* et *Misajour*

Les fonctions présentées dans les sections précédentes font appel à des fonctions *Aligner*, *Annoter* et *Misajour*. Elles sont données dans le tableau 3.7.

```
WebAlign (Ontologie O, Ensemble<Terme> TermesASoumettreAuWeb, Ensemble<Pattern> P,
                Ensemble <Document> Corpus, Ensemble<Terme> Echecs, Ensemble<Terme> I)

   variables : Terme t;
                    Ensemble<Terme> termesWeb;
                    Ensemble<Concept> concepts;
   DEBUT
       pour chaque ( t dans TermesASoumettreAuWeb ) faire
         debut
                //recherche de termes pour  t sur le Web
                termesWeb = search(t);
                //Alignement entre les termes du Web et les labels et termes des concepts de O
                concepts = Aligner(O,termesWeb,type(P));
                Si (concepts non vide) alors
                debut
                      GénérerTripletsRDF //Annoter(O, concepts, Pen, D, t)
                      EnrichirOntologie
                      Misajour(I, t)
                fin
                sinon
                      Echecs.ajouter(t, termesWeb);
         fin
   FIN.
```

TABLE 3.6 – Fonction WebAlign : alignement utilisant le Web

La fonction *Annoter* génère les triplet RDF qui décrivent les annotations du nœud contenant le terme que nous lui donnons en argument. Aussi elle ajoute le terme candidat aux entités nommées ou aux termes suivant le type de pattern qui a permis de l'extraire, ajoute les termes provenant du Web aux termes et ainsi enrichie l'ontologie.

La fonction *Aligner* rapproche un terme ou une entité nommée avec les labels et les termes des concepts de l'ontologie. Elle peut utiliser des mesures de similarité simples comme celle du cosinus, celle de Jaccard [TSK05] ou de Lin [Lin98]. Elle peut aussi implémenter des outils existants comme Taxomap [HZSR08].

La fonction *Misajour* supprime toutes les entités nommées et termes inclus dans une entité nommée ou terme aligné. Si t_i^k est aligné, tous les t_p^j appartenant à l'ensemble $\{t_p^j, p = i + q, q \in [0, k-1] et j \in [1, k-q]\}$ sont supprimés de l'ensemble des termes.

3.2.5 Illustration

En guise d'illustration, considérons le fragment de document HTML du tableau 3.8.

L'extrait d'ontologie que nous considérons dans cette illustration est décrit dans la figure 3.3.

Nous supposons que nous faisons appel à l'algorithme *Extract-Align* pour la première fois et donc $\mathcal{E}checs$ est vide.

Nous étiquetons morpho-syntaxiquement ce fragment de documents afin de pouvoir lui appliquer les patterns de localisation des entités nommées et des termes. Nous commençons par les entités nommées.

```
Aligner(Ensemble<Label> set, Ontologie O)
  DEBUT
      rapprocher chaque élément de set avec les labels,
      les termes des concepts de O
  FIN.
```

```
Misajour(Ensemble<Terme> I, Terme terme)
  DEBUT
      pour chaque (t dans I) faire
      debut
        si ( terme contient t ) alors //contient avec numéros
          I.enlever(t);
      fin
  FIN.
```

TABLE 3.7 – Les fonctions *Aligner*, *Annoter* et *Misajour*

CNRIA's second edition is organised by FST and SAT faculties and is about :
≺ul≻
≺li≻distributed systems area≺/li≻
≺li≻knowledge management≺/li≻
≺/ul≻
≺p≻
The conference is located at Gaston Berger University. The invited speaker at the conference Abdourahmane FAYE works on project management at Microsoft SWITZERLAND. Microsoft was implanted in ...
≺/p≻

TABLE 3.8 – Fragment de document HTML

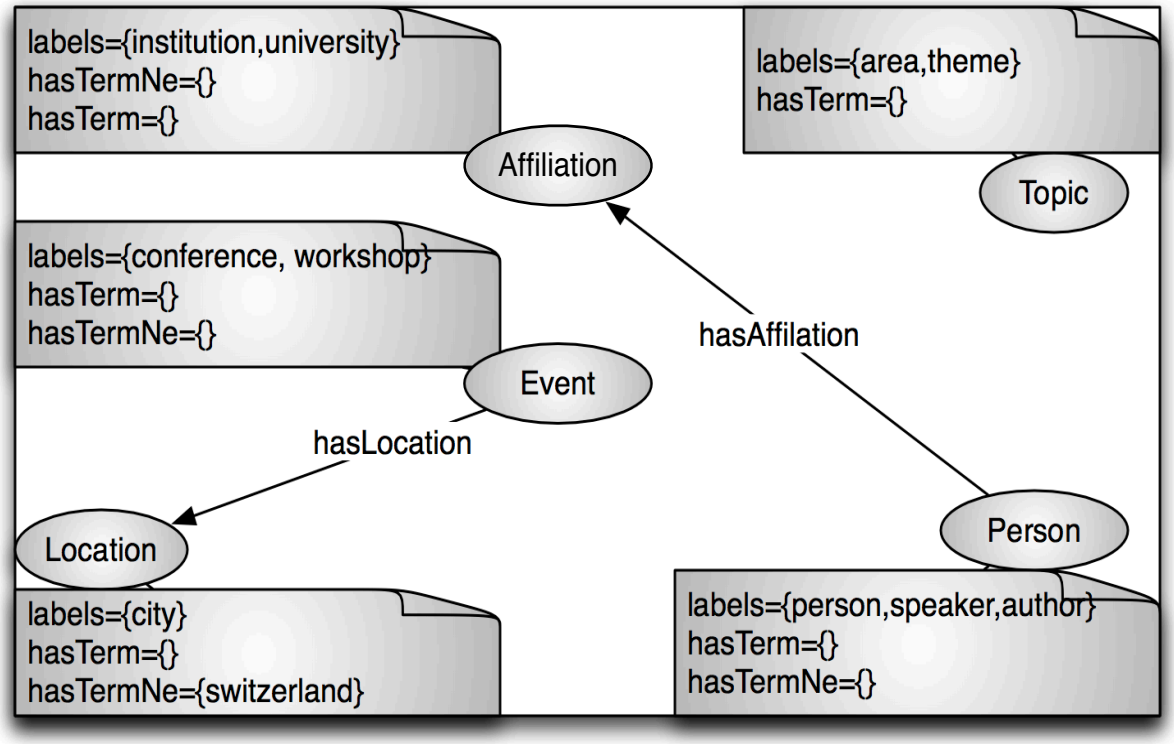


FIGURE 3.3 – Extrait de $\mathcal{O}$ utilisé dans l'illustration

Traitement des entités nommées

Les patterns d'entités nommées (tableau 3.3) permettent d'extraire les entités nommées présentées dans le tableau 3.9.

Taille 1 (I_1)	Taille 2 (I_2)	Taille 3 (I_3)
FST_7, SAT_9, $Gaston_{22}$, $Berger_{23}$, $Microsoft_{38}$, $SWITZERLAND_{39}$, $Microsoft_{40}$	$Abdourahmane_{31}$ $FAYE_{32}$, $Gaston_{22}$ $Berger_{23}$	$Gaston_{22}$ $Berger_{23}$ $University_{24}$

TABLE 3.9 – Résultat de l'extraction des entités nommées

Nous traitons ces entités nommées des plus longues aux plus courtes. Nous commençons par $\mathcal{I}_3$ qui contient un seul élément t^3_{22} qui est *"$Gaston_{22}$ $Berger_{23}$ $University_{24}$"*. *Extract-Align* fait appel à la fonction *AlignNe* qui vérifie si t^3_{22} est dans les entités nommées de l'ontologie. Comme ce n'est pas le cas, elle vérifie si *"Gaston Berger University"* est dans $\mathcal{E}checs$ (entités nommées traitées et non alignées). Comme ce n'est pas le cas, elle retourne *FAUX* à la fonction *Extract-Align* qui insère t^3_{22} dans *TermesASoumettreAuWeb*. $\mathcal{I}_3$ devient vide. La fonction *Extract-Align* fait appel à *WebAlign* qui construit les requêtes suivantes et les soumet au moteur de recherche Google.

1. *"such as Gaston Berger University"*
2. *"including Gaston Berger University"*
3. *", especially Gaston Berger University"*
4. *"Gaston Berger University or other"*
5. *"Gaston Berger University and other"*
6. *"Gaston Berger University is"*

La dernière requête renvoie les extraits suivants :

1. ... the Dakar Cheikh Anta Diop University ; this university, Gaston Berger University, is located in Saint-Louis. There are also other private universities : ... [0.294]
2. GESTES, a research institute within Gaston Berger University, is conducting an assessment of Senegalese women's current land rights, practices and .. [0.185]
3. 10 Jul 2009 ... We can tell you that Gaston Berger University is a ..... College/University ... Employer. If you know more about Gaston Berger University, ... [0.196]

4. In sum, Gaston Berger University is a young, dynamic, and forward-looking institution that has a modern, progressive philosophy and approach which affords ... [0.229]

5. ...

Nous choisissons les dix meilleurs extraits. Nous avons indiqué après chaque extrait la mesure du cosinus calculée entre le vecteur de mots de l'extrait et le vecteur de mots du document en entrée 3.8.

Ces extraits nous permettent de découvrir les labels *college*, *university*, *institution* qui s'alignent avec les labels *university* et *institution* du concept *Affiliation*. Nous en déduisons que t^3_{22} est une instance du concept *Affiliation*.

L'entité nommée *"Gaston Berger University"* est ajouté au *hasTermNe* de *Affiliation* et les triplets de pré-annotation sont générés.

La fonction *Misajour* appliquée à t^3_{22} et l'ensemble des termes et entités nommées extraites supprime les entités nommées *"Gaston*$_{22}$ *Berger*$_{23}$*"*, *"Gaston*$_{22}$*"* et *"Berger*$_{23}$*"* et le terme *"University*$_{24}$*"* de I_2 et I_1. Les triplets de pré-annotation sont générés.

Nous traitons ensuite les entités nommées de taille deux ($\mathcal{I}_2$). La seule entité nommée qu'il contient t^2_{31} qui est *"Abdourahmane*$_{31}$ *FAYE*$_{32}$*"* est soumis aux même processus que t^3_{22}. t^2_{31} n'est pas dans l'ontologie mais les labels du Web nous permettent de l'aligner avec le concept *Person*. L'entité nommée *"Abdourahmane Faye"* est ajouté au *hasTermNe* du concept *Person* et les triplets sont générés comme pour le traitement de t^3_{22}.

Enfin nous traitons l'ensemble des entités nommées de taille une ($\mathcal{I}_1$). *"FST*$_7$*"* et *"SAT*$_9$*"* n'ont pu être alignées ni en local, ni avec le Web qui n'a pas permis d'obtenir de labels. Nous les sauvegardons dans l'ensemble $\mathcal{E}checs$.

"Microsoft$_{38}$*"* n'est ni dans l'ontologie $\mathcal{O}$, ni dans $\mathcal{E}checs.WebAlign$ permet d'obtenir les labels *society*, *application* et *site* qui ne permettent pas de l'aligner avec un concept de notre ontologie. L'entité nommée *"Microsoft"* et ses labels sont sauvegardés dans $\mathcal{E}checs$.

L'entité nommée *"SWITZERLAND*$_{39}$*"* est directement alignée avec le concept *Location* car elle est déjà présente dans l'ontologie. Nous générons les triplets de pré-annotation.

"Microsoft$_{40}$" n'est pas dans l'ontologie mais est présente dans $\mathcal{E}checs$. Les labels obtenus lors de son précédent traitement ne permettent toujours pas de l'aligner avec un concept de notre ontologie.

Après le traitement des entités nommées, l'ontologie de la figure 3.3 enrichie devient celle présentée dans la figure 3.4.

Les triplets de pré-annotation sont présentés dans le tableau 3.10.

L'ensemble $\mathcal{E}checs$ contient maintenant (FST,{}), (SAT,{}) et (Microsoft, {site, society, application}).

Traitement des termes

Les patterns de termes présentés dans *Extract-Align* (tableau 3.3), permettent d'extraire les termes présentés dans le tableau 3.11. Le terme *"University*$_{24}$*"* a été supprimé des termes de longueur un car il est contenu dans l'entité nommée *"Gaston*$_{22}$ *Berger*$_{23}$

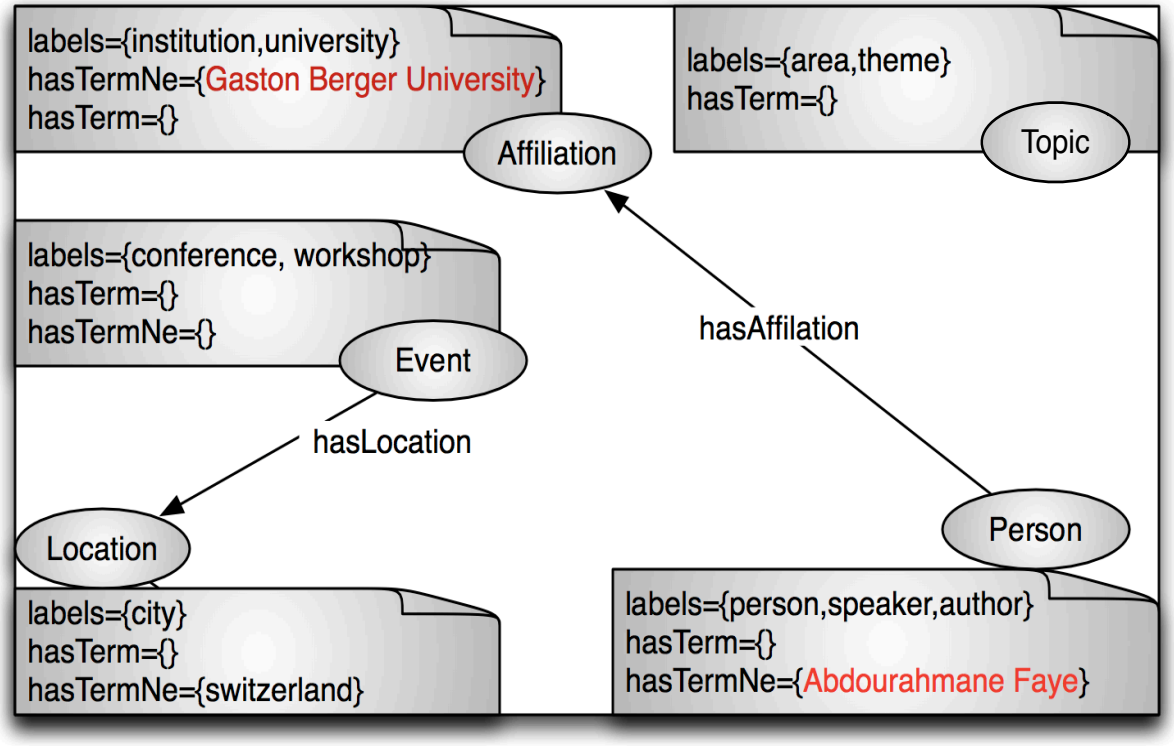


FIGURE 3.4 – Ontologie après traitement des entités nommées

(http ://.../doc.html/body/.../ul/p, hasValueInstance, "Gaston Berger University") ; *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Affiliation)).* *(uri(Affiliation), hasTermNe, "Gaston Berger University")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "Abdourahmane FAYE") ; *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Person)).* *(uri(Person), hasTermNe, "Abdourahmane FAYE")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "SWITZERLAND") ; *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Location)).* *(uri(Location), hasTermNe, "SWITZERLAND")*

TABLE 3.10 – Triplets de pré-annotation résultant du traitement des entités nommées

Taille 1 (I_1)	Taille 2 (I_2)	Taille 3 (I_3)
$edition_3$, $faculties_{10}$, $systems_{13}$, $area_{14}$, $knowledge_{15}$, $management_{16}$, $conference_{18}$, $speaker_{27}$, $project_{35}$, $management_{36}$	$distributed_{12}$ $systems_{13}$, $knowledge_{15}$ $management_{16}$, $invited_{26}$ $speaker_{27}$, $project_{35}$ $management_{36}$	$distributed_{12}$ $systems_{13}$ $area_{14}$

TABLE 3.11 – Extraction des termes

University$_{24}$*"* traitée et alignée précédemment. Nous traitons les termes, des plus longs aux plus courts.

Soit $\mathcal{I}_3$ les termes de taille trois. *Extract-Align* choisit t^3_{12} ou *"distributed*$_{12}$ *systems*$_{13}$ *area*$_{14}$*"*. Elle fait appel à *AlignTerm* qui vérifie que t^3_{12} n'est pas inclus dans les termes de l'ontologie $\mathcal{O}$. Ensuite elle vérifie que *"distributed systems area"* n'est pas inclus dans $\mathcal{E}checs$, les termes déjà soumis au Web. *AlignTerm* tente maintenant d'aligner *"distributed systems area"* avec les labels puis les termes des concepts de l'ontologie. t^3_{12} s'aligne avec le label *area* du concept *Topic*. Nous en déduisons que t^3_{12} est une instance du concept *Topic*.

Le terme *"distributed systems area"* est ajouté au *hasTerm* du concept *Topic* et nous obtenons les triplets de pré-annotation dans le tableau 3.12

AlignTerm renvoie *VRAI*.

Extract-Align applique la fonction *Misajour* à t^3_{12} et l'ensemble des termes extraits. *Misajour* supprime les termes *"distributed*$_{12}$ *systems*$_{13}$*"*, *"systems*$_{13}$*"* et *"area*$_{13}$*"*.

$\mathcal{I}_3$ est vide et *Extract-Align* passe à $\mathcal{I}_2$. Le terme t^2_{15} ou *"knowledge*$_{15}$ *management*$_{16}$*"* est soumis à *TermAlign*. Cette dernière vérifie que t^2_{15} n'est pas dans les termes de l'ontologie $\mathcal{O}$ et que *"knowledge management"* n'a pas été soumis au Web, c'est à dire n'est pas dans $\mathcal{E}checs$. *AlignTerm* tente maintenant d'aligner *"knowledge management"* avec les labels puis avec les termes des concepts de $\mathcal{O}$. t^2_{15} ne s'alignant pas avec un label ou terme, il est gardé dans *TermesASoumettreAuWeb*.

Ensuite le terme t^2_{26} ou *"invited*$_{26}$ *speaker*$_{27}$*"* est soumis à *TermAlign* qui vérifie que t^2_{26} n'est pas dans les termes de l'ontologie $\mathcal{O}$ et que *"invited speaker"* n'a pas été soumis au Web autrement dit n'est pas dans $\mathcal{E}checs$. *AlignTerm* tente maintenant d'aligner *"invited speaker"* avec les labels puis avec les termes des concepts de $\mathcal{O}$ et trouve le concept *Person*.

Le terme *"invited speaker"* est ajouté au *hasTerm* du concept *Person* et nous obtenons les triplets comme ceux obtenus avec le terme t^3_{12} ou *"distributed*$_{12}$ *systems*$_{13}$ *area*$_{14}$*"*. La fonction retourne *VRAI*.

La fonction *Misajour* appliquée à t^2_{26} et l'ensemble des termes extraits supprime le terme *"speaker*$_{27}$*"*.

Ensuite *Extract-Align* soumet le terme t^2_{35} ou *"project*$_{35}$ *management*$_{36}$*"* à *TermAlign* qui vérifie que t^2_{35} n'est pas dans les termes de $\mathcal{O}$ et que *"project management"* n'a pas été soumis au Web autrement dit n'est pas dans $\mathcal{E}checs$. *AlignTerm* tente maintenant d'aligner *"project management"* avec les labels puis avec les termes des concepts de $\mathcal{O}$ et trouve le concept *Topic*.

Le terme *"project management"* est ajouté au *hasTerm* du concept *Topic* et nous obtenons les triplets comme ceux obtenus avec le terme t^2_{26} ou *"invited*$_{26}$ *speaker*$_{27}$*"*. La fonction retourne *VRAI*.

La fonction *Misajour* appliquée à t^2_{35} et l'ensemble des termes extraits supprime les termes *"project*$_{35}$, *"management*$_{36}$*"* et ne supprime pas *"management*$_{16}$*"* qui n'est pas inclus dans la même occurrence que le terme t^2_{35}.

$\mathcal{I}_2$ devient vide et la fonction *Extract-Align* applique *WebAlign* sur *TermesASoumettreAuWeb* qui contient un seul terme *"knowledge*$_{15}$ *management*$_{16}$*"*. *WebAlign* construit les requêtes qui sont posées au moteur de recherche Google qui renvoie des extraits de documents. Ces extraits permettent d'avoir les labels *theme*, *system* et *aim* qui s'alignent avec les labels du concept *Topic* de $\mathcal{O}$.

Le terme *"knowledge management"* est ajouté au *hasTerm* du concept *Topic* et nous obtenons les triplets comme ceux obtenus avec le terme t^2_{26} ou *"invited*$_{26}$ *speaker*$_{27}$*"*. La fonction retourne *VRAI*.

Extract-Align applique la fonction *Misajour* à t^2_{15} et l'ensemble des termes extraits. Cette dernière supprime les termes *"knowledge*$_{15}$*"* et *"management*$_{16}$*"*.

Nous passons maintenant aux termes de taille une $\mathcal{I}_1$. *Extract-Align* applique la fonction *AlignTerm* au terme t^1_{18} ou *"conference*$_{18}$*"*. *AlignTerm* vérifie que *conference* est un label du concept *Event*. Nous obtenons les triplets comme ceux obtenus avec le terme t^2_{26} ou *"invited*$_{26}$ *speaker*$_{27}$*"*. La fonction retourne *VRAI*.

Extract-Align applique la fonction *AlignTerm* au terme t^1_3 ou *"edition*$_3$*"*. *AlignTerm* vérifie que *edition* n'est inclus ni dans les labels, ni dans les termes des concepts de l'ontologie $\mathcal{O}$. *AlignTerm* tente de l'aligner avec ces labels et termes. t^1_3 ne s'alignant pas avec un label ou terme, il est gardé dans *TermesASoumettreAuWeb*.

Extract-Align applique la fonction *AlignTerm* au terme t^1_{10} ou *"faculty*$_{10}$*"*. *AlignTerm* vérifie que *"faculty"* n'est inclus ni dans les labels, ni dans les termes des concepts de l'ontologie $\mathcal{O}$. *AlignTerm* tente de l'aligner avec ces labels et termes. t^1_{10} s'aligne avec un label du concept *Location*.

Le terme *"faculty"* est ajouté au *hasTerm* du concept *Location* et *AlignTerm* génère les triplets avant de retourner *VRAI*.

Maintenant que $\mathcal{I}_1$ est vide, *Extract-Align* fait appel à *WebAlign*. Cette dernière cherche des labels pour t^1_3 sur le Web en construisant des requêtes avec les patterns de Hearst [Hea92]. Ces requêtes sont posées au moteur de recherche Google qui retourne des extraits de documents. Ces extraits sont comparés avec le *fragment de document* et des labels sont extraits. Cependant ces labels ne permettent pas de rapprocher t^1_3 à un concept de $\mathcal{O}$. Nous gardons *"edition"* et ses labels dans $\mathcal{E}checs$.

Après le traitement des termes l'ontologie devient celle de la figure 3.5.

Les triplets de pré-annotation sont présentés dans le tableau 3.12.

L'ensemble $\mathcal{E}checs$ contient maintenant (FST,{}), (SAT,{}) et (Microsoft, {site, society, application}), et (edition,{l_i, i=1,n}).

Conclusion

Dans ce chapitre, nous avons présenté notre approche *Extract-Align* d'extraction et d'alignement qui permet : (i) d'extraire des entités nommées et des termes en utilisant des patterns d'extraction indépendants du domaine, (ii) de les aligner avec des concepts de l'ontologie soit localement lorsqu'ils sont similaires à un terme ou une EN de la composante lexicale, soit en utilisant des termes extraits de documents du Web, (iii) d'enrichir l'ontologie avec les entités nommées et les termes alignés en utilisant les propriétés $hasTerm$ et $hasTermNe$, (iv) enfin, de pré-annoter les nœuds de document où ces termes ont été localisés en utilisant les propriétés $containInstanceOf$ et $hasValueInstance$.
Dans le chapitre suivant, nous présentons les règles d'annotation définitive des nœuds qui prennent en compte à la fois les pré-annotations des nœuds de document et leur voisinage structurel.

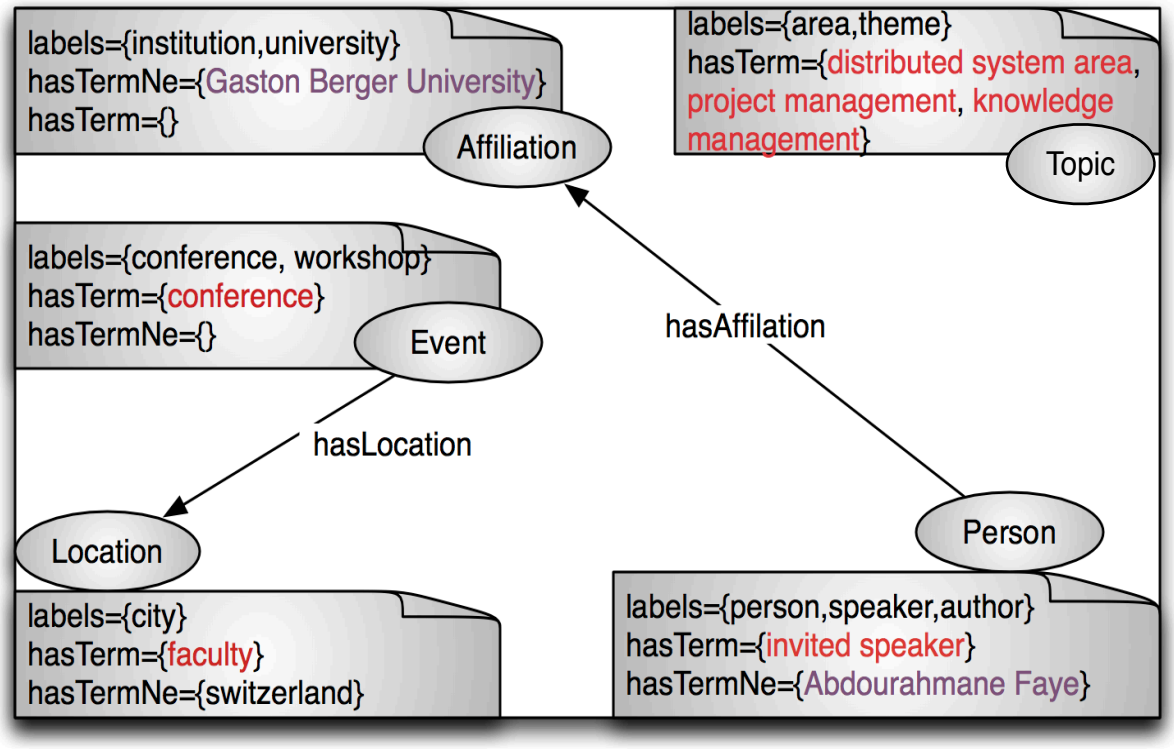


FIGURE 3.5 – Ontologie après traitement des termes

(http ://.../doc.html/body/.../ul/p, hasValueInstance, "distributed systems area"); *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Topic)).* *(uri(Affiliation), hasTerm, "distributed systems area")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "project management"); *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Topic)).* *(uri(Affiliation), hasTerm, "project management")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "knowledge management"); *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Topic)).* *(uri(Affiliation), hasTerm, "knowledge management")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "invited speaker"); *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Person)).* *(uri(Affiliation), hasTerm, "invited speaker")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "faculty"); *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Location)).* *(uri(Affiliation), hasTerm, "faculty")*
(http ://.../doc.html/body/.../ul/p, hasValueInstance, "conference"); *(http ://.../doc.html/body/.../ul/p, containInstanceOf, uri(Event)).* *(uri(Affiliation), hasTerm, "conference")*

TABLE 3.12 – Triplets de pré-annotation résultant du traitement des termes

Chapitre 4

ANNOTATION DES DOCUMENTS ET INTERROGATION

Sommaire

Introduction

L'objectif de *SHIRI-Annot* est d'annoter les nœuds de documents conformément au modèle d'annotation que nous avons défini. Nous avons défini un ensemble de règles déclaratives exprimées en logique du premier ordre pour générer l'ensemble des annotations. Ces règles exploitent la sémantique associée aux nœuds par *SHIRI-Extract* et leur proximité structurelle.

L'annotation d'un nœud est définie en fonction de la présence d'un ou de plusieurs termes ou entités nommées alignés dans ce nœud et des concepts avec lesquels ils ont été alignés. Un nœud annoté est soit une instance d'un ou de plusieurs concepts ontologiques, soit une instance de $PartOfSpeech$ ou de $SetOfc$ ($SetOfConcept$).

Les règles génèrent les annotations de la relation $neighborOf$ entre les nœuds en tenant compte de la longueur du chemin les reliant dans l'arbre repésentant le document et de leur sémantique.

La base d'annotations RDF ainsi construite peut être interrogée en utilisant des requêtes SPARQL. Nous pésentons un ensemble de requêtes types reformulées avec les métadonnées du modèle d'annotation, à partir d'une requête utilisateur exprimée avec les métadonnées de l'ontologie. Ces reformulations peuvent être construites en appliquant différentes fonctions de transformations. Elles permettent d'atteindre tous les nœuds réponses triées selon le niveau d'agrégation des termes ou entités nommées repérés dans ces nœuds.

Dans ce chapitre, nous commençons par présenter les règles d'annotation des nœuds et de la relation $neighborOf$. Ensuite, nous présentons l'interrogation des annotations dans le module *SHIRI-Querying*.

4.1 Règles d'annotation

L'objectif des règles d'annotation est de générer un ensemble d'annotations sémantiques conforme au modèle d'annotation à partir des pré-annotations utilisant les propriétés $containInstanceOf$ et $hasValueInstance$ produites par *SHIRI-Extract*.

Nous présentons quelques notations avant de présenter les règles d'annotation permettant de typer les nœuds et d'annoter la relation $neighborOf$ entre les nœuds.

4.1.1 Notations

Les règles utilisent les propriétés RDFS suivantes que nous exprimons en logique des prédicats. Soient $c \in \mathcal{C}_\mathcal{O}$ et $r \in \mathcal{R}_\mathcal{O}$:

- $domain(r, c)$ est vrai si c est un domaine de r.
- $range(r, c)$ est vrai si c est un co-domaine de r.
- $subClassOf(c, c')$ est vrai si c' subsume ou est équivalent à c.

Pour plus de clarté dans l'expression des règles nous définissons les prédicats unaires suivants :

- $singleTerm(n)$ est vrai pour tout nœud n ne contenant qu'un seul terme ou une seule entité nommée.
- $singleConcept(n)$ est vrai pour tout nœud n contenant un ou plusieurs termes ou entités nommées qui sont uniquement alignés à des concepts comparables.
- $path(n_i, n_j)$ est vrai si la longueur du chemin dans l'arbre DOM entre le nœud n_i et le nœud n_j est inférieure à une valeur d.

4.1.2 Génération du typage des nœuds

Le choix de la métadonnée à utiliser pour l'annotation d'un nœud dépend de la présence d'un ou plusieurs termes ou entités nommées dans ce nœud. Ces métadonnées sont définies dans le modèle d'annotation (Figure 2.3) : *Concept*, $PartOfSpeech$ et $SetOfc$.

Présentation des patterns de règles

Les règles de typage des nœuds permettent d'annoter un nœud soit comme une instance d'un concept du domaine ($\mathcal{C}_\mathcal{O}$), soit comme une instance d'un concept spécifique à la tâche d'annotation ($\mathcal{C}_\mathcal{S}$) exprimant un type d'agrégation structurelle.

1. Si un nœud n ne contient qu'un seul terme, ou une seule entité nommée, aligné à un concept $c \in \mathcal{C}_\mathcal{O}$, ce nœud est typé par c.

   $singleTerm(\mathrm{n}) \wedge containInstanceOf(\mathrm{n}, c) \rightarrow type(\mathrm{n}, c)$

   Exemple :
   $singleTerm(http://www..../cfp.html/body/p) \wedge$
   $containInstanceOf(http://www..../cfp.html/body/p, Person)$
   $\rightarrow type(http://www..../cfp.html/body/p, Person)$

   Notons qu'un même nœud peut être typé par plusieurs concepts non comparables si le seul terme ou la seule entité nommée qu'il contient est alignée à plusieurs concepts non comparables. Dans ce cas, $\exists c_i, c_j \; c_i \neq c_j$ tel que $containInstanceOf(n, c_i) \wedge containInstanceOf(n, c_j) \wedge singleTerm(n)$

2. Rechercher des instances en utilisant leur label est particulièrement fréquent en recherche d'information. Nous générons les labels des instances de concepts nommés afin de pouvoir exploiter cette propriété dans la requête de l'utilisateur.

   Si un nœud n est typé par un concept c, si n est relié par la propriété *hasValueInstance* à un littéral *val* tel que *val* appartient à l'ensemble des valeurs de la propriété $hasTermNe$ du concept c, nous inférons la propriété *label* entre n et *val*.

   $type(n, c) \wedge hasTermNe(c, val) \wedge hasValueInstance(n, val) \rightarrow label(n, val)$

   Exemple :
   $type(http://www..../cfp.html/body/p), Person) \wedge$
   $hasTermNe(Person, "John\ Woe") \wedge$
   $hasValueInstance(http://www..../cfp.html/body/p, "John\ Woe")$
   $\rightarrow label(http://www..../cfp.html/body/p, "John\ Woe")$

3. Si un nœud n contient plusieurs termes ou entités nommées de plusieurs concepts non comparables, alors ce nœud est typé par la métadonnée $PartOfSpeech$.

$containInstanceOf(n, c) \wedge \neg singleTerm(n) \wedge \neg singleConcept(n)$
$\rightarrow type(n, PartOfSpeech)$

Exemple :
$containInstanceOf(http://www..../cfp.html/body/p, Person) \wedge$
$\neg singleTerm(http://www..../cfp.html/body/p) \wedge$
$\neg singleConcept(http://www..../cfp.html/body/p)$
$\rightarrow type(http://www..../cfp.html/body/p, PartOfSpeech)$

Tout nœud de type $PartOfSpeech$ est indexé en utilisant la relation $isIndexedBy$ par tous les concepts c reliés à ce nœud par la relation $containInstanceOf$.

$type(n, PartOfSpeech) \wedge containInstanceOf(n, c) \rightarrow isIndexedBy(n, c)$
$isIndexedBy(n, c) \wedge subClassOf(c, c') \rightarrow isIndexedBy(n, c')$

Exemple :
$type(http://www..../cfp.html/body/p, PartOfSpeech) \wedge$
$containInstanceOf(http://www..../cfp.html/body/p, Person)$
$\rightarrow isIndexedBy(http://www..../cfp.html/body/p, Person)$

4. Nous générons les instances de $SetOfConcept$ pour les nœuds susceptibles de représenter un ensemble d'instances de type comparables. Si un nœud contient plusieurs termes ou entités nommées et si celles-ci sont uniquement alignées à des concepts comparables, ce nœud est typé par la métadonnée s qui est décrite dans la propriété $hasSet$ du concept c_i le concept ancêtre de cet ensemble de concepts.

$containInstanceOf(n, c_i) \wedge singleConcept(n) \wedge \neg singleTerm(n)$
$\wedge \neg \exists\, c_j(containInstanceOf(n, c_j) \wedge subClassOf(c_i, c_j)) \wedge hasSet(c_i, s) \rightarrow type(n, s)$

Exemple :
Soit un noeud $http://www..../cfp.html/body/p$ qui contient uniquement une entité nommée alignée au concept $ComiteeMember$ et une entité nommée alignée au concept $Person$, et $ComiteeMember$ est une sous-classe de $Person$, alors la règle va typer le nœud par $SetOfPerson$.

$containInstanceOf(http://www..../cfp.html/body/p, Person) \wedge$
$singleConcept(http://www..../cfp.html/body/p)$
$\wedge \neg singleTerm(http://www..../cfp.html/body/p) \wedge \nexists c_j(containInstanceOf(http:$
$//www..../cfp.html/body/p, c_j) \wedge subClassOf(Person, c_j))$
$\wedge hasSet(Person, SetOfPerson) \rightarrow type(n, SetOfPerson)$

4.1.3 Génération de relations de voisinage entre nœuds

Par ailleurs, les règles d'annotation exploitent la présence d'instances susceptibles de partager une relation sémantique dans des nœuds voisins pour générer des annotations reliant ces deux nœuds par la propriété $neighborOf$ définie dans le modèle d'annotation.

La propriété $neighborOf$ relie deux nœuds n_i et n_j voisins tel que n_i contient un terme ou une entité nommée aligné à un concept c_i et tel que n_j contient un terme ou une entité nommée aligné à un concept c_j et tel qu'il existe une relation r dans l'ontologie pour laquelle $domain(r, c_i)$ et $range(r, c_j)$.

Les nœuds peuvent être de type $PartOfSpeech$ (indexés par c_i ou c_j) ou $SetOfConcept$ ($SetOfc_i$ ou $SetOfc_j$) ou $Concept$ (c_i ou c_j).

$n_i \neq n_j \wedge containInstanceOf(n_i, c_i) \wedge containInstanceOf(n_j, c_j) \wedge domain(r, c_i)$
$\wedge range(r, c_j) \wedge path(n_i, n_j) \rightarrow neighborOf(n_i, n_j)$

Exemple :
Soit deux nœuds n_1 =*http ://www.cfp.html/body/p*, et n_2 = *http ://www.cfp.html/body/p /ul/li* voisins (distance $\leq d$) typés respectivement par $PartOfSpeech$ indexé par $Event$ et $Topic$ et par $Location$.

$n_1 \neq n_2 \wedge containInstanceOf(n_1, Event) \wedge containInstanceOf(n_2, Location) \wedge domain(hasLocation, Event) \wedge range(hasLocation, Location) \wedge path(n_1, n_2) \rightarrow neigborOf(n_1, n_2)$

4.1.4 Exemple d'application des règles d'annotations

L'application des règles à l'ensemble des pré-annotations représentées dans l'exemple décrit dans la figure 4.1 génère les annotations représentées dans la figure 4.2. Nous supposons que la distance d est fixée à 3.

Dans le document *Document 1*, les deux nœuds contiennent des termes ou des entités nommées associés à des concepts différents tels que $Topic$, $Date$, $Event$ ou $Location$. Ils sont donc considérés comme des instances de $PartOfSpeech$ et indexés par ces concepts. De plus, ces deux nœuds sont considérés comme deux nœuds voisins (reliés par *neighborOf*) : ils contiennent des termes de concepts qui sont domaine et co-domaine d'une relation sémantique de l'ontologie. Par exemple, l'entité nommée "$ESWC2009$" associée à $Event$ peut être reliée à "$Heraklion$" par la relation sémantique $hasLocation$.

Dans le document *Document 2*, le premier nœud ne contient qu'une entité nommée associée au concept $Event$. Ce nœud est donc un $Event$ qui a pour *label "12th International Conference on Data Warehousing and Knowledge Discovery*". Le deuxième nœud contient deux entités nommées "*Bilbao*" et "*Spain*" associées au concept $Location$. Ce nœud est donc un $SetOfLocation$.

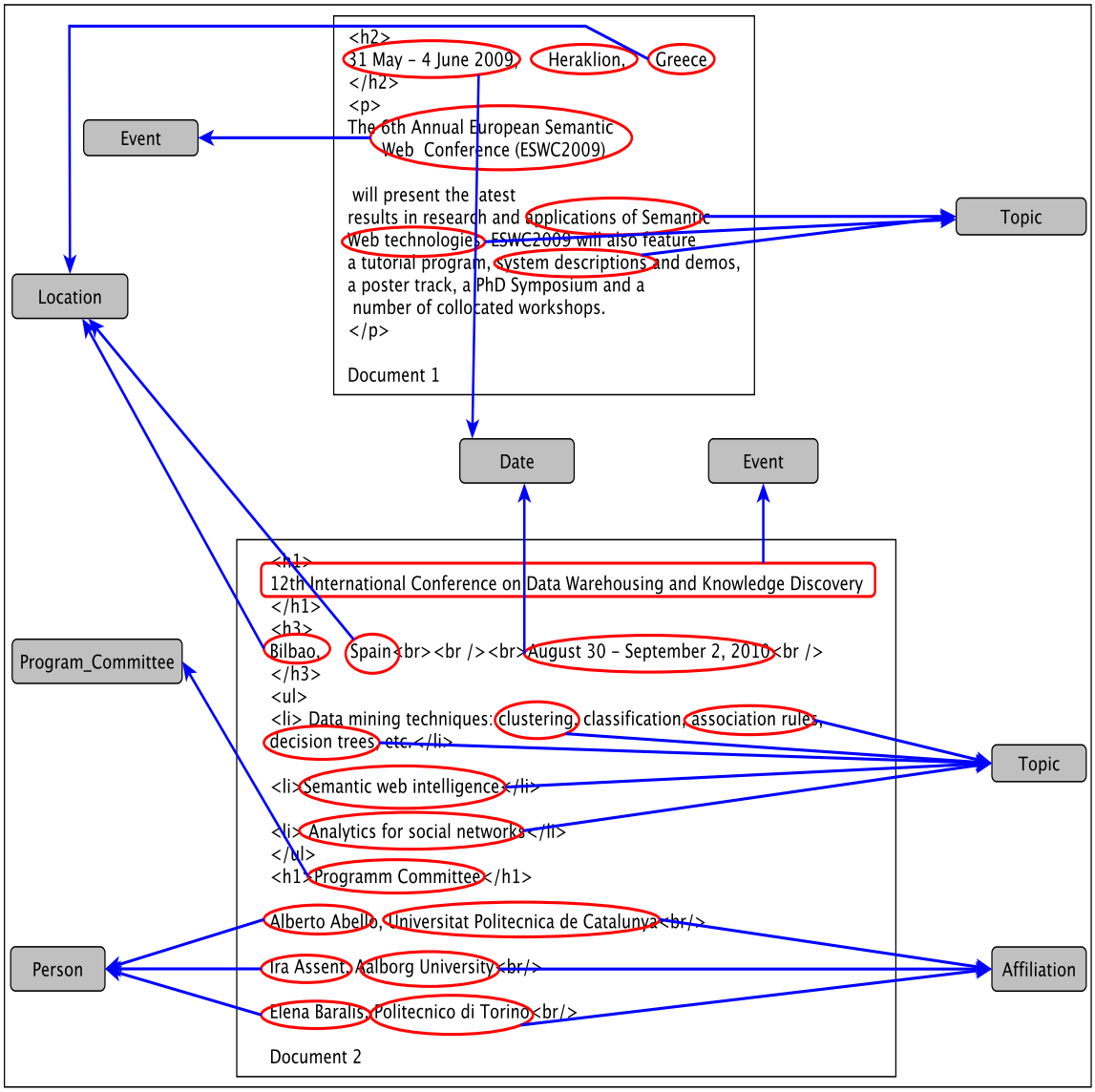


FIGURE 4.1 – Extraction des termes et entités nommées et leur alignement avec des concepts de l'ontologie

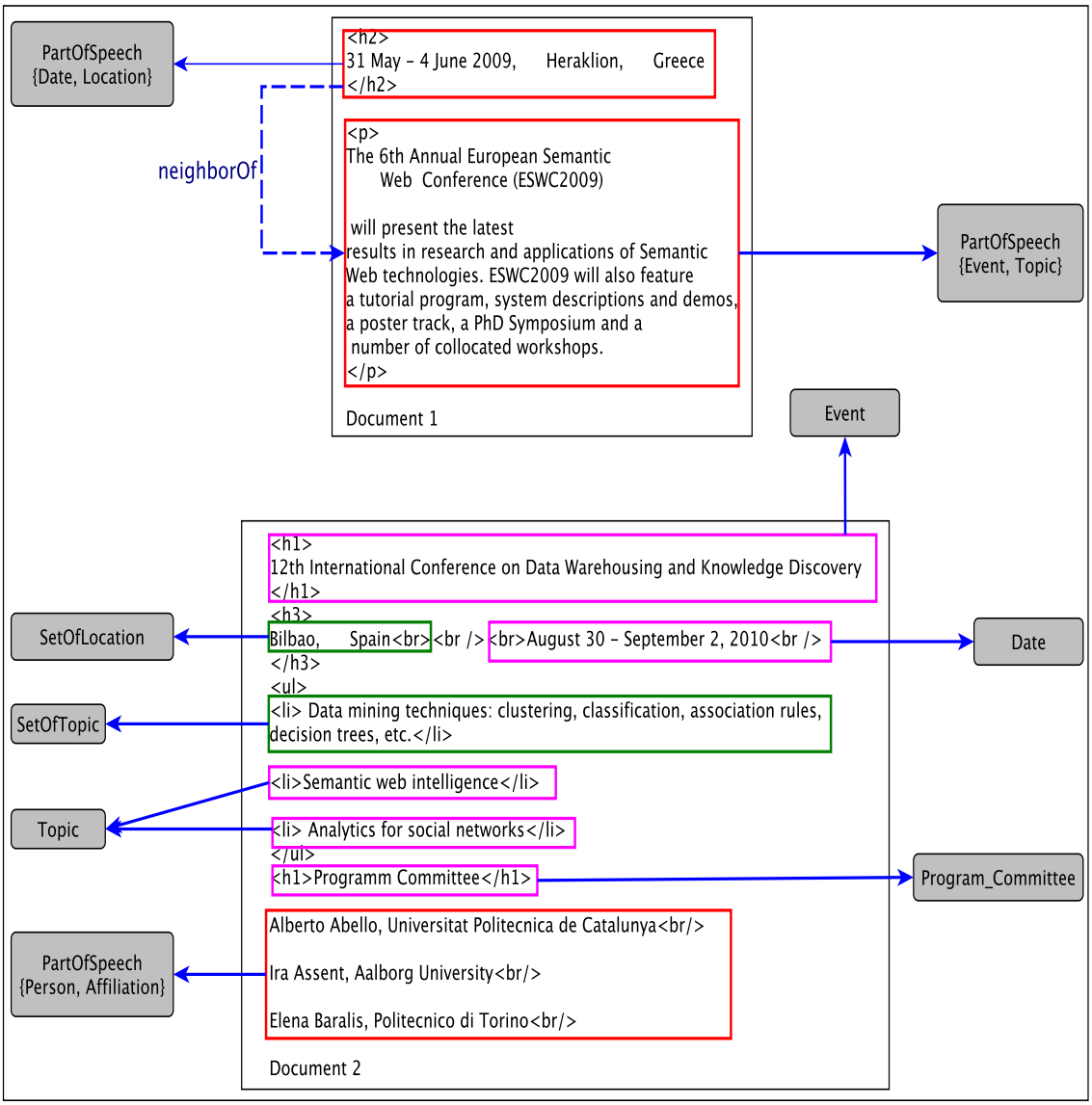


FIGURE 4.2 – Annotation des nœuds de documents comportant des termes ou entités nommées extraits et alignés avec des concepts de l'ontologie

4.2 Interrogation des annotations

SHIRI-Querying (cf. Figure 2.7) est le module qui permet de traiter les requêtes pour interroger la base d'annotations. Ce module exploite le modèle d'annotation pour construire, trier et exécuter les reformulations à partir d'une requête utilisateur exprimée à l'aide des métadonnées définies dans l'ontologie. Il utilise le langage SPARQL pour interroger la base des annotations RDF.

4.2.1 Définitions préliminaires

Nous introduisons tout d'abord quelques notions préliminaires. Les graphes RDF et les patrons de graphe élémentaires sont définis comme suit [W3Cb] :

Graphe RDF. Soient les ensembles I, B et L (respectivement des IRIs[7], nœuds et Littéraux) infinis et disjoints deux à deux. Un graphe RDF est un ensemble de triplets RDF $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ où s est le sujet, p le prédicat et o l'objet.

Patrons de graphe élémentaires. Soit l'ensemble de variables V disjoint des ensembles I, B, et L. Un patron de triplet est un triplet $(s, p, o) \in (I \cup V) \times (I \cup V) \times (I \cup V \cup L)$. Un patron de graphe élémentaire P est un ensemble de patrons de triplets. Dans un triplet, une variable est précédée par un point d'interrogation (e.g. $?v$).

Requête SPARQL. Une requête SPARQL est construite à partir de patrons de graphe élémentaires en utilisant des constructeurs, tels que union, des clauses, telles que OPTIONAL et des filtres. Un filtre est une expression booléenne qui permet de contraindre l'ensemble des réponses à la requête. Dans ce qui suit, nous considérons uniquement l'égalité et l'inclusion entre variables et littéraux d'un filtre.

4.2.2 Types de requête utilisateur

Dans notre approche, une requête utilisateur est définie par le quadruplet (P, S, F, D) où :

- P est un patron de graphe élémentaire conforme au modèle d'annotation $\mathcal{A}$.
- S est l'ensemble des variables utilisées dans la clause SELECT.
- F est un filtre.
- D est l'ensemble des triplets RDF de la base d'annotations à interroger.

Une requête peut être formulée différemment selon le degré d'agrégation des termes et des entités nommées repérés dans les nœuds de document que l'on souhaite atteindre.

7. Internationalized Resource Identifier

Soit une requête utilisateur qui cherche "*les noms des conférences pour lesquels l'un des thèmes est Semantic Web*" sur les annotations décrites dans la figure 4.2.

Requête 1. L'une des formulations possible en SPARQL de cette requête est de rechercher les nœuds qui sont instances de *Event* et de *Topic* tels que le nœud *Topic* contienne le litéral *"Semantic Web"*. Cette requête q_1 est définie par (P_1,F_1,S_1,D) où :

P_1 : { $(?conf, rdf:type, Event)$, $(?t, rdf:type, Topic)$,
$(?conf, hasTopic, ?t)$, $(?conf, label, ?cName)$
$(?t, hasText, ?v)$ }
F_1 : $regex(?v, "Semantic\ Web")$
S_1 : { $?cName$}
D : Ensemble des annotations de *Document 1* et *Document 2*.

Dans la figure 4.2, cette requête q_1 n'obtient pas de réponse.

Requête 2. Une autre formulation possible de la requête est de rechercher les nœuds qui sont instances de *Event* et de *Topic*, qui ne sont pas liés par la relation *hasTopic* mais pour lesquelles il existe une relation de voisinage. Le nœud instance de *Topic* contient le litéral *"Semantic Web"*. Cette requête q_2 est définie par (P_2,F_2,S_2,D) où :

P_2 : { $(?conf, rdf:type, Event)$, $(?t, rdf:type, Topic)$,
$(?conf, neighborOf, ?t)$, $(?conf, label, ?cName)$
$(?t, hasText, ?v)$ }
F_2 : $regex(?v, "Semantic\ Web")$
S_2 : { $?cName$}
D : Ensemble des annotations de *Document 1* et *Document 2*.

Dans la figure 4.2, cette requête q_2 obtient la réponse : *"12th International Conference on Data Warehousing and Knowledge Discovery"*.

Requête 3. Une autre formulation possible de la requête est de rechercher les nœuds qui sont instances de *Event* et de *SetOfTopic* pour lesquelles il existe une relation de voisinage. Le nœud instance de *SetOfTopic* contient le litéral *"Semantic Web"*. Cette requête q_3 est définie par (P_3,F_3,S_3,D) où :

P_3 : { $(?conf, rdf:type, Event)$, $(?t, rdf:type, SetOfTopic)$,
$(?conf, neighborOf, ?t)$, $(?conf, label, ?cName)$
$(?t, hasText, ?v)$ }
F_3 : $regex(?v, "Semantic\ Web")$
S_3 : { $?cName$}
D : Ensemble des annotations de *Document 1* et *Document 2*.

Dans la figure 4.2, cette requête q_3 n'obtient pas de réponse.

Requête 4. Une autre formulation possible de la requête est de rechercher les nœuds instances de *PartOfSpeech* indexé par *Event*, *Topic* et qui contient le littéral *"Se-*

mantic Web". Cette requête q_4 est définie par (P_4,F_4,S_4,D) où :

P_4 : { $(?p, rdf:type, PartOfSpeech)$, $(?p, isIndexedBy, Event)$, $(?p, isIndexedBy, Topic)$, $(?p, hasText, ?v)$ }
F_4 : $regex(?v, "Semantic\ Web")$
S_4 : { $?v$}
D : Ensemble des annotations de *Document 1* et *Document 2*.

Dans la figure 4.2, cette requête q_4 obtient la réponse : *"The 6th Annual European Semantic Web Conference ... collocated workshops"*.

Requête 5. Une autre formulation possible de la requête est de rechercher deux nœuds voisins instances de $PartOfSpeech$, l'un indexé par $Event$, et l'autre par $Topic$ qui contient le littéral *"Semantic Web"*. Cette requête q_5 est définie par (P_5,F_5,S_5,D) où :

P_5 : { $(?p1, rdf:type, PartOfSpeech)$, $(?p1, isIndexedBy, Event)$, $(?p2, rdf:type, PartOfSpeech)$,$(?p2, isIndexedBy, Topic)$, $(?p1, neighborOf, ?p2)$, $(?p1, hasText, ?v1)$, $(?p2, hasText, ?v2)$ }
F_5 : $regex(?v2, "Semantic\ Web")$
S_5 : { $?v1$}
D : Ensemble des annotations de *Document 1* et *Document 2*.

Dans la figure 4.2, cette requête q_5 n'obtient pas de réponse.

Dans notre approche, nous considérons que la pertinence des nœuds réponses est liée au fait que les instances soient plus ou moins agrégées. Par exemple, on préférera un nœud typé par le concept du domaine plutôt qu'un nœud typé par un sous-concept de $SetOfConcept$ ou un $PartOfSpeech$ indexé par le concept.

4.2.3 *SHIRI-Querying*

L'objectif de *SHIRI-Querying* est de définir un ensemble de fonctions qui permettent de construire automatiquement les différentes reformulations à partir d'une requête utilisateur exprimée avec les concepts et les relations de l'ontologie. Ces fonctions exploitent les métadonnées du modèle d'annotation que nous avons défini afin d'atteindre tous les nœuds réponses et de trier ces réponses en fonction de leur agrégation dans les documents.

Prenons l'exemple d'une requête simple dont le patron de graphe est $P = \{(?c1, r, ?c2)$, $(?c1, type, c1)$, $(?c2, type, c2)\}$ les reformulations possibles sont :

- $P_1 = \{(?c1, neighborOf, ?c2), (?c1, type, c1), (?c2, type, c2)\}$
- $P_2 = \{(?c1, neighborOf, ?c2), (?c1, type, c1), (?c2, type, SetOfc2)\}$
- $P_3 = \{(?c1, neighborOf, ?c2), (?c1, type, SetOfc1), (?c2, type, c2)\}$
- $P_4 = \{(?c1, neighborOf, ?c2), (?c1, type, SetOfc1), (?c2, type, SetOfc2)\}$
- $P_5 = \{(?c1, neighborOf, ?c2), (?c1, type, PartOfSpeech)$, $(?c1, isIndexedBy, c1)$, $(?c2, type, c2)\}$

- $P_6 = \{(?c1, neighborOf, ?c2), (?c2, type, PartOfSpeech)$, $(?c2, isIndexedBy, c2)$, $(?c1,\ type, c1)\}$
- $P_7 = \{(?c1, neighborOf, ?c2), (?c1, type, PartOfSpeech)$, $(?c1, isIndexedBy, c1)$, $(?c2,\ type, SetOfc2)\}$
- $P_8 = \{(?c1, neighborOf, ?c2), (?c2, type, PartOfSpeech)$, $(?c2, isIndexedBy, c2)$, $(?c1,\ type, SetOfc1)\}$
- $P_9 = \{(?c1, neighborOf, ?c2), (?c2, type, PartOfSpeech)$, $(?c2, isIndexedBy, c2)$ $(?c1,\ type, PartOfSpeech), (?c1, isIndexedBy, c1)\}$
- $P_{10} = \{(?c1, type, SetOfc1)\}$ si $c1$ subsume $c2$ ou
  $P_{10} = \{(?c1, type, SetOfc2)\}$ si $c2$ subsume $c1$ ou
  $P_{10} = \{(?c12, type, PartOfSpeech), (?c12, isIndexedBy, c2), (?c12, isIndexedBy, c1)\}$
  sinon.

D'une manière générale, pour un patron de graphe à n concepts, nous pouvons distinguer les transformations suivantes :

- La transformation qui substitue toutes les relations par $neighborOf$.
- Les transformations appelées *set* qui substituent des variables de k types différents en une variable de type $SetOfC$, k allant de 1 à n. Cette transformation n'est possible que si les k concepts sont comparables et C étant le concept ancêtre.
- Les transformations appelées *pos* qui substituent des variables de k types différents en une variable de type $PartOfSpeech$ indexée par C_i, i allant de 1 à k, k allant de 1 à n. Cette transformation n'est possible que si les k concepts ne sont pas comparables.
- Les transformations qui combinent *pos* et *set*.

Le nombre de reformulations par transformation *set* d'un c en $SetOfc$ ou le nombre de reformulations par transformation *pos* d'un c en $PartOfSpeech$ indexée par c est au pire égal à $\sum_{k=1}^{n} C_n^k = 2^n - 1$.

Il est clair que le nombre de reformulations, obtenues à partir des différentes transformations, croît de manière exponentielle avec le nombre de concepts. Cependant, plusieurs reformulations ne seront pas construites car elles ne seront sémantiquement pas possibles. Par exemple, pour une requête :

$P = \{(?c1, r, ?c2), (?c2, r, ?c3)$ $(?c1, type, c1)$, $(?c2, type, c2)$, $(?c3, type, c3)\}$,
appliquer la transformation *set* à $c1$, à $c2$ ensuite à $c3$ donne les requêtes suivantes :

$\{(?c1, neighborOf, ?c2), (?c2, neighborOf, ?c3)$ $(?c1, type, SetOfc1)$, $(?c2, type, c2)$, $(?c3,$ $type, c3)\}$,
$\{(?c1, neighborOf, ?c2), (?c2, neighborOf, ?c3)$ $(?c1, type, c1)$, $(?c2, type, SetOfc2)$,$(?c3,$

$type, c3)\}$
$\{(?c1, neighborOf, ?c2), (?c2, neighborOf, ?c3)$ $(?c1, type, c1)$, $(?c2, type, c2)$, $(?c3, type,$ $SetOfc3)\}$.

Si $c1$, $c2$ et $c3$ forment un ensemble comparable tel que $c1$ subsume $c2$ et $c2$ subsume $c3$, appliquer la transformation *set* à deux variables de types $c1$ et $c2$, de types $c1$ et $c3$ et ensuite de types $c2$ et $c3$ donne les requêtes suivantes :

$\{(?c12, neighborOf, ?c3)$, $(?c12, type, SetOfc1)$, $(?c3, type, c3)\}$
$\{(?c13, neighborOf, ?c2)$, $(?c13, type, SetOfc1)$,(?c2, type, c2),$\}$
$\{(?c1, neighborOf, ?c23)$, $(?c23, type, SetOfc2)$, $(?c1, type, c1)\}$

Appliquer la transformation *set* aux trois variables de types $c1$, $c2$ et $c3$ donne la requête suivante :
$\{(?c123, type, SetOfc1)\}$

Dans ce cas de figure, il n y a pas aucune reformulation utilisant la transformation *pos*.

D'une manière générale, les éléments suivants peuvent réduire considérablement le nombre de reformulations :

- Dans la réalité, le nombre de concepts dans une requête utilisateur n'est pas très grand.

- Les reformulations peuvent être construites selon une fonction d'ordre dépendant du nombre de transformations *set* et *pos*. Ce qui permet d'arrêter la génération des reformulations pour un ordre donné.

- Aussi, étant donné que les règles que nous avons définies sont déclaratives, certaines pourraient être supprimées ou relâchées par l'utilisateur. Par exemple, dans un cas extrême, on suppose que tous les nœuds sont de type $PartOfSpeech$ reliés par la relation $neighborOf$.

  $containInstanceOf(n, c) \wedge \neg type(n, PartOfSpeech) \rightarrow type(n, PartOfSpeech)$
  $type(n, PartOfSpeech) \wedge containInstanceOf(n, c) \rightarrow isIndexedBy(n, c)$
  $isIndexedBy(n, c) \wedge subClassOf(c, c') \rightarrow isIndexedBy(n, c')$
  $n_1 \neq n_2 \wedge containInstanceOf(n_1, c_i) \wedge containInstanceOf(n_2, c_j) \wedge$
  $domain(r, c_i) \wedge range(r, c_j) \wedge path(n_1, n_2) \rightarrow neighborOf(n_1, n_2)$

Une étude plus complète a été réalisée dans [MBPT10a, MBPT10b] qui a traité les points suivants :

- Définition formelle des différentes fonctions de transformations, et la fonction d'ordre.

- Définition de l'algorithme appelé $DREQ$ de construction et d'exécution des refor-

mulations selon la fonction d'ordre.

- Implémentation de l'agorithme en apportant d'autres optimisations comme vérifier l'existence de certains types de nœuds dans la base d'annotations. Ainsi les reformulations utilisant des types non référencés ne seront pas construites. Cette implémentation utilise Corese, un moteur de recherche sémantique qui permet d'interroger en SPARQL des annotations RDF.

- Réaliser une interface qui permet de visualiser l'arbre de génération des reformulations d'une requête utilisateur, les réponses sous forme de triplets RDF ou de documents surlignant le texte des nœuds réponses (Figures 4.3 et 4.4).

- Expérimenter sur deux corpus différents. En particulier sur notre corpus les résultats ont bien montré l'intérêt des métadonnées définies dans notre modèle d'annotation pour trier les réponses.

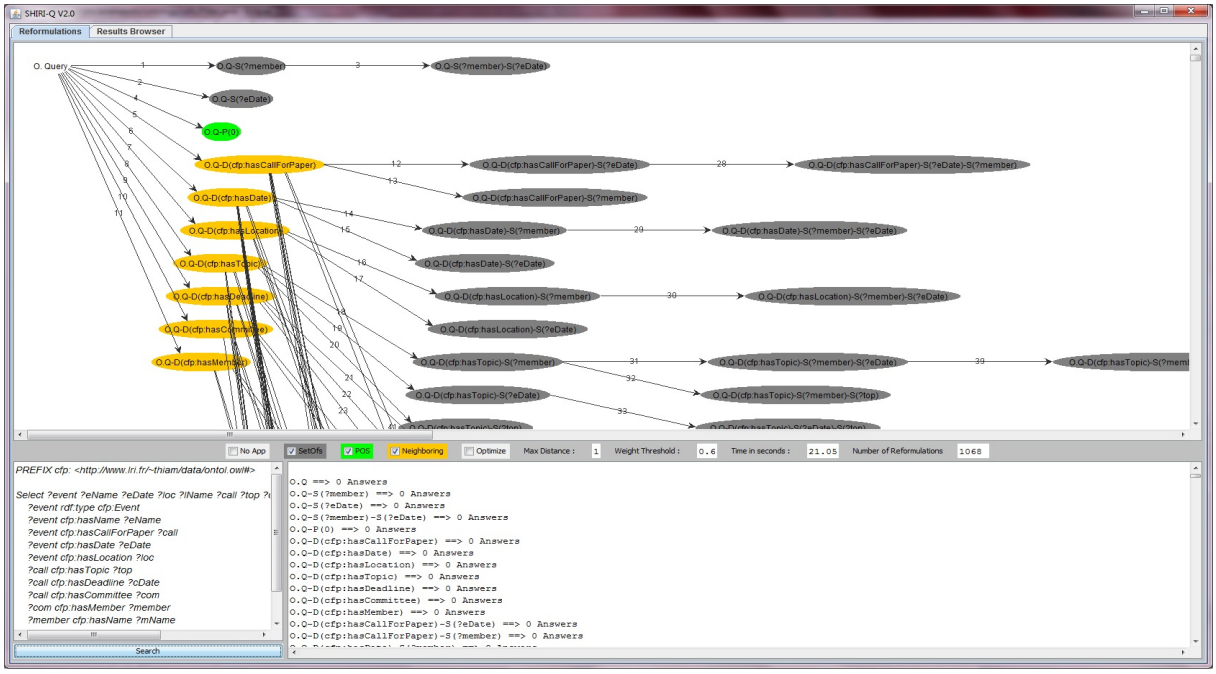


FIGURE 4.3 – Interface d'interrogation de SHIRI-Querying : reformulation des requêtes

4.3 Conclusion

Nous avons présenté les règles d'annotation qui génèrent les nœuds instances du modèle d'annotation. Ces règles exploitent la sémantique associée aux nœuds par *SHIRI-Extract* et leur proximité structurelle. Un nœud annoté est soit une instance d'un ou plusieurs concepts ontologiques, soit une instance de $PartOfSpeech$ ou de $SetOfc$. Ces nœuds

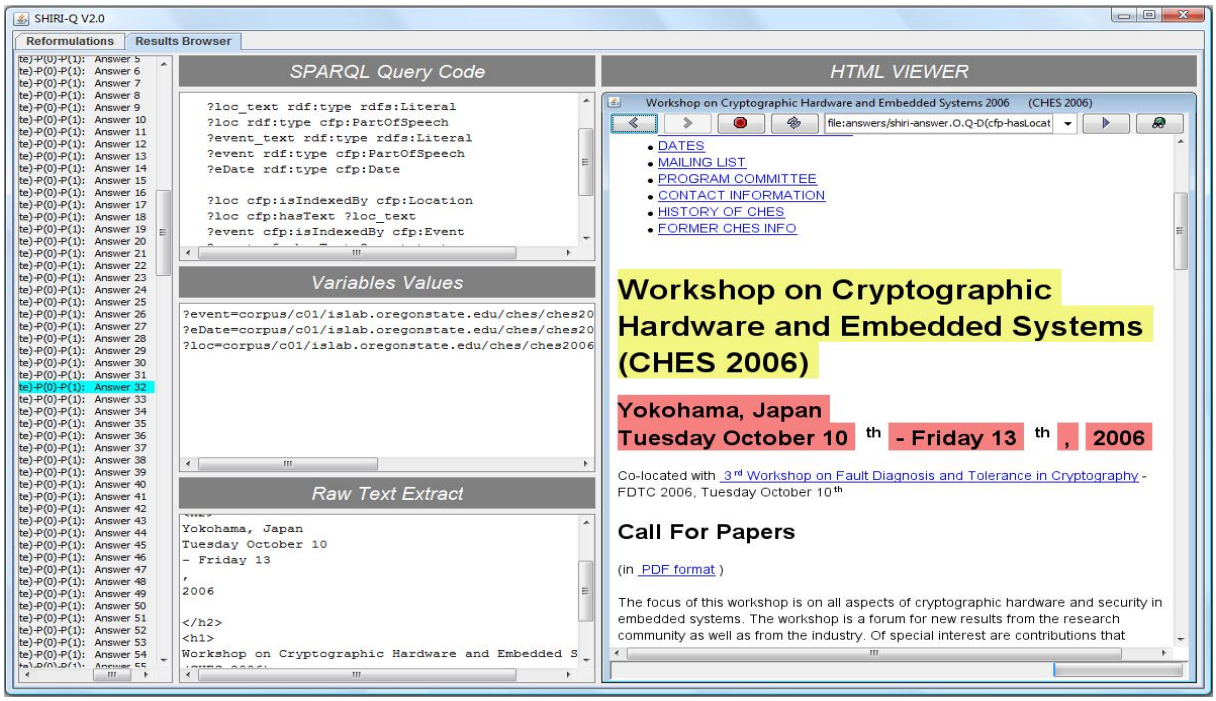


FIGURE 4.4 – Interface d'interrogation de SHIRI-Querying : navigation dans les résultats

peuvent être reliés par la relation $neighborOf$ avec une longueur maximale fixée du chemin les reliant dans l'arbre structurel représentant le document.

Nous avons présenté les fonctions de transformations qui permettent de construire les différentes reformulations à partir d'une requête utilisateur exprimée avec les concepts et les relations de l'ontologie. Ces fonctions exploitent les métadonnées du modèle d'annotation que nous avons défini afin d'atteindre et de trier tous les nœuds réponses.

Ces fonctions ont bien été formellement définies, implémentées et expérimentés sur notre base d'annotations dans un autre travail [MBPT10a]. Les résultats obtenus ont bien montré l'intérêt de notre modèle d'annotation pour trier les réponses selon le niveau d'agrégation des termes et entités nommées repérés dans les nœuds. De plus, la granularité de l'annotation permet de présenter à l'utilisateur les textes des nœuds réponses.

Chapitre 5

EXPÉRIMENTATIONS ET ÉVALUATION DES RÉSULTATS

Sommaire

Introduction

SHIRI-Extract permet d'extraire certains termes ou entités nommées d'une collection de documents semi-structurés et *SHIRI-Annot* permet d'annoter sémantiquement certaines parties de ces documents. La granularité de l'annotation est le nœud de l'arbre XML du document. Tout nœud de l'arbre XML du document contenant un terme ou une entité nommée alignée avec un concept de l'ontologie est annoté soit avec un concept, soit avec une métadonnée provenant de l'extension de l'ontologie. Ces métadonnées permettent à notre approche d'annotation de s'adapter aux degrés de structuration des documents.

L'approche est expérimentée sur un corpus de documents HTML provenant du Web. Ces documents décrivent des appels à participation à des conférences scientifiques en in-

formatique. Nous avons défini et utilisé une ontologie de domaine décrivant les principaux concepts apparaissant dans ces appels.

L'objectif principal de ce chapitre est de présenter l'expérimentation sur ces documents réels et d'évaluer les résultats des différentes phases de l'approche que sont l'extraction, l'alignement et l'annotation. Notre objectif n'est pas d'évaluer les patterns d'extraction mais de distinguer les erreurs dues à l'utilisation de ces patterns, de celles provenant de notre méthode d'alignement.

En ce qui concerne les phases d'extraction et d'alignement, l'évaluation consiste à :

- évaluer les mesures classiques de rappel et de précision ;
- étudier l'impact du choix des termes qui vont enrichir l'ontologie et permettre l'alignement en fonction des types de correspondance (subsomption, équivalence et proximité sémantique) obtenus lors de l'alignement ;
- évaluer le nombre d'alignements obtenus grâce aux appels Web par rapport aux alignements obtenus en utilisant uniquement la composante lexicale de l'ontologie. Nous étudions l'évolution de ce nombre au fur et à mesure du traitement des documents et donc de l'enrichissement de la composante lexicale.

Pour la phase d'annotation :

- nous évaluons le nombre de nœuds de type $SetOfConcept$, $PartOfSpeech$ et $Concept$;
- nous évaluons le nombre de fois où la relation $neighborOf$ est instanciée en faisant varier la distance maximale d entre nœuds de documents ;
- toujours, en faisant varier d, nous évaluons l'intérêt de la relation $neighborOf$ en calculant la proportion de relations sémantiques présentes entre les nœuds reliés par cette relation $neighborOf$ sur un échantillon.

Dans ce chapitre, nous présentons dans la section 5.1 l'ontologie du domaine des appels à participation à des conférences en informatique ainsi que le corpus de documents HTML. Dans la section 5.2 nous présentons les principes et mesures d'évaluation ainsi que les résultats de l'extraction puis de l'alignement. Enfin, dans la section 5.3, nous présentons les résultats de l'annotation.

5.1 Entrées des expérimentations

5.1.1 Ontologie du domaine

Pour expérimenter notre approche, nous avons défini une ontologie comportant les principaux concepts du domaine des appels à participation à des conférences en informa-

tique. L'ontologie comporte une composante lexicale que nous avons initialisée à l'aide de WordNet.

Ontologie du domaine des *appels à participation à des conférences*

L'ontologie $\mathcal{O}$ du domaine des *"appels à participation à des conférences en informatique"*, (*"call for papers"* en anglais, Figure 5.1), comporte des concepts, leurs propriétés et des relations sémantiques entre ces concepts. Nous nous intéressons aux évènements tels que les conférences, les workshops (concept $Event$), à leurs comités d'organisation ou de programme (concept *committee*), aux membres de ces comités (concept $Person$), à leurs équipes, laboratoires et universités (concept $Affiliation$), aux villes ou pays où ont lieu les évènements (concept $Location$), aux dates (concept $Date$) et aux thèmes des conférences (concept $Topic$).

Certains de ces concepts ont des instances qui peuvent être représentées par des entités nommées. Il s'agit des événements, des personnes, de leurs affiliations, des lieux et des dates.

Composante lexicale

L'ontologie comporte une composante lexicale faite de labels, de termes et d'entités nommées. Chaque concept est décrit par un label préféré qui désigne le concept auprès de l'utilisateur. Par exemple, *"person"* est le label préféré du concept $Person$ et *"call for paper"* est celui du concept $CallForPaper$. Tout concept de $\mathcal{O}$ dispose d'un ensemble de labels alternatifs, de termes et d'entités nommées. Si un concept n'a pas d'entités nommées, son ensemble $hasTermNe$ est vide.

Pour ces expérimentations, compte tenu du domaine choisi, nous avons pu jouer le rôle d'expert et nous avons initialisé la composante lexicale de l'ontologie en nous appuyant sur le thésaurus WordNet.

Pour initialiser les labels alternatifs d'un concept, nous avons sélectionné, dans WordNet, certains des synonymes ou des hyponymes du label préféré. Il s'agit de ceux qui sont équivalents au label préféré quand on se place dans le domaine des appels à participation. Nous les avons insérés dans l'ensemble des labels $\mathcal{L}$ et les avons reliés aux concepts par la propriété *altLabel*.

Pour les termes, nous avons choisi des hyponymes ou des termes appartenant à la liste des termes du domaine trouvés dans WordNet pour le concept. Ces termes sont insérés dans $\mathcal{T}$ et reliés aux concepts par la propriété *hasTerm*.

Par exemple, pour le concept $Topic$ nous avons sélectionné un ensemble d'environ 350 termes du domaine tels que *"Communications protocol", "data encryption", "information processing system", "object-oriented programming language"*, etc.

L'ensemble des labels et des termes sont présentés dans le tableau 5.1.

Nous avons choisi de ne pas utiliser de ressources lexicales pour initialiser les entités nommées de la composante lexicale : ce choix permet de mieux montrer l'apport de notre recherche utilisant le Web et de notre enrichissement. En effet, certaines entités nommées ne vont être alignées qu'en utilisant uniquement les termes extraits du Web.

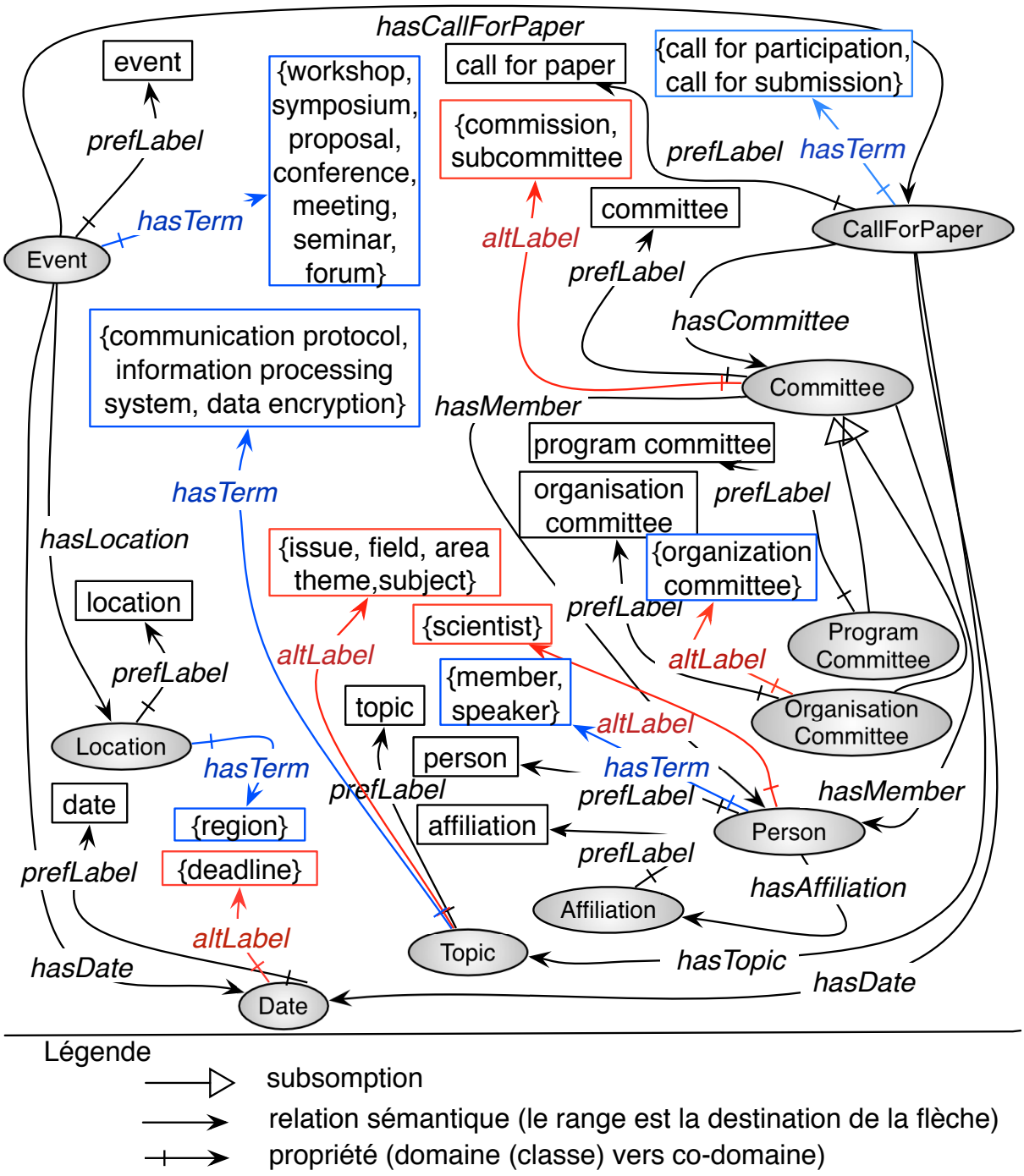


FIGURE 5.1 – Ontologie $\mathcal{O}$ des appels à participation à des conférences en informatique

Modèle d'annotation

Nous avons défini, dans le chapitre 2, le modèle d'annotation par un ensemble de métadonnées spécifiques pour le processus d'annotation *SHIRI-Annot*. Ces métadonnées expriment le degré plus ou moins fort d'agrégation des instances dans les nœuds de documents.

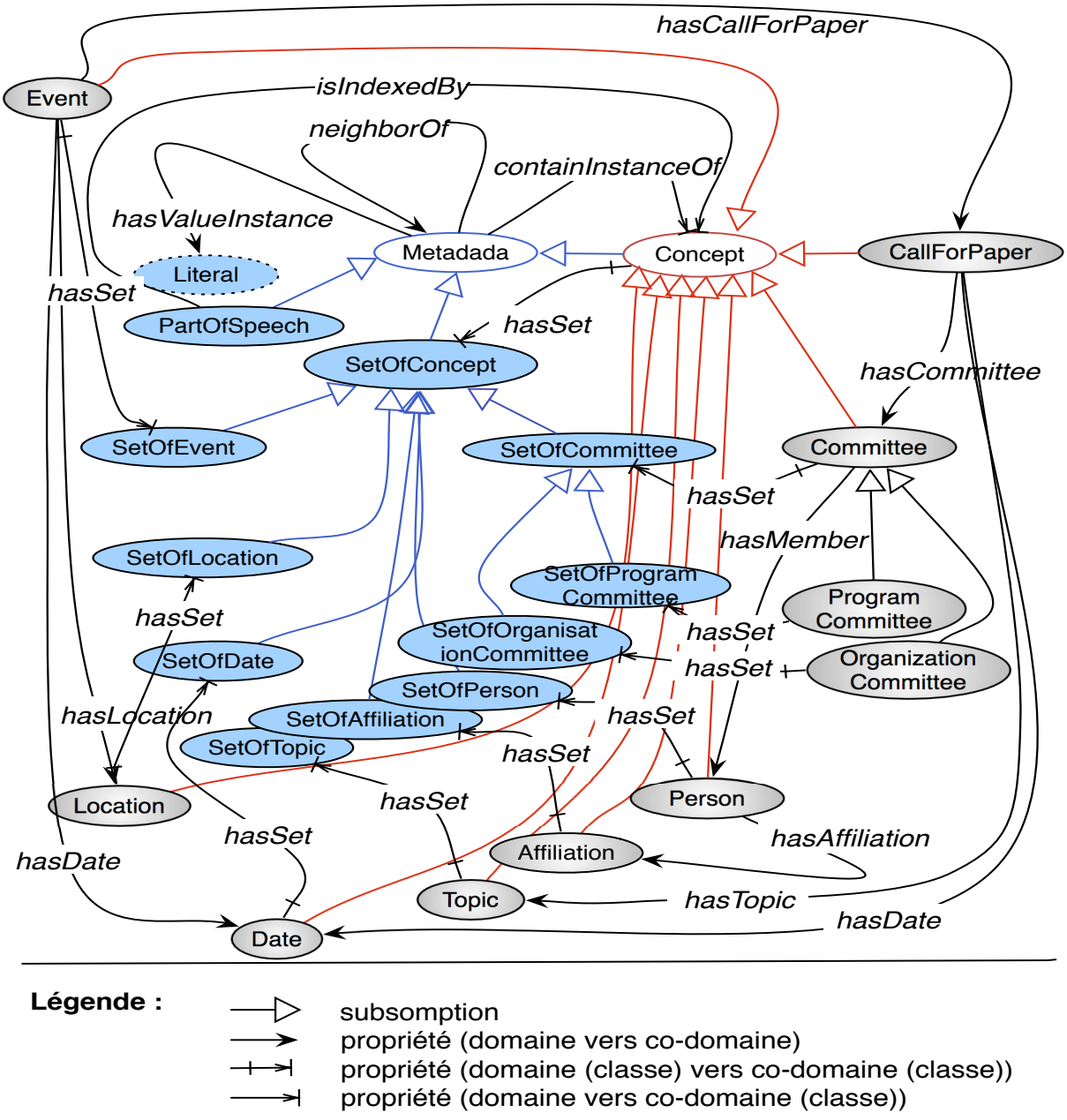


FIGURE 5.2 – Modèle d'annotation généré à partir de $\mathcal{O}$ sans la composante lexicale

Le modèle d'annotation généré à partir de l'ontologie est présenté dans la figure 5.2. Tous les concepts de l'ontologie sont des sous-concepts de *Concept*.

Le modèle permet de représenter les résultats de l'extraction. Un nœud *Nœud* est lié à un ou plusieurs littéraux par la propriété $hasValueInstance$. Un nœud est relié à un concept par la relation $containInstanceOf$ lorsque ces littéraux représentent des instances du concept.

Le modèle permet aussi de représenter les résultats de l'annotation. Les nouvelles métadonnées *PartOfSpeech*, *SetOfConcept*, *SetOfEvent*, *SetOfTopic*, *SetOfLocation*, *SetOfDate*, *SetOfAffiliation*, *SetOfPerson*, *SetOfCommittee*, *SetOfProgramCommittee* et *SetOfOrganizationCommittee* sont générées. La définition et l'utilité de ces métadonnées sont présentées dans le chapitre 2. Ces dernières et les sous-concepts de $Concept$ héritent de $Metadata$ qui représente l'ensemble des concepts du modèle permettant d'annoter des nœuds de documents.

5.1.2 Constitution du corpus

Le corpus est composé de documents HTML provenant du web, plus précisément de sites relatifs à des conférences scientifiques en informatique.

La constitution du corpus s'est fait en deux étapes : (i) chargement du corpus portant sur le domaine d'intérêt et (ii) pré-traitement du corpus.

Chargement

Le chargement a été réalisé en utilisant le moteur de recherche Google. Nous formons un ensemble de mots clé appelés graines (seed en anglais) à partir des labels de concepts et éventuellement des termes déjà associés aux concepts de l'ontologie. Ces mots sont combinés aléatoirement pour former des n-uplets qui sont fournis au moteur de recherche pour obtenir une liste d'URLs de documents à télécharger.

Le crawler que nous avons utilisé pour aspirer ces documents est l'outil *web get*[8]. Ce dernier prend en entrée la liste d'URLs de documents obtenue avec le moteur de recherche et les enregistre en local. L'outil nous permet de conserver la manière dont les pages constituant une conférence sont organisées.

Prétraitement

Les documents chargés sont pré-traités pour obtenir un ensemble de documents XHTML bien formés grâce à des outils comme JTidy et TidyHtml et des APIs comme SAX (Simple API for XML et DOM (Document Object Model).

Chaque document est ensuite représenté par un ensemble de mots numérotés par ordre d'occurrence dans le document. Les caractères de ponctuation sont bien entendu conservés pour la tâche d'extraction, et les balises remplacées par des points.

Nous effectuons un étiquetage morphosyntaxique des documents du $\mathcal{C}orpus$ avec l'outil QTag [TM98].

Après pré-traitement, le corpus est constitué de :

8. *wget* : Gestionnaire de téléchargement libre issu du projet GNU qui permet le téléchargement avancé de fichiers sur des réseaux et sur Internet, http ://jp.barralis.com/linux-man/man1/wget.1.php

- 412 fichiers non vides, contenant du texte, des images et d'autres types de contenus ;
- 202737 mots ;
- 1429616 octets.

Notre approche est basée sur l'hypothèse que les pages Web traitées sont du même domaine mais ne comportent pas forcément de régularités structurelles puisqu'elles proviennent de sites différents, construits par des auteurs indépendants et n'obéissant pas à des règles communes. Les documents du corpus que nous avons constitué sont effectivement structurés de manière hétérogène : nous retrouvons parfois des instances des mêmes concepts dans des nœuds bien structurés et parfois non structurés. Par exemple le tableau 5.2 montre des nœuds du corpus où les instances du concept *Topic* apparaissent parfois seules, parfois en liste, parfois dans un paragraphe avec d'autres instances. Dans le tableau, des exemples d'instances de *Topic* sont en bleu et des instances des autres concepts sont en rouge.

5.2 Expérimentation et évaluation de l'extraction et de l'alignement

5.2.1 Résultats de l'extraction

SHIRI-Extract permet d'extraire des termes et des entités nommées. Nous présentons les principes et les mesures d'évaluation de cette étape avant de présenter les résultats obtenus.

Principes et mesures d'évaluation

Certains termes ou entités nommées extraits sont corrects, d'autres comportent un mot en plus ou en moins et d'autres ne sont pas porteurs de sens. En effet, compte tenu de la nature générique des patterns utilisés et du degré d'automaticité de l'approche d'extraction, il n'est pas surprenant de rencontrer des termes ou entités nommées non porteurs de sens dans l'ensemble des instances extraites. Le repérage de certaines instances nécessiterait la mise en œuvre de techniques plus sophistiquées de TALN (Traitement Automatique de la Langue Naturelle). Nous sommes conscients de la difficulté consistant à décider si une entité nommée ou un terme est complet, comporte un mot en plus, est incomplet ou incorrect. Souvent ces choix ne peuvent être faits qu'après un examen du document d'origine.

En tenant compte des difficultés pré-citées, nous classons les termes et entités nommées extraits dans les catégories suivantes :

Faux termes ou fausses entités nommées

Concept (prefLabel)	**Labels alternatifs (altLabel)**	**Termes (hasTerm)**
Person	*scientist*	*member, speaker*
Topic	*issue, theme, subject, field, area*	*communication protocol, information processing system, data encryption (etc.)*
Event	∅	*workshop, symposium, proposal, conference, seminar, meeting, forum*
Committee	*commission, subcommittee*	∅
Affiliation	∅	∅
Date	*deadline*	∅
Location	∅	*region*
Call For Paper	∅	*call for participation, call for submission*
Program committee	∅	∅
Organisation committee	*organization committee*	∅

TABLE 5.1 – Labels alternatifs et termes des concepts de $\mathcal{O}$

Instances bien structurées	**Instances non structurées**
  - Data mining techniques : [clustering], classification, [association rules], [decision trees], etc. [Semantic web intelligence] [Analytics for social networks] 	[The 6th Annual European Semantic Web Conference (ESWC2009)] will present the latest results in research and [applications of Semantic Web technologies]. ESWC2009 will also feature a tutorial program, [system descriptions] and demos, a poster track, a PhD Symposium and a number of collocated workshops.

TABLE 5.2 – Exemples d'instances bien structurées et non structurées dans le corpus

Définition : Nous nommons *faux termes* ou *fausses entités nommées* les termes ou entités nommées que l'expert (donc nous) considère comme n'ayant pas de signification.

Ce caractère de *non sens* dépend aussi bien de la langue mais aussi du document contenant le terme ou l'entité nommée.

Exemples Les exemples de faux termes et entités nommées qui suivent proviennent de l'évaluation de *SHIRI-Extract.*

- Termes
  1. Name Email Institution ;
  2. aspect and Locations for rooftop ;
  3. results to scientists.
- Entités nommées
  1. Yes ;
  2. Photo Gallery.

Sur-extraction ou extraction partielle d'entités nommées

Définition : Certaines entités nommées sont incomplètes ou comportent un (des) mot(s) en plus.

- L'entité nommée extraite $t_j = w_j...w_m, k \preceq j \preceq m \preceq n$ est incomplète si celle qu'il fallait extraire est $t_k = w_k...w_n$. Nous parlons alors d'*extraction partielle*. Par exemple, le nom de la conférence est t_k =*International*$_k$ *conference*$_{k+1}$ *on*$_{k+2}$ *machine*$_{k+3}$ *learning*$_{k+4}$ alors que l'entité nommée extraite est t_{k+1} =*conference*$_{k+1}$ *on*$_{k+2}$ *machine*$_{k+3}$ *learning*$_{k+4}$.
- Lorsque l'entité nommée extraite $t_k = w_k...w_n, k \preceq j \preceq m \preceq n$ comporte un (des) mot(s) de trop et que la bonne entité nommée est $t_j = w_j...w_m$, nous parlons de *sur-extraction*. Par exemple, le nom de l'université est t_{k+1} =*Washington*$_{k+1}$ *University*$_{k+2}$ alors que l'entité nommée extraite est t_k =*About*$_k$ *Washington*$_{k+1}$ *University*$_{k+2}$.

Exemples Les exemples d'*extraction partielle* qui suivent proviennent de l'évaluation de *SHIRI-Extract.*

1. University of South (partielle) ;
2. International Conference on Field (partielle) ;

Mesures d'évaluation

Les mesures du rappel et de la précision sont calculées conformément à la catégorisation précédente des termes et entités nommées. Ces mesures sont connues et très utilisées dans des approches de traitement de l'information et de gestion de la connaissance. Considérons les notations suivantes :

- T_c est l'ensemble des termes ou des entités nommées extraits qui sont corrects ;

- T_f représente le nombre de faux termes ou de fausses entités nommées ;
- T_{sp} représente le nombre d'entités nommées extraites partiellement, ou sur-extraites ;
- T_o représente le nombre de termes ou le nombre d'entités nommées non trouvés.

Puisque la granularité de l'annotation est le nœud de l'arbre XML du document, nous présentons les résultats obtenus pour la précision en considérant : (a) que les entités nommées de T_{sp} sont incorrectes ($précision_{ssp}$) et (b) que les entités nommées de T_{sp} sont correctes ($précision_{asp}$) puisque le plus souvent le nœud contient le terme ou l'entité nommée en entier.

Précision

Définition : La précision est la proportion du nombre de termes ou entités nommées correctement extraits par rapport au nombre total de termes ou entités nommées extraits.
Pour les entités nommées :

$$précision_{ssp} = \frac{T_c}{T_c + T_f + T_{sp}} \tag{5.1}$$

$$précision_{asp} = \frac{T_c + T_{sp}}{T_c + T_f + T_{sp}} \tag{5.2}$$

Pour les termes :

$$précision = \frac{T_c}{T_c + T_f} \tag{5.3}$$

Rappel

Définition : Le rappel est la proportion du nombre de termes ou d'entités nommées correctement extraits par rapport au nombre de termes ou entités nommées qui auraient dû être extraits.

Son calcul nécessiterait de retrouver les entités nommées oubliées dans le texte des documents par *SHIRI-Extract*. Comme notre objectif n'est pas d'évaluer les patterns d'extraction mais de distinguer les erreurs provenant de ces patterns, de celles provenant de notre méthode d'alignement, nous n'avons pas cherché à calculer ce rappel.

Extraction des entités nommées et évaluation

L'ontologie du domaine comporte plusieurs concepts qui ont des instances de type entité nommée. Ce sont les concepts *date*, *personne*, *event*, *location* et *affiliation*.

Les entités nommées sont obtenues en appliquant sur le corpus étiqueté morpho-syntaxiquement des patterns utilisés dans *SHIRI-Extract* et présentés dans la section 3.1.2.

Le tableau 5.3 présente le nombre d'entités nommées extraites suivant leur taille avec ou sans répétitions. Il montre que dans un corpus de documents relatifs au même domaine, les entités nommées se répètent plusieurs fois : en moyenne ici plus de 10 fois. De là apparait bien l'intérêt de garder celles-ci dans l'ontologie lorsqu'elles sont alignées avec des concepts ou dans un ensemble conservant les échecs sinon. Ceci nous permet de diminuer le temps de traitement d'une entité nommée puisqu'une entité nommée n'est soumise qu'une seule fois au Web.

Longueur	Nombre d'entités nommées	Nombre d'entités nommées distinctes
1	23728	1860
2	11874	1420
3	4604	406
4	1053	129
5	217	20
TOTAL	41476	3835

TABLE 5.3 – Nombre d'entités nommées extraites

La validation des résultats de l'extraction sur 354 entités nommées, soit 114 entités nommées distinctes extraites d'un échantillon de documents pris au hasard est présentée dans le tableau 5.4. Ces entités nommées sont de longueur variable et 17% de celles-ci ne sont pas extraites avec précision (sur-extraction ou extraction partielle). La précision sans les entités nommées sur-extraites ou partielles $précision_{ssp}$ (74,56%) est beaucoup plus faible que la $précision_{asp}$ (91,23%).

Total entités nommées	Sans Répétition	Fausses	Sur-extraites ou partielles	$précision_{ssp}$	$précision_{asp}$
354	114	10	19	74,56%	91,23%

TABLE 5.4 – Evaluation de la précision de l'extraction d'entités nommées

Extraction des termes et évaluation

Le tableau 5.5 montre le nombre de termes extraits suivant leur taille avec ou sans répétitions. Ces termes sont obtenus après un étiquetage morpho-syntaxique des documents du corpus et l'application des patterns définis dans [Arp95].

L'évaluation de la précision de l'extraction des termes a porté sur 2300 termes distincts extraits de documents choisis au hasard. Nous avons 147 termes qui n'appartiennent pas au domaine comme *Fax or Regular Mail*, *audiovisual aids* et *Train hours*. Ces termes ne sont pas considérés dans l'évaluation. Donc la précision a été calculée sur 2153 termes. Les résultats de l'extraction des termes sont présentés dans le tableau 5.6. La majorité des termes extraits sont porteurs de sens mais les nombreux *faux termes* s'expliquent par la présence d'erreurs typographiques comme l'absence d'un blanc. C'est le cas de nombreux

Longueur	Nombre de termes	Nombre de termes distincts
1	101430	14413
2	32712	15806
3	17912	10020
4	5704	3797
5	966	602
7	104	48
TOTAL	158828	44680

TABLE 5.5 – Nombre de termes extraits

exemples comme *onsemistructured data in order* et *ofmachine learning*. Dans d'autres cas, le pattern syntaxique a extrait une suite de mots n'ayant pas de sens comme *reviewers to make* et *large populations of self*.

Total termes	Faux termes	Précision
2153	968	55,04%

TABLE 5.6 – Evaluation de la précision de l'extraction des termes

5.2.2 Outil de Pierre Senellart

Nous avons utilisé un outil externe permettant d'extraire des noms de personnes et des dates, développé par Pierre Senellart dans le cadre d'une approche qui permet de découvrir et d'analyser les données d'un domaine d'intérêt accessibles via des formulaires [P.07].

Les dates sont extraites avec des expressions régulières classiques. Les noms et les prénoms des personnes sont extraits de DBLP [9]. Environ 210.000 noms de famille et 85.000 prénoms sont extraits. Ensuite, l'outil utilise des expressions régulières qui définissent les combinaisons possibles de noms et de prénoms. Par exemple à partir du nom *Smith* et des prénoms *John* et *Jacques*, les expressions régulières permettent d'obtenir les combinaisons suivantes : *Smith*, *Smith, John*, *John Smith*, *Smith John* et *Smith John Jack*.

Les probabilités associées aux expressions régulières sont définies de manière ad hoc bien que Pierre Senellart indique qu'elles pourraient provenir d'études statistiques sur de larges corpus [PAD+08].

Nous avons extrait 4334 noms de personnes et 2190 dates. La validation des résultats obtenus sur un échantillon de 100 documents pris au hasard (parmi les 691 documents du corpus), soit 1201 entités nommées est présentée dans le tableau 5.7. Comme il s'agit d'un outil qui extrait et aligne simultanément, le rappel et la précision dans le tableau sont calculés par rapport au nombre de personnes et de dates bien alignées trouvées.

9. DBLP : Digital Bibliography and Library Project, http ://dblp.uni-trier.de/

	Trouvées	Correctes	Oubliées	rappel	précision
Personnes	745	690	97	87.67%	92.62%
Dates	456	445	144	75.55%	97.59%
Total	1201	1135	241	82.49%	94.5%

TABLE 5.7 – Evaluation des résultats de l'outil de Pierre Senellart

5.2.3 Résultats de l'alignement

Après l'extraction des entités nommées et des termes, SHIRI-Extract permet de les aligner avec des concepts de l'ontologie. Nous présentons les principes et les mesures d'évaluation de cette étape d'alignement avant de présenter les résultats obtenus.

Principes et mesures d'évaluation

Nous tentons d'aligner les entités nommées et les termes résultant de la phase d'extraction avec des concepts de l'ontologie. Nous commençons par un alignement local basé sur l'égalité pour les entités nommées et sur Taxomap [HZSR08] pour les termes. Si ce dernier échoue nous cherchons des labels sur le Web pour le terme ou l'entité nommée et tentons de l'aligner à nouveau.

Afin de distinguer les erreurs introduites par la phase d'extraction de celles introduites par la phase d'alignement, nous avons ignoré les faux termes ou fausses entités nommées dans lévaluation des résultats de l'alignement. Ainsi les alignements de *results to scientists* et de *Fair Student Tavel* ne sont pas comptabilisés.

En tenant compte de la catégorisation des termes ou des entité nommées (corrects, sur-extraits ou partiels), nous classons les résultats de l'alignement dans les catégories suivantes.

Alignements corrects

Définition : Un alignement est *correct* si le terme ou l'entité nommée extrait est correct, partiel ou sur-extrait et est associé au(x) bon(s) concept(s). Les alignements des faux termes ou des fausses entités nommées sont ignorés.

Exemples

1. University of South (alignement correct car l'entité nommée est partielle mais annotée avec le bon concept *Affiliation*) ;
2. privacy and safety issues (alignement correct car le terme est correct et annoté avec le bon concept *Topic*).

Faux alignements

Définition : Un alignement est *faux* si l'association d'un terme correct ou d'une entité nommée correcte ou partielle ou sur-extraite avec un mauvais concept.

Exemples

1. Conferences themes (alignement faux car le terme est correct mais annoté avec un mauvais concept *Event*) ;

2. meeting room (alignement faux car le terme est correct mais annoté avec un mauvais concept *Event*) ;
3. Regions in spatial data (alignement faux car le terme est correct mais annoté avec un mauvais concept *Location*).

Pour évaluer le rappel et la précision, nous utilisons les notations suivantes :

- A_c représente le nombre d'alignements corrects ; nous distinguons A_{sp} inclus dans A_c et qui représente le nombre d'entités nommées partielles ou sur-extraites alignées correctement ;
- A_f représente le nombre de *faux alignements* ;
- A_o représente le nombre de termes ou entités nommées extraits non alignés et qui auraient dû être alignés avec des concepts de l'ontologie.

Précision

Définition : La précision est la proportion du nombre de termes ou d'entités nommées correctement alignés par rapport au nombre total de termes ou d'entités nommées alignés.
Pour les entités nommées :

$$précision_{ssp} = \frac{A_c - A_{sp}}{A_c + A_f} \tag{5.4}$$

$$précision_{asp} = \frac{A_c}{A_c + A_f} \tag{5.5}$$

Pour les termes :

$$précision = \frac{A_c}{A_c + A_f} \tag{5.6}$$

Rappel

Définition : Le rappel est la proportion du nombre de termes ou du nombre d'entités nommées correctement alignés par rapport au nombre total de termes ou des entités nommées qui auraient dû être alignés.

$$rappel = \frac{A_c}{A_c + A_o} \tag{5.7}$$

Alignement des entités nommées

Nous alignons avec des concepts de $\mathcal{O}$ l'ensemble des entités nommées obtenues avec les patterns syntaxiques définis dans 3.1.2. Pour rapprocher ces entités nommées des concepts

de l'ontologie, nous les comparons, pour ces expérimentations, en utilisant l'égalité (=) stricte, avec les entités nommées présentes dans les ensembles *hasTermNe* des concepts de $\mathcal{O}$.

Si cette tentative d'alignement échoue, nous cherchons des labels de l'entité nommée sur le Web et tentons de nouveau d'aligner les labels obtenus avec les labels et les termes des concepts présents dans $\mathcal{O}$ et dont les instances sont des entités nommées.

		Affiliation	Location	Person	Event	Date	Total
Alignements Trouvés	Avec $\mathcal{O}$	0	0	745	0	456	1201(25.46%)
	Avec Web	1317	1097	362	741	0	3517(74.54%)
	Total	1317	1097	1107	741	456	4718

TABLE 5.8 – Nombre d'alignements obtenu en local ou via le Web pour les entités nommées

Après l'alignement, nous obtenons les résultats présentés dans les tableaux 5.8 et 5.9. Ces résultats incluent des entités nommées obtenues en utilisant l'outil spécialisé de Senellart [P.07]. Il s'agit des noms de personnes et les dates qui sont directement alignés avec les concepts *Person* et *Date* de $\mathcal{O}$, voir section 5.2.2. L'évaluation de la phase d'alignement a porté sur 4718 entités nommées distinctes qui sont soit correctes, soit partielles ou sur-extraites. Le tableau 5.8 montre le nombre d'entités nommées alignées directement avec l'ontologie et le nombre de celles alignées en utilisant des labels provenant du Web : 74.54% des entités nommées ont été alignées grâce à des labels provenant du Web. Toutes les affiliations, les événements et les lieux ont été retrouvés grâce au Web. En effet l'ensemble des entités nommées de l'ontologie est vide au départ. Les dates et les personnes qui sont alignées directement avec l'ontologie le sont grâce à l'outil de Pierre Senellart.

Le tableau 5.9 montre la précision de l'alignement des entités nommées.

Concepts	Alignements trouvés			A_o	$Précision_{ssp}$	$Précision_{asp}$	Rappel
	A_c-A_{sp}	A_{sp}	A_f				
Affiliation	1109	25	183	384	84.18%	86.07%	74.69%
Location	1081	5	11	89	98.53%	99.02%	92.40%
Person	990	15	101	227	89.47%	90.85%	81.59%
Event	477	161	103	115	64.35%	86.13%	84.72%
Date	445	0	11	118	97.58%	97.58%	79.07%
Total/Moyenne	4096	207	409	933	86.93%	91.32%	82.18%

TABLE 5.9 – Précision de l'alignement des entités nommées

Le tableau 5.9 montre qu'en moyenne la précision est de 91.32%. Si nous n'avions pas pris en compte les entités nommées partielles ou sur-extraites alignées correctement, la précision n'aurait été que de 86.93%. En effet, les événements (concept *Event*) sont souvent partiellement extraits en raison de leur longueur et de leur complexité. Par exemple, l'entité nommée *"International Conference"* est incomplète, mais l'unité structurelle la

Concepts	Alignements trouvés	Corrects (A_c)	Faux (A_f)	Oubliés (A_o)	Précision	Rappel
Affiliation	165	160	5	9	96.97%	94.67%
Location	143	115	28	3	80.42%	97.46%
Person	206	131	75	16	63.59%	89.12%
Event	80	52	28	0	65.00%	100.00%
Topic	276	181	95	29	65.58%	86.19%
Total	870	639	231	57	73.45%	91.81%

TABLE 5.10 – Précision et rappel de l'alignement des termes

contenant contient le nom complet qui est "*International Conference on Web Services*". Donc si *"International Conference"* est alignée avec *Event*, nous considérons comme correct cet alignement. Pour ce concept, la $précision_{asp}$ est de 86.13% alors que la $précision_{ssp}$ n'est que de 64.35%.

La précision de l'alignement des lieux (concept *Location*) n'a pas beaucoup varié quand on prend en compte ceux partiellement extraites ou sur-extraites parce qu'ils sont décrits par des entités nommées courtes et non complexes. Ils sont facilement alignés en utilisant des labels du Web car ils sont fréquemment cités sur les pages Web : le rappel est de 92.4%.

Les dates (concept *Date*) sont obtenues avec des expressions régulières qui sont appropriées aux formes des dates qui apparaissent dans le corpus. C'est la raison pour laquelle elles sont toutes complètes.

Certaines entités nommées, bien que souvent décrites avec des expressions complexes, ont été également trouvées avec une bonne précision et un bon rappel. C'est le cas des affiliations (concept *Affiliation*) et des personnes (concept *Person*) qui comportent souvent des mots étrangers à la langue.

Alignement des termes

Après l'alignement, nous avons évalué les résultats sur 927 termes provenant de 30 documents que nous avons choisis au hasard. Parmi ces termes, 870 ont été alignés avec des concepts de l'ontologie. La précision de l'alignement est de 73.45% et le rappel est de 91.81%. Le nombre de faux alignements est 231 et le nombre d'alignements oubliés est de 57. Les résultats sont présentés dans le tableau 5.10.

Globalement la précision de l'alignement des entités nommées (91.93%) est meilleure que celle des termes (74.31%) car les termes varient plus syntaxiquement que les entités nommées. Ces erreurs de l'alignement des termes s'expliquent par l'automaticité de la méthode d'une part et d'autre part par les manquements des techniques d'alignement. Ces techniques doivent tenir compte de la langue du document et plus précisément du rôle des mots dans le groupe nominal. Par exemple *"Conferences themes"* est aligné avec le concept *Event* car il contient le mot *conference* qui est un terme de *Event*. De la même manière, *"Regions in spatial data"* est annoté avec le concept *Location* à cause de la présence de son terme *region* dans l'instance.

Historisation des entités nommées et termes non alignés

Un terme ou une entité nommée qui n'a pas pu être aligné avec un concept de l'ontologie est stocké avec les labels candidats trouvés sur le Web. Ces échecs sont gardés dans un fichier XML dont nous présentons un extrait ci-dessous. Le contenu de ce fichier est chargé à chaque appel au programme dans le but de ne pas rechercher un label inutilement sur le Web.

```
<term id="Software Technology">
        <label name="market leader"/>
        <label name="archival journal"/>
</term>
<term id="Dining Service">
        <label name="food service provider"/>
</term>
<term id="International Money Orders">
        <label name="alternative"/>
</term>
        <term id="Process Integration">
        <label name="process"/>
</term>
<term id="Systems Society">
        <label name="association"/>
</term>
```

Le nombre d'échecs après annotation de 23300 termes provenant du corpus est de 8284.

Enrichissement de l'ontologie

Notre approche enrichit l'ensemble des termes et des entités nommées associés aux concepts. Le nombre de nouveaux termes et de nouvelles entités nommées obtenus après annotation de tout le corpus est montré dans la table 5.11. Par exemple le concept *Topic* qui disposait de 352 termes au départ a été enrichi de 333 termes.

Courbes d'évolution d'alignements locaux et des appels Web

Le comportement incrémental de l'algorithme *Extract-Align* doit permettre que le nombre de recours au Web diminue au fur et à mesure que le nombre de termes traités augmente (i) plus l'ontologie est peuplée par de nouveaux termes, plus un candidat terme est susceptible d'être aligné directement et (ii) tous les termes dont les tentatives d'alignement via le web échouent sont stockés avec les labels retournés par le Web et ne seront plus soumis au Web, même si les prochaines tentatives d'alignement en local échouent. Nous supposons que le Web n'a pas subi entre temps des modifications permettant de retrouver d'autres labels.

	Nombre de Termes		Nombre d'entités nommées	
Concept	avant annotation	après annotation	avant annotation	après annotation
Topic	352	685	0	0
Event	7	704	0	638
Person	2	212	0	3419
Date	0	48	0	445
Location	1	74	0	1086
Affiliation	0	551	0	1134
Committee	0	123	0	0
Program Committee	0	1	0	0
Organisation Committee	0	0	0	0
Call For Paper	2	2	0	0

TABLE 5.11 – Enrichissement de la composante lexicale de l'ontologie après un traitement de 66 fichiers, soit 233000 termes

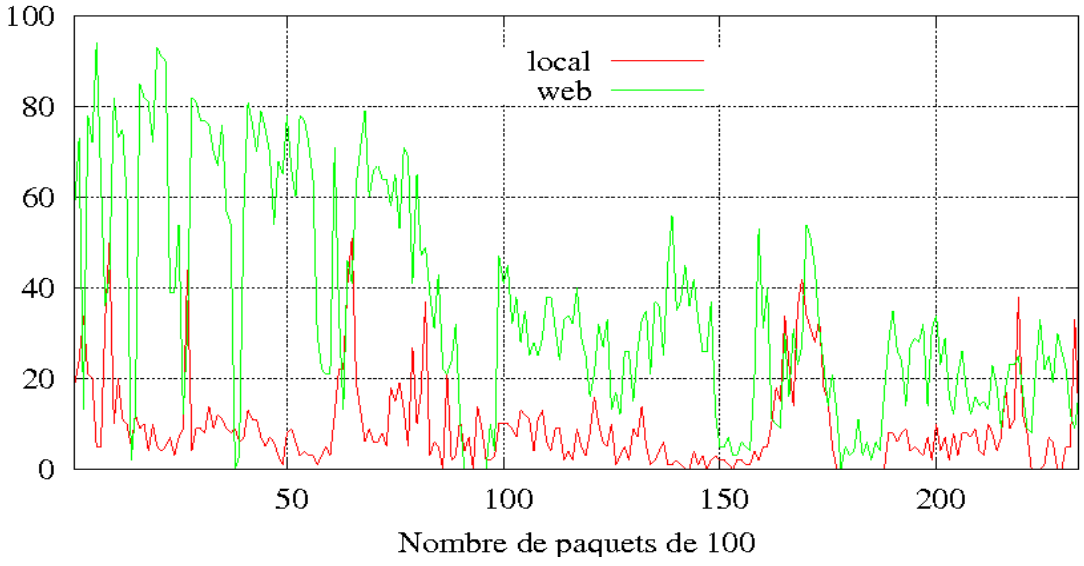


FIGURE 5.3 – Nombre d'alignements locaux et d'appels Web en fonction du nombre de paquets traités

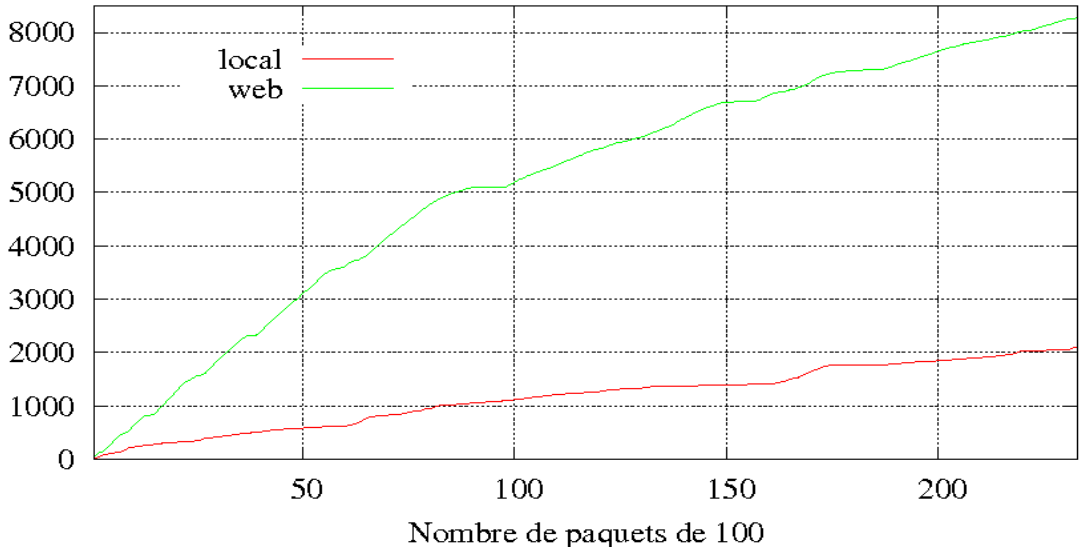


FIGURE 5.4 – Nombre cumulé d'alignements locaux et d'appels Web en fonction du nombre de paquets traités

Le recours au Web nécessite beaucoup de ressources machine, surtout en termes de temps. Pour minimiser le coût des appels Web, l'ensemble des termes extraits est découpé en paquets de 100 termes (paramètre *TaillePaquet* du tableau 3.3). Nous ne faisons appel au Web qu'après avoir atteint la taille spécifiée dans ce paramètre.

Les courbes des figures 5.3 et 5.4 montrent : (i) l'évolution du nombre de termes alignés en local par rapport au nombre de de termes traités et (ii) l'évolution du nombre d'appels Web par rapport au nombre de termes traités. Ces résultats montrent que le nombre d'invocations du Web diminue avec le nombre de termes traités. En effet, la courbe 5.4 montre que pour 5000 termes, le nombre d'appels Web est d'environ 3100 (62%), pour 10000 termes, ce nombre est d'environ 5200 (52%) et pour 200000 terme, il est d'environ 7700 (38.5%).

La courbe 5.3 montre que le pourcentage de termes et d'entités nommées alignés localement reste assez régulier : environ 10%. Pour ces expérimentations la diminution des appels Web est essentiellement due au stockage des échecs.

Le tableau 5.5 montre que beaucoup de termes se répètent plusieurs fois. Ceci est dû au fait que les documents sont du même domaine. Cependant, comme l'algorithme (cf. Tableau 3.3) traite les termes du plus long au plus court, la plupart d'entre eux ne sont pas traités car ils sont inclus ou équivalents à des termes déjà traités.

Impact du choix de types de correspondances sur le processus d'alignement

Dans le processus d'alignement, lorsque nous tentons de rapprocher les instances extraites avec les labels et les termes des concepts de l'ontologie, nous utilisons l'outil d'alignement Taxomap [HZSR08]. Cet outil découvre trois types de correspondances entre

une instance donnée et les labels et/ou termes des concepts de l'ontologie. Ces correspondances, présentées dans la section 3.1.4 du chapitre 3 sont :

- l'équivalence ($isEq$ ou $=$) ;
- la subsomption (isA) ;
- la proximité sémantique ($isClose$).

Notre algorithme étant incrémental, les alignements à l'itération k influent sur les alignements aux itértions $(k+i)_{i>0}$ suivantes parce que les instances alignées à l'itération k enrichissent l'ontologie.

Tous les résultats présentés dans les sections précédentes sont obtenus en considérant toutes les instances alignées à l'itération k dans l'enrichissement de l'ontologie quelque soit le type de correspondances utilisé lors de l'alignement, mais seules celles obtenues avec les relations $isEq$ et isA sont utilisées dans les tentatives d'alignement suivantes.

Nous avons également expérimenté l'utilisation de toutes les instances alignées à l'itération k pour enrichir l'ontologie et pour rechercher des correspondances dans les itérations $(k+i)_{i>0}$ suivantes pour un échantillon donné de 3500 termes. Les résultats présentés dans les figures 5.5 et 5.6 montrent que (i) la courbe des appels Web décroit alors rapidement et que (ii) celle des alignements obtenus en local croît considérablement. Cependant la précision est alors très faible, de l'ordre de 30%.

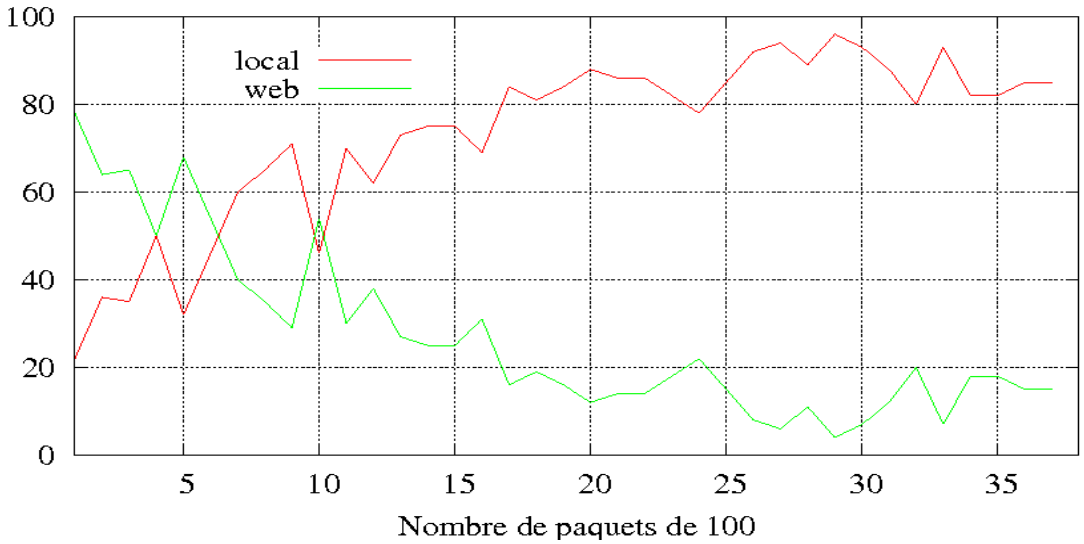


FIGURE 5.5 – Courbe des appels locaux et Web utilisant les relations *isA*, *isEq* et *isClose*

En résumé, nous notons que le nombre d'appels au Web décroît plus lentement quand nous n'utilisons que les correspondances isA et $isEq$ comme le montre les courbes de la

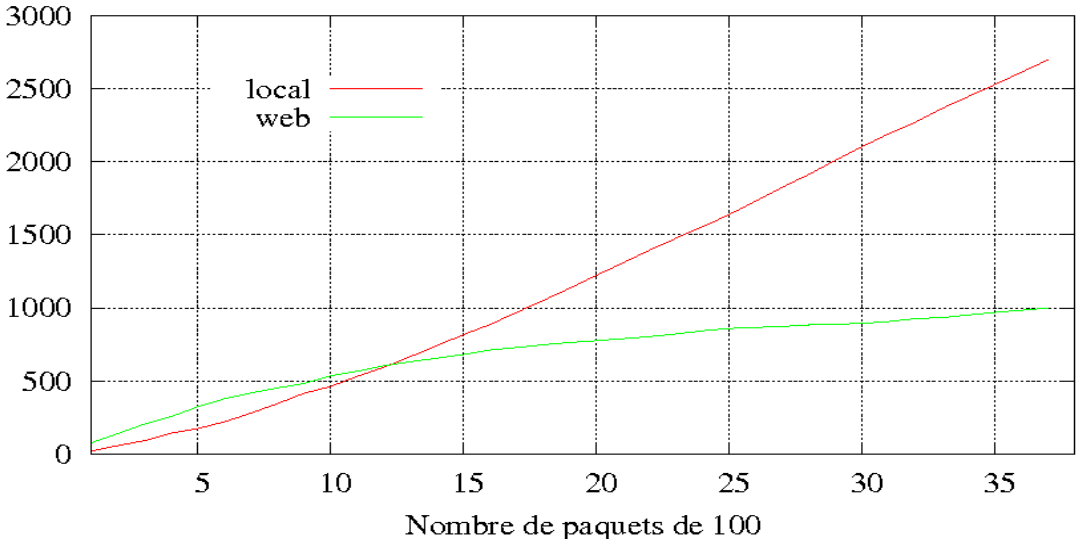


FIGURE 5.6 – Courbe des appels locaux et Web cumulés utilisant les relations *isA*, *isEq* et *isClose*

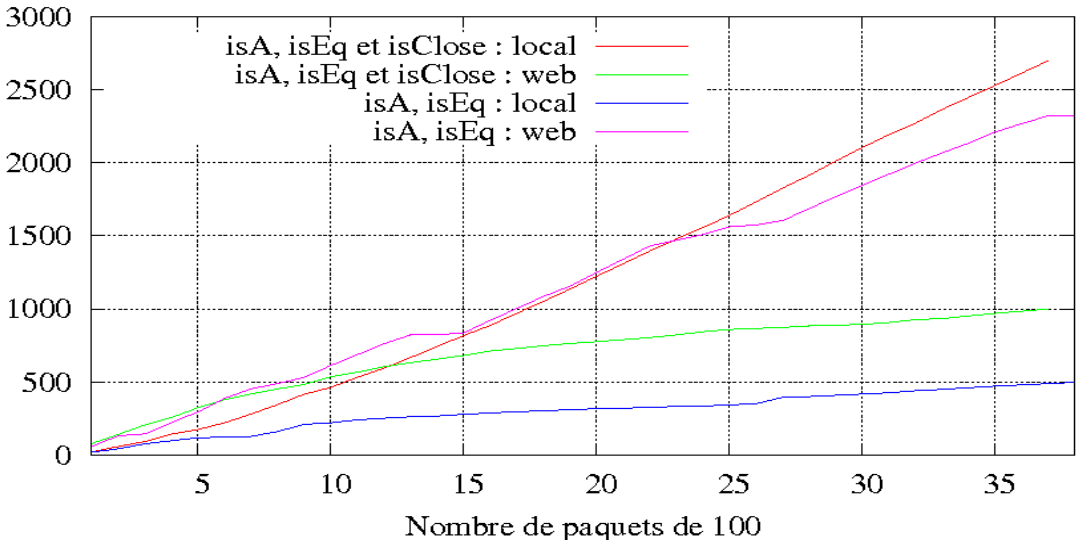


FIGURE 5.7 – Comparaison des courbes des appels locaux et Web

figure 5.7, mais les résultats de l'alignement ont alors une bien meilleure précision. Il s'agit des résultats que nous avons présentés dans les tableaux 5.10 et 5.9.

5.3 Expérimentation et évaluation de l'annotation

Les systèmes d'interrogation basés sur les ontologies sont fréquemment utilisés dans l'extraction d'information. Ils améliorent la recherche basée sur les mots clé en une recherche basée sur les ontologies. Plusieurs approches approximent sémantiquement la requête de l'utilisateur. CORESE [CDKFZG05] est un de ces outils qui proposent différentes approximations utilisant soit la distance ontologique basée la relation de subsomption, soit une proximité contextuelle, soit la longueur du chemin les séparant les concepts dans l'ontologie. CORESE propose un langage de règles permettant de faire des inférences sur les connaissances. Pour générer les annotations finales, nous posons des règles définies dans le langage de règle de CORESE sur les triplets de pré-annotation.

5.3.1 Résultats du typage des nœuds

Sur un échantillon formé de 30 documents du corpus, nous avons annoté 2225 nœuds : 1525 (68.54%) sont typés avec des concepts de l'ontologie, 165 (7.42%) sont typés par des sous-concepts de *SetOfConcept* et 501 (24.04%) sont typés par *PartOfSpeech*, indexés avec des concepts de l'ontologie (5.12).

Le grand nombre de nœuds annotés par des concepts s'explique par le fait que le langage HTML dispose de balises (*ul*, *pl*, *li*) permettant une représentation structurée des listes et que ces listes sont largement utilisées dans des documents tels que les appels à participation à des conférences scientifiques. De plus, certains nœuds sont typés par des concepts bien que contenant probablement d'autres instances de concepts que *SHIRI-Extract* n'a pas détecté. Nous avons obtenu un nombre assez élevé de nœuds de type $PartOfSpeech$ parce que ces documents comportent également des nœuds peu structurés pour lesquels leurs concepteurs se soucient peu de la séparation des informations délivrées selon les concepts décrivant le domaine. Le nombre faible de nœuds de type $SetOfConcept$ est inhérent à sa définition même. L'ontologie ne comporte pas beaucoup de concepts comparables et il est plus rare de trouver des listes non structurées dans les documents.

	Concepts	SetOfConcept	PartOfSpeech
Nombre	1525	165	535

TABLE 5.12 – Typage des nœuds

5.3.2 Évaluation de la relation *neighborOf* selon la distance

Le voisinage des nœuds dans un document peut être source de relation entre les instances de concepts localisées dans ces nœuds. Par exemple une personne et son affiliation se trouveront probablement dans le même nœud ($d = 0$) ou dans des nœuds distincts mais

séparés par une distance d donnée inférieure à un seuil. Dans ce dernier cas, la relation *neighborOf* est générée entre les nœuds.

Nous donnons ci-dessous un exemple de nœud comportant une personne et son affiliation. Les deux instances sont reliées avec la relation $hasAffiliation$. Le nœud est annoté avec la métadonnée *PartOfSpeech* (notée $d = 0$).

< *span* >
Communication Technology and Protocols
Duc A. Tran, University of Dayton, USA
< /*span* >

L'évaluation de cette phase consiste à vérifier si les instances représentées par les termes ou entités nommées extraits du même noeud ($partOfSpeech$) ou de nœuds voisins sont liés sémantiquement. Le tableau 5.13 montre que la meilleure précison est obtenue pour $d = 1$. Le nombre de relations trouvées croît rapidement avec la distance et plus la distance est grande, moins la précision est bonne. Ces résultats vérifient bien avec notre hypothèse de départ qui consiste à dire que les instances qui se trouvent soit dans le même nœud, soit dans des nœuds voisins ont de bonnes chances d'être sémantiquement liées.

Distance	0	1	2	3	4
Nombre	368	178	378	123	916
Nombre Cumulés	368	546	924	1047	1963
Validés	100	200	300	400	500
Corrects	80	180	207	211	213
Précision	80.00%	90.00%	69.00%	52.75	42.60%

TABLE 5.13 – Évaluation du *neighborOf* suivant la distance

Bien sûr, nous n'avons pas essayé de calculer le rappel mais même si seulement une partie des relations sémantiques est détectée par cette heuristique, les relations trouvées en fixant une faible longueur de chemin sont souvent justes.

Conclusion

Nous avons expérimenté notre approche sur un corpus de 412 documents HTML provenant du Web. Ces documents décrivent des appels à participation à des conférences scientifiques en informatique. L'ontologie comporte une composante lexicale où les labels des concepts et les termes ont été sélectionnés manuellement à l'aide de WordNet.

Nous avons évalué séparément les résultats obtenus par les patterns d'extraction, de ceux obtenus par notre méthode d'alignement lors de l'exécution de l'algorithme *Extract-Align*. Cette séparation nous a permis de mieux évaluer les erreurs introduites par la méthode d'alignement.

Les patterns d'extraction ne délimitent pas parfaitement les termes et les entités nommées susceptibles de représenter des instances de concepts de notre ontologie. Cependant, même s'ils sont extraits de manière partielle ou sont sur-extraits, le nœud du document contient le plus souvent le terme ou l'entité nommée recherché. Les résultats de l'extraction montrent que la précision calculée au niveau du nœud est nettement meilleure que celle calculée au niveau des termes et des entités nommées extraits.

L'alignement d'un terme exploite l'outil d'alignement de taxonomies Taxomap. Les résultats que nous obtenons sont des correspondances d' équivalence ($isEq$) ou de subsomption (isA) ou de proximité ($isClose$). Ces correspondances sont exploitées pour la pré-annotation et pour l'enrichissement de la composante lexicale. Au moment de l'alignement, nous avons choisi de n'utiliser dans cette composante lexicale que les termes déjà validés par l'expert, auxquels s'ajoutent, ceux obtenus uniquement par $isEq$ ou de subsomption isA . Nous avons montré que la précision de l'alignement est mauvaise si nous utilisons également les termes obtenus par les correspondances $isClose$.

Nous avons pu montrer que l'utilisation du web permettait de découvrir des alignements qui n'auraient pas été possibles en utilisant un alignement direct avec les élements de la composante lexicale pour les entités nommées (74,54 % sont trouvées via le Web). Pour les termes, nous avons constaté que le web ne permet d'en aligner que très peu. L'alignement via le web est une technique coûteuse mais, dans un tel corpus, les termes et les entités nommées se répètent (en moyenne 10 fois pour une EN, 3 fois pour un terme). Les résultats ont montré que l'enrichissement de l'ontologie avec les termes et les entités nommées allié à l'historisation des tentatives d'alignement qui ont conduit à un échec permet de réduire l'utilisation du Web au fur et à mesure que des documents sont annotés.

Après application des règles d'annotation, nous obtenons différents types de nœuds ($Concept$, $SetOfConcept$, $partOfSpeech$). Les termes et entités nommées sémantiquement liés se trouvent souvent soit dans le même nœud, soit dans des nœuds voisins. Nous avons pu montrer, qu'en l'absence de relations sémantiques annotées, il était intéressant de représenter la présence possible d'une relation sémantique en utilisant la relation $neighborOf$ car ces liens sont souvent justes lorsque la distance d paramétrée est faible.

CONCLUSIONS ET PERSPECTIVES

L'annotation sémantique de documents permet d'associer à des éléments de document des concepts, des instances de concepts, ou des instances de relations qui sont décrits formellement dans une ontologie. L'objectif est de permettre de formuler des requêtes basées sur le vocabulaire de cette ontologie, de raisonner sur les annotations sémantiques, et de recueillir des réponses combinant des données provenant de différents documents.

Nous nous sommes intéressés à l'annotation sémantique de documents HTML relatifs à un domaine d'intérêt. Ce domaine d'interêt est décrit dans une ontologie qui comporte une composante lexicale. Cette composante lexicale associe à chaque concept un ensemble de labels équivalents le représentant, ainsi qu'un ensemble de termes ou d'entités nommées (EN). Les documents peuvent avoir été collectés manuellement ou par un robot d'indexation utilisant les labels ou les termes associés aux concepts de l'ontologie de domaine comme mots clef. Ces documents étant créés par des personnes ou des applications différentes, leur structure HTML n'est pas régulière, contrairement, par exemple, aux pages d'un site web qui seraient générées automatiquement en utilisant des patrons structurels alimentés à partir d'une base de données. Lorsque le nombre de documents est grand, l'annotation sémantique manuelle est trop coûteuse. Aussi, nous proposons une approche complètement automatique dont l'objectif n'est pas de délimiter des instances de concepts le plus parfaitement possible mais de repérer la présence possible d'une instance dans un noeud de document et d'annoter ce noeud.

Nous présentons tout d'abord les apports de notre approche et ensuite, nous présentons les perspectives que nous avons dégagées pour nos travaux.

1 Apports de notre approche

Nous avons défini une approche d'annotation sémantique qui combine l'usage d'une terminologie présente dans l'ontologie lexicale et la capacité d'apprendre de nouveaux termes en utilisant la base généraliste de documents qu'est le Web. L'approche que nous avons proposée a été définie pour être indépendante du domaine et pour annoter des documents de structure hétérogène. Notre approche permet également d'enrichir cette terminologie du domaine et l'ensemble des entités nommées associées aux concepts de l'ontologie.

- Le premier apport de notre approche concerne les étapes d'extraction et d'alignement définies dans *SHIRI-Extract*. Une fois que les termes ou EN candidats sont repérés dans les documents en utilisant des patterns d'extraction classiques et indépendants du domaine, ces termes sont rapprochés (alignés) en utilisant une méthode locale ou, en cas d'échec, en exploitant les documents présents sur le Web.
  Pour une EN, la méthode locale recherche une EN similaire parmi les EN décrites par la relation $hasTermNe$ dans l'ontologie. Dans l'implémentation, la mesure de similarité pour l'alignement d'une EN est pour l'instant très stricte puisqu'il s'agit de l'égalité.
  Pour un terme, les variantes syntaxiques sont plus nombreuses et plus complexes. Notre méthode utilise des mesures de similarité qui tiennent compte de la catégorie syntaxique des mots composant le terme, de l'importance des mots dans le terme. Elles permettent également d'exploiter l'ensemble des labels et des termes associés à un concept de l'ontologie et aussi ceux associés aux concepts frères ou pères de ce concept dans l'ontologie. Notre approche utilise (et détourne) l'outil d'alignement de taxonomies Taxomap. Cet outil permet de découvrir trois types de relations de correspondance : l'équivalence, la subsumption et la proximité sémantique que l'on a distinguées au moment de l'alignement.

  Il est clair que ces méthodes locales sont insuffisantes pour aligner toutes les EN et tous les termes. La composante lexicale ne sera jamais suffisamment riche pour qu'une EN appartienne toujours à la composante lexicale ou qu'un terme candidat corresponde toujours à une variation syntaxique des termes présents dans cette composante. La capacité d'un système d'annotation de découvrir de nouveaux termes ou de nouvelles EN est une caractéristique importante d'un système d'annotation. En effet, les ressources lexicales sont rarement complètes et leur mise à jour est une tâche fastidieuse : il se crée par exemple de nouvelles conférences régulièrement. Aussi, en cas d'échec des méthodes locales, nous utilisons le Web pour découvrir des termes qui peuvent être des hypéronymes du terme candidat à aligner ou des termes décrivant la catégorie de l'EN candidate. Ce sont ces termes que l'on va alors tenter d'aligner avec l'ontologie, et qui permettront alors d'aligner l'EN ou le terme candidat.
  La méthode d'alignement via le Web suppose que l'EN ou le terme candidat soit suffisamment redondant sur le Web et que le terme provenant du web puisse être aligné en utilisant la méthode locale.

  Nos expériences ont porté sur une collection de 412 documents HTML relatifs à des conférences en informatique. Les labels et les termes de la composante lexicale ont été sélectionnés en utilisant WordNet. En revanche, la composante lexicale ne comporte aucune entité nommée.
  Les résultats montrent que la méthode d'alignement via le Web a permis de rapprocher environ 75 % des entités nommées comme des événements, des personnes et des lieux. La précision des résultats de l'alignement obtenus en utilisant les méthodes locales ou via le Web sur un échantillon de termes et d'EN est d'environ 73 % pour les termes et de 92 % pour les EN. Nous soulignons que le calcul de ces précisions ne

tient pas compte des erreurs dues à l'extraction (erreurs typographiques ou morpho-syntaxiques).

- Le deuxième apport de notre approche concerne l'enrichissement de la composante lexicale de l'ontologie au fur et à mesure de l'annotation des documents. Cet enrichissement concerne à la fois la liste des termes associés aux concepts et la liste des EN : l'ontologie est en effet enrichie par les nouveaux termes ou nouvelles EN découverts dans les documents et par les termes découverts sur le web. Ces nouveaux termes ou EN devront être validés par l'expert qui pourra modifier, supprimer ou reclasser le terme ou l'EN, ou choisir d'en faire un nouveau label de concept. Il pourra également choisir de faire évoluer l'ontologie conceptuellement en créant de nouveaux concepts. Cet enrichissement permet à la fois de faire évoluer l'ontologie mais aussi de limiter le besoin d'utilisation du web.
  Nous avons également conservé pour chaque terme et ENs candidats qui n'ont pas pu être alignés, les termes du web fournis par les réponses aux requêtes soumises au moteur de recherche. Si ce terme apparaît une nouvelle fois, nous ne tentons plus de le réaligner via le Web mais, l'ontologie étant enrichie, il est possible que l'alignement aboutisse.
  Nous avons pu montrer dans les expérimentations que plus le nombre de documents (et donc de termes) traités augmentait, plus le nombre d'appels web par rapport au nombre de termes traités baissait.

- Le troisième apport de notre approche est d'avoir défini un modèle d'annotation sémantique qui est généré à partir de l'ontologie de domaine. Il permet de représenter différents types d'annotations sémantiques de nœuds de document : (1) un nœud qui ne contient qu'un terme ou qu'une EN alignée avec des concepts est considéré comme étant une instance de chaque concept, (2) un nœud qui ne contient que des termes ou EN alignées avec des concepts comparables sera considéré comme une instance du concept de type ensemble (*SetOfConcept*) correspondant au concept le plus général repéré, (3) un nœud qui contient des termes ou EN alignées avec des concepts non comparables sera considéré comme une instance d'une partie de discours (*PartOfSpeech*) qui sera indexée par les concepts repérés. Ainsi, si un document présente la liste des noms de chercheurs d'un laboratoire, chaque élément HTML de la liste sera considéré comme une instance de *Chercheur* tandis qu'un extrait textuel de rapport d'activité parlant d'un chercheur, et de ses thèmes de recherche sera simplement indexé, entre autres, par le concept *Chercheur*. Une telle distinction permet de formuler des requêtes qui vont atteindre certains types de noeuds selon les objectifs de l'utilisateur ou de l'application utilisant les annotations. Par exemple, les mêmes données sont parfois présentées de manière structurée et parfois mêlées à d'autres informations dans des parties textuelles du document, et l'utilisateur peut préferer obtenir, si elles existent, les annotations associées aux nœuds structurés. Le modèle d'annotation permet également de représenter des noeuds voisins qui contiennent des termes ou ENs de concepts susceptibles d'être liés par une relation sémantique. Nous avons pu montrer que si la distance d utilisée pour définir le voisinage est petite ($d < 4$), il existe bien des instances de relations sémantiques entre instances

décrites dans les nœuds.

SHIRI-Annot définit un ensemble de règles d'annotations qui permettent de passer de la pré-annotation des nœuds de documents à leur annotation sémantique conforme au modèle d'annotation. Elles sont déclaratives et peuvent facilement être modifiées ou utilisées partiellement.
Pour interroger les annotations produites, l'utilisateur peut utiliser le module *SHIRI-Querying* qui permet à l'utilisateur de soumettre des requêtes exprimées dans le vocabulaire de l'ontologie de domaine. *SHIRI-Querying* construit toutes les reformulations d'une requête utilisateur pour atteindre tous les types de nœuds réponses. Ces reformulations sont construites et exécutées selon une fonction d'ordre qui tient compte du niveau d'agrégation des termes et des EN dans les nœuds.
L'utilisation de ce module d'interrogation sur notre base d'annotations a permis de montrer l'intérêt de notre modèle pour obtenir les réponses les plus précises d'abord.

2 Perspectives

Nous présentons un ensemble de perspectives dont certaines sont des améliorations possibles de notre approche et d'autres sont plus prospectives.

- La première perspective à court terme de ce travail est d'expérimenter notre approche en exploitant une ontologie de taille plus importante, portant sur un autre domaine. Il faudrait adapter les composants *SHIRI-Extract* et *SHIRI-Annot* pour permettre l'intégration d'outils existants dans une même plateforme.

- L'extraction pourrait être améliorée : les patterns utilisés par *SHIRI-Extract* sont très généraux et extraient de nombreux termes qui ne font pas partie du domaine d'interêt défini dans l'ontologie ou des faux termes dus, en partie, à la présence d'erreurs typographiques. L'approche pourrait utiliser des méthodes de filtrage statistiques afin de ne pas traiter l'ensemble des termes extraits par les patterns. Ceci diminuerait le nombre de faux termes ou de fausses entités nommées et ainsi le bruit introduit dans les annotations. De plus, notre approche pourrait apprendre des patterns spécifiques plus précis en utilisant les EN et les termes validés de la composante lexicale comme dans KnowItAll [ECD+05].

- La méthode d'alignement locale utilisée pour rapprocher un terme de l'ontologie exploite les variations syntaxiques des termes mais ne permet pas de rapprocher un terme d'un concept de l'ontologie s'il ne possède aucune ressemblance syntaxique avec les termes associés au concept. L'alignement utilisant le web doit permettre d'aligner des termes syntaxiquement très différents mais les expérimentations ont montré que cette méthode est beaucoup moins performante pour les termes que pour les EN. Aussi, pour améliorer le rappel de l'alignement, il nous faudrait intégrer à la méthode locale l'utilisation d'une ressource lexicale permettant de retrouver des liens de synonymie ou d'hyponymie telle que WordNet.

- Les méthodes d'alignement utilisées associent des mesures de similarité au rapprochement fait entre un terme et un concept du domaine. Nous n'exploitons pas ces mesures et conservons tous les rapprochements pour lesquels la valeur de similarité est au dessus d'un seuil donné. Nous pourrions attribuer une mesure de confiance à chaque annotation de nœud en exploitant (1) la valeur obtenue par la mesure de similarité exploitée par la méthode d'alignement, (2) l'origine des termes utilisées dans la composante lexicale pour effectuer le rapprochement : un label de concept, un terme validé par l'expert, un terme non encore validé. Les mesures de confiance associées aux annotations pourraient alors être représentées dans la base d'annotations et exploitées à l'interrogation pour classer les réponses.

- Il est fréquent qu'un élément dans une page doive être annoté par un concept parce que d'autres éléments de cette même page sont annotés par d'autres concepts. Ainsi, un nœud dont les nœud fils sont des instances de chercheurs est peut-être une instance de comité. Un nœud de document dans lequel apparaissent des dates et un nom de conférence décrit peut-être un appel à participation.

- Notre approche permet d'enrichir la composante lexicale de l'ontologie. Pour compléter notre approche, il faudrait développer une interface permettant à l'expert de valider ou de modifier les nouveaux éléments de cette composante lexicale et éventuellement d'enrichir conceptuellement l'ontologie. Cela suppose également de savoir gérer les conséquences de l'évolution de l'ontologie sur les annotations : une annotation réalisée avant modification faite sur l'ontologie est susceptible d'être modifiée pour rester conforme au modèle d'annotation et donc à l'ontologie de domaine.

Table des figures

Liste des tableaux

Bibliographie

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia : A nucleus for a web of open data. In *ISWC/ASWC*, pages 722–735, 2007.

[AGM03] Arvind Arasu and Hector Garcia-Molina. Extracting structured data from web pages. In *SIGMOD Conference*, pages 337–348, 2003.

[AH06] Sophie Aubin and Thierry Hamon. Improving term extraction with terminological resources. In *FinTAL*, pages 380–387, 2006.

[AKM+03] Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1) :14–21, 2003.

[Ama07] Florence Amardeilh. *Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle*. PhD thesis, Université de Nanterre - Paris X, France, Mai 2007.

[Arp95] Antti Arppe. Term extraction from unrestricted text. In *the Nordic Conference on Computational Linguistics (NoDaLiDa)*, 1995.

[BAGC04] Didier Bourigault, Nathalie Aussenac-Gilles, and Jean Charlet. Construction de ressources terminologiques ou ontologiques à partir de textes un cadre unificateur pour trois études de cas. *Revue d'Intelligence Artificielle*, 18(1) :87–110, 2004.

[BCS+07] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[BFG01a] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. The elog web extraction language. In *LPAR*, pages 548–560, 2001.

[BFG01b] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual web information extraction with lixto. In *VLDB*, pages 119–128, 2001.

[BL98] Tim Berners-Lee. Semantic web road map. 1998.

[Bou] Didier Bourigault. Conception et exploitation d'un logiciel d'extraction de termes : problèmes théoriques et méthodologiques.

[Bou92] Didier Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *COLING*, pages 977–981, 1992.

[BS99] Brigitte Biebow and Sylvie Szulman. Terminae : A linguistic-based tool for the building of a domain ontology. pages 49–66, 1999.

[CCDW04] Fabio Ciravegna, Sam Chapman, Alexiei Dingli, and Yorick Wilks. Learning to harvest information for the semantic web. In *ESWS*, pages 312–326, 2004.

[CDKFZ04] Olivier Corby, Rose Dieng-Kuntz, and Catherine Faron-Zucker. Querying the semantic web with corese search engine. In *ECAI*, pages 705–709, 2004.

[CDKFZG05] Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker, and Fabien Gandon. Ontology-based approximate query processing for searching the semantic web with corese. In *INRIA Report*, july 2005.

[CDWP03] Fabio Ciravegna, Alexiei Dingli, Yorick Wilks, and Daniela Petrelli. Using adaptive information extraction for effective human-centred document annotation. In *Text Mining*, pages 153–164, 2003.

[CGW95] H. Cunningham, R.G. Gaizauskas, and Y. Wilks. A General Architecture for Text Engineering (GATE) – a new approach to Language Engineering R&D. Technical Report CS – 95 – 21, Department of Computer Science, University of Sheffield, 1995. `http://xxx.lanl.gov/abs/cs.CL/9601009`.

[Cir01] Fabio Ciravegna. Adaptive information extraction from text by rule induction and generalisation. In *IJCAI*, pages 1251–1256, 2001.

[CLS05] Philipp Cimiano, Günter Ladwig, and Steffen Staab. Gimme' the context : context-driven automatic semantic annotation with c-pankow. In *WWW*, pages 332–341, 2005.

[DES05] Doug Downey, Oren Etzioni, and Stephen Soderland. A probabilistic model of redundancy in information extraction. In *IJCAI*, pages 1034–1041, 2005.

[DVN05] Hasan Davulcu, Srinivas Vadrevu, and Saravanakumar Nagarajan. Ontominer : automated metadata and instance mining from news websites. *IJWGS*, 1(2) :196–221, 2005.

[ECD+05] Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web : An experimental study. volume 165, pages 91–134, 2005.

[Eva03] Richard J. Evans. A framework for named entity recognition in the open domain. In *RANLP*, pages 267–276, 2003.

[Gru93] Tom R. Gruber. Towards principles for the design of ontologies used for knowledges sharing. In *Conference that does not exist*, 1993.

[GS96] Ralph Grishman and Beth Sundheim. Message understanding conference-6 : A brief history. In *COLING*, pages 466–471, 1996.

[HBDBH07] Gaëlle Hignette, Patrice Buche, Juliette Dibie-Barthélemy, and Ollivier Haemmerlé. Semantic annotation of data tables using a domain ontology. In *Discovery Science*, pages 253–258, 2007.

[Hea92] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545, 1992.

[HSC02] Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-cream - semi-automatic creation of metadata. In *EKAW*, pages 358–372, 2002.

[HZSR08] Faycal Hamdi, Haifa Zargayouna, Brigitte Safar, and Chantal Reynaud. Taxomap in the oaei 2008 alignment contest, ontology alignment evaluation initiative (oaei) 2008 campaign. In *Int. Workshop on Ontology Matching*, 2008.

[Jac94] Christian Jacquemin. Fastr : A unification-based front-end to automatic indexing. In *RIAO*, pages 34–48, 1994.

[KK01] José Kahan and Marja-Riitta Koivunen. Annotea : an open rdf infrastructure for shared web annotations. In *WWW*, pages 623–632, 2001.

[KT07] Jun'ichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL*, pages 698–707, 2007.

[Lin98] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, pages 296–304, 1998.

[LSF07] Dimitrios P. Lyras, Kyriakos N. Sgarbas, and Nikolaos D. Fakotakis. Using the levenshtein edit distance for automatic lemmatization : A case study for modern greek and english. In *ICTAI (2)*, pages 428–435, 2007.

[MBPT10a] Yassine Mrabet, Nacéra Bennacer, Nathalie Pernelle, and Mouhamadou Thiam. Supporting semantic search on heterogeneous semi-structured documents. In *CAISE*, Tunisie, 2010.

[MBPT10b] Yassine Mrabet, Nacéra Bennacer, Nathalie Pernelle, and Mouhamadou Thiam. Une approche pour la recherche sémantique de l'information dans les documents semi-structurés hétérogènes. In *CORIA*, pages 195–210, 2010.

[McB01] Brian McBride. Jena : Implementing the rdf model and syntax specification. In *SemWeb*, 2001.

[P.07] Senellart P. *Understanding the Hidden Web*. PhD thesis, University of Paris 11, France, December 2007.

[PAD+08] Senellart P., Mittal A., Muschick D., Gilleron R., and Tommasi M. Automatic wrapper induction from hidden-web sources with domain knowledge. In *WIDM'08, ACM*, 2008.

[PKK+04] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. Kim - semantic annotation platform. pages 375–392, 2004.

[RTAG07] Axel Reymonet, Jérôme Thomas, and Nathalie Aussenac-Gilles. Modélisation de ressources termino-ontologiques en owl. In *Actes d'IC*, pages 169–181, 2007.

[SC04] Sekine Satoshi and Nobata Chikashi. Definition, dictionaries and tagger for extended named entity hierarchy. In *Conference on Language Resources and Evaluation*, 2004.

[SM07] Stephen Soderland and Bhushan Mandhani. Moving from textual relations to ontologized relations. 2007.

[TBP07] Mouhamadou Thiam, Nacéra Bennacer, and Nathalie Pernelle. Webdocenrich : enrichissement sémantique flexible de documents semi-structurés. In *EGC*, pages 211–212, 2007.

[TBPL09] Mouhamadou Thiam, Nacéra Bennacer, Nathalie Pernelle, and Moussa Lo. Incremental ontology-based extraction and alignment in semi-structured documents. In *DEXA*, pages 611–618, 2009.

[TM98] Dan Tufis and Oliver Mason. Tagging romanian texts : a case study for qtag, a language independent probabilistic tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, pages 589–596, 1998.

[TPB08] Mouhamadou Thiam, Nathalie Pernelle, and Nacéra Bennacer. Contextual and metadata-based approach for the semantic annotation of heterogeneous documents. In *SeMMA*, pages 18–30, 2008.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[VGP01] Crescenzi V., Mecca G., and Merialdo P. Roadrunner : Towards automatic data extraction from large web sites. In *Very Large Data Bases Conference (VLDB)*, 2001.

[W3Ca] W3C. http ://www.w3c.org.

[W3Cb] W3C. http ://www.w3.org/tr/2004/rec-rdf-primer-20040210/.

[W3Cc] W3C. http ://www.w3.org/tr/2004/rec-rdf-schema-20040210/.

Résumé

Le web sémantique est défini par un ensemble de méthodes et de technologies permettant à des agents logiciels de raisonner sur le contenu des ressources du Web. Cette vision du Web dépend de la construction des ontologies et de l'utilisation de métadonnées pour représenter ces ressources. L'objectif de notre travail de thèse est d'annoter sémantiquement des documents balisés et relatifs au mÃ^ame domaine. Ces documents peuvent comporter des parties bien structurées et d'autres textuelles. Nous supposons disposer d'une ontologie de domaine définie par des concepts, des relations entre ces concepts et des propriétés. Cette ontologie comporte une composante lexicale où chaque concept est accompagné de labels, d'un ensemble d'entités nommées (EN) et de termes du domaine. Nous avons défini une approche automatique SHIRI-Extract *qui permet d'extraire des termes et des EN de manière indépendante du domaine et de les aligner aux concepts de l'ontologie. L'alignement utilise la composante lexicale ou le Web pour découvrir de nouveaux termes. Nous avons défini un modèle d'annotation représentant les résultats d'extraction et d'annotation. Les métadonnées de ce modèle distinguent les nœuds selon que les termes ou les EN agrégés dans un même nœud sont alignés avec un ou plusieurs concepts différents. Elles permettent également d'annoter la relation de voisinage entre les nœuds. Nous avons défini* SHIRI-Annot*, un ensemble de règles déclaratives pour annoter les nœuds et leurs relations. La base d'annotations RDF(S) construite peut être interrogée à l'aide de requêtes SPARQL. L'évaluation a porté sur une collection de documents portant sur des appels à participation à des conférences.*

Mots-clés: Web sémantique, RDF(S), SPARQL, Annotation sémantique, Extraction de termes et d'entités nommées, Documents semi-structurés, Recherche d'information, Ontologies

Abstract

The semantic web is defined by a set of methods and technologies enabling software agents to reason about the contents of Web resources. This vision of the Web depends on the construction of ontologies and the use of metadata to represent these resources. The objective of our thesis is to annotate semantically tagged documents related to a domain of interest. These documents may contain well-structured nodes and textual ones. We assume having a domain ontology defined by concepts, relations between these concepts and their properties. This ontology includes a lexical component (labels, a set of named entities (NE) and terms) for each concept. We have defined an automatic and domain independent approach SHIRI-Extract that extracts terms and NE and aligns them with the concepts of the ontology. The alignment uses the lexical component or the Web to discover new terms. We have defined an annotation model which represents the results of extraction and annotation. The metadata of this model distinguish nodes depending on

how the terms and NE are aligned and aggregated in a node. The model can also represent the structural neighbor relations between nodes. We have defined SHIRI-Annot, a set of declarative rules to annotate the nodes and their relations. The constructed RDF(S) annotation base can be queried using SPARQL. We have implemented and evaluated our approach on a collection of call for participation to computer science conferences.

Keywords: Semantic Web, Semantic Annotation, Ontologies, Semi-structured documents, REF(S), SPARQL, Information retrieval, Terms and named entities extraction.