B. Espinasse, Lima R., Magdy D., « Extraction automatique d'entités et de relations par ontologies et programmation logique inductive », Journée Francophones sur les Ontologies - JFO 2016, 13-14 Octobre 2016, Bordeaux, France.

Extraction automatique d'entités et de relations par ontologies et programmation logique inductive

Bernard Espinasse[1], Rinaldo Lima[2], Diana Magdy[1]

1. Aix-Marseille Université, LSIS UMR CNRS 6168
Domaine Universitaire de St Jérôme, F-13997, Marseille cedex 20, France
bernard.espinasse@lsis.org, diana.magdi@gmail.com

2. Universidade Federal Rural de Pernambuco – UFRPE-DEINFO)
Rua Dom Manoel de Medeiros, s/n, Campus Dois Irmãos, Recife/PE, Brasil
rinaldo.jose@ufrpe.br

RESUME. *Pour être plus précis, les systèmes d'extraction d'information automatiques doivent exploiter des ressources sémantiques, notamment des ontologies. Pour être plus rapidement développés et adaptables à d'autres domaines d'application, ils se doivent d'utiliser des techniques d'apprentissage automatique. Le système OntoILPER est un système d'extraction d'information utilisant des ontologies et la programmation logique inductive (PLI), une technique d'apprentissage symbolique, pour induire des règles d'extraction symboliques. Ce papier présente comment, dans sa phase d'extraction, ce système applique ces règles pour extraire des instances d'entités et de relations d'un corpus de textes conséquent en utilisant un triple-store. Le système est évalué sur l'extraction d'instances d'entités et de relations du corpus de référence reACE2004, et le peuplement d'une ontologie de domaine à partir de ces instances extraites.*

MOTS-CLES : *Extraction d'information dans des documents textuels, extraction d'entités et de relations; ontologies; Apprentissage symbolique; Programmation logique inductive; Peuplement d'ontologie.*

1. Introduction

L'extraction d'information (EI) consiste à reconnaître et extraire certains types d'informations à partir de textes. Deux sous-tâches importantes dans l'EI sont la *Reconnaissance d'Entités Nommées* (REN) et *l'Extraction de Relation* (ER). La première vise à extraire des instances nommées dans le texte, notamment des noms de personnes, de lieux, tandis que la seconde consiste à extraire des relations entre ces entités nommées. Le développement de systèmes d'EI efficaces et robustes constitue un grand défi, notamment dans le contexte du Web, ainsi que d'exploitation de bibliothèques numériques.

Pour être plus précis, de tels systèmes d'IE se doivent d'exploiter de plus en plus de ressources sémantiques disponibles (notamment des ontologies) (Nédellec & Nazarenko, 2005). Récemment, l'EI basée sur des ontologies (*Ontology-Based Information Extraction* - *OBIE*) a émergé comme un sous-domaine de l'EI dans lequel les ontologies sont utilisées tant dans le processus d'EI même, que dans sa sortie assimilée à le peuplement d'ontologie.

Pour être plus rapidement développés et adaptables à d'autres domaines d'application, de tels systèmes d'EI se doivent d'utiliser des techniques d'apprentissage automatique. L'apprentissage automatique supervisé est largement utilisé pour les deux tâches de REN et de ER, principalement en utilisant des méthodes statistiques. Pour la REN, la performance de systèmes d'EI utilisant ces méthodes statistiques est autour de 90%, nettement inférieure pour l'ER (Giuliano et al., 2007). Ainsi, ces *méthodes statistiques supervisées* qui s'appuient principalement sur la distribution des données, font face à des difficultés significatives dans l'ER (Bach et Badaskar, 2007). Une alternative à l'approche supervisée statistique est *l'approche supervisée symbolique*, dans laquelle des exemples d'entités ou de relations cibles sont utilisées pour la construction de règles d'extraction basées sur des prédicats logiques et exploitant les informations structurelles des exemples (Muggleton et al., 1991).

Un système d'extraction d'information, nommé OntoILPER (ONTOlogy and Inductive Logic Programming based method to extract instances of Entities and Relations from texts) a été développé selon une approche supervisée symbolique (Lima, 2014). Ce système permet d'extraire des instances d'entités et de relations d'un document textuel, par l'usage d'ontologies et de Programmation Logique Inductive (*Inductive Logic Programming* - ILP), une technique d'apprentissage automatique symbolique (Lavrac et Dzeroski, 1994). Ce système fonctionne selon deux grandes phases distinctes : une *phase d'apprentissage (ou d'induction)* et une *phase d'extraction*. Il a été testé sur plusieurs corpus de référence et a donné des résultats surpassant ceux d'autres systèmes d'EI mettant en œuvre des méthodes statistiques (Lima et al., 2015, Lima, 2014).

Dans OntoILPER, la phase d'extraction était jusqu'à présent réalisée de façon semi-automatique dans l'environnement Protégé conduisant à des temps de traitement importants, et ne permettant pas de traiter de gros corpus. Ce papier, centré sur cette phase d'extraction, présente comment elle a été rendue opérationnelle avec l'intégration du triple-store StarDog, pour extraire automatiquement et efficacement des instances d'entités et de relations dans des corpus de textes de grande taille.

Ce papier est organisé de la façon suivante. La Section 2 introduit l'OBIE (Ontology Based Information Extraction) en montrant l'intérêt d'utiliser des ontologies dans l'extraction d'information. La section 3 présente brièvement le système OBIE OntoILPER avec ses phases d'apprentissage et d'extraction. La section 4 présente comment la phase d'extraction a été opérationnalisée dans OntoILPER en intégrant le triple-store StarDog. La section 5 présente des résultats d'expérimentation obtenus avec le corpus de référence reACE2004. Enfin, la section 6 conclut cet article en présentant plusieurs perspectives de recherche.

2. L'extraction d'information à base d'ontologies (OBIE)

Le terme *OBIE « Extraction d'Information Basée sur les Ontologies »* a été introduit depuis plusieurs années et constitue un sous domaine de l'EI. L'OBIE est différente de l'EI traditionnelle du fait qu'elle identifie les types des entités extraites en les reliant à leur description sémantique dans l'ontologie. L'ontologie, spécification explicite et formelle d'une conceptualisation, joue un rôle majeur dans le processus d'extraction. Selon (Wimalasuriya et Dou, 2010), ce qui caractérise le plus les systèmes OBIE est d'avoir recours au traitement du langage naturel en profondeur, et d'utiliser des ontologies tant en entrée du processus d'extraction, qu'en sortie de celui-ci :

- *Ontologie en entrée* : le processus d'EI est guidé par une ontologie pour extraire des informations telles que les classes, les propriétés et les instances, ceci grâce à une annotation sémantique des textes à traiter.

- *Ontologie en sortie* : les systèmes OBIE utilisent une ontologie pour représenter, stocker les informations extraites, par le peuplement d'une ontologie.

Un des plus importants potentiels de l'OBIE réside dans sa capacité à générer automatiquement des contenus sémantiques pour le Web sémantique (Wu and Weld, 2008), qui a pour but d'ajouter de la sémantique au Web classique pour le rendre plus intelligent et plus explicite. Il est assez difficile d'imaginer que ces contenus doivent être annotés manuellement, étant donné l'énorme quantité des informations sur le Web. En conséquence, une génération massive de métadonnées est nécessaire (Popov et al., 2004). Dans ce contexte, l'OBIE permet d'ajouter de la sémantique au Web sémantique en convertissant les informations contenues dans les sources semi ou non structurées en éléments ontologiques. Dans le système OntoILPER, ce processus appelé aussi « *d'annotation sémantique* », est utilisé dans l'apprentissage supervisé pour induire des règles d'extraction.

Enfin, le peuplement (ou population) d'ontologie (PO) consiste à ajouter de nouvelles instances de classes, de propriétés et de relations dans une ontologie existante (Petasis et al., 2011). Notons que la PO doit être clairement distinguée de l'enrichissement d'ontologie, dans lequel il s'agit d'ajouter de nouveaux concepts et relations au modèle formel d'une ontologie. La PO joue un rôle important dans la construction de base de connaissances (Maynard et al., 2008) en permettant de relier des données écrites en langage naturel avec des ontologies, ce qui facilite la génération de contenus sémantiques (Wimalasuriya et Dou, 2010).

3. Le système OntoILPER

Le système d'extraction d'information OntoILPER (Lima et al., 2015) permet d'extraire des instances d'entités et de relations d'un document textuel, par l'usage d'ontologie et de Programmation Logique Inductive (PLI), une technique d'apprentissage machine symbolique (Lavrac et Dzeroski, 1994). Sur le corpus TREC d'articles du Wall Street Journal[1], en extraction d'instances de relations,

[1] (http://cogcomp.cs.illinois.edu/Data/ER/conll04.corp),

OntoILPER a surpassé deux autres systèmes statistiques en terme de *précision*, *rappel* et *F-mesure* (Lima et al., 2013).

3.1 Usage de la PLI pour l'extraction d'information

La Programmation Logique Inductive (PLI), est une technique d'apprentissage symbolique, permettant de trouver des modèles à partir de données stockées dans des structures de données complexes. Ces modèles sont utilisés pour classer de nouveaux exemples en positif ou en négatif. La PLI utilise la programmation logique, les clauses de premier ordre, comme langage de représentation uniforme pour les exemples, les connaissances a priori ou *Background Knowledge* (BK), des hypothèses, et les modèles induits (Lavrac et Dzeroski, 1994). Elle permet aussi un apprentissage en prenant en compte des connaissances expertes disponibles afin d'augmenter l'expressivité de l'espace d'hypothèses.

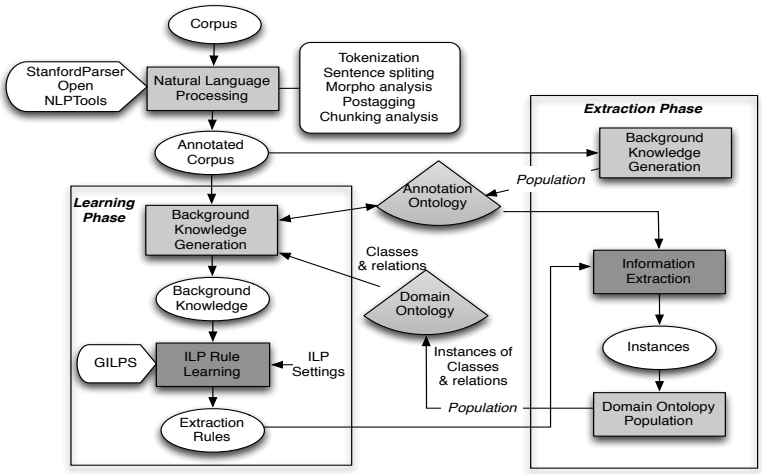

Figure 1. Les composants du système OntoILPER.

L'extraction dans OntoILPER avec PLI s'appuie sur une représentation symbolique (non vectorielle) de la phrase sous forme de graphe dont chaque arête donne un prédicat logique binaire. Cette représentation peut incorporer d'autres connaissances notamment issues d'ontologies sous la forme d'attributs (ou caractéristiques) pouvant améliorer les performances du processus d'extraction. Dans OntoILPER, l'extraction d'informations est réalisée en deux phases distinctes, mettant en œuvre des composants logiciels, communs ou spécifiques à chacune des phases, comme l'illustre la Fig. 1. La première phase est la *phase d'apprentissage* dans laquelle, à partir d'un corpus d'exemples de documents annotés, d'une ontologie de domaine, est induite par apprentissage en PLI une théorie, théorie correspondant à un ensemble de règles symboliques d'extraction (cf. Fig. 1). La seconde phase est la *phase d'extraction*, dans laquelle l'ensemble final de règles d'extraction induites est appliqué à des documents annotés pour en extraire des instances d'entités et de relations entre elles. Ces instances sont ensuite utilisées pour peupler l'ontologie de domaine. Ces deux phases partagent un même

prétraitement des documents permettant de les annoter en utilisant différentes techniques de TAL (Traitement Automatique des Langues), ainsi qu'un traitement permettant à partir de ces documents annotés de générer une BK.

3.2 Utilisation d'ontologies dans OntoILPER

OntoILPER utilise deux ontologies différentes comme l'illustre la Fig. 1. La première est une *ontologie de domaine*, associée au domaine dans lequel se fait l'extraction. Elle est utilisée dans la phase d'apprentissage pour réaliser une annotation sémantique du corpus d'apprentissage, ainsi que pour préciser les instances que l'on souhaite extraire du corpus de documents. Elle est aussi utilisée pour stocker les instances extraites en la peuplant. La seconde ontologie est *l'ontologie d'annotation*. C'est une ontologie opératoire qui a été rajoutée à OntoILPER pour améliorer sa performance et son adaptabilité. Comme nous le verrons par la suite, elle permet de stocker les annotations des documents en phase d'extraction et de mémoriser les instances extraites en la peuplant.

3.3 Principaux composants de OntoILPER

Comme l'illustre la Fig. 1. OntoILPER est composé de différents composants logiciels, que nous décrivons succinctement.

- Le premier composant "*Natural Language Processing*" – Traitement Automatique des Langues est mis en œuvre en amont des deux phases d'apprentissage et d'extraction. Il effectue une annotation automatique du corpus de document textuels d'entrée au moyen d'outils de TAL (morphosyntaxique et sémantique) utilisé pour l'élaboration d'un modèle de phrase symbolique spécifique. Rappelons que dans OntoILPER, on s'intéresse ici essentiellement à des documents en langue anglaise. Ce composant intègre ainsi principalement les outils Stanford CoreNLP[2] et OpenNLP[3]. Ils réalisent séquentiellement les tâches suivantes : segmentations des phrases, tokenisation, étiquetage morpho-syntaxique, lemmatisation, analyse de syntagmes (*chunking*), et analyse de dépendances (de Marneffe et Manning, 2008).

Dans la phase d'apprentissage on distingue les composants suivants :

- Le composant "*Background Knowledge Generation*". Il génère automatiquement la base de connaissances *a priori* (*Background Knowledge*), contenant les caractéristiques (*features*) pertinentes à la fois pour le corpus annoté de documents et pour l'ontologie de domaine. La représentation symbolique des phrases est un graphe, dont chaque arête est une caractéristique, principalement associée à une annotation spécifique du texte, et représentée par un prédicat logique binaire. Cet ensemble de caractéristiques constituant les connaissances de la BK se traduit en une base factuelle Prolog, et comme des instances de l'ontologie d'annotation dans la phase d'extraction. On distingue quatre principaux groupes de caractéristiques :

. [2] - Stanford CoreNLP Tools. http://nlp.stanford.edu/software/corenlp.shtml.

. [3] - Apache OpenNLP. The Apache Software Foundation. http://opennlp.apache.org

(i) les *caractéristiques lexicales* concernent le mot lui-même, son lemme, la longueur, et des informations générales de type morphologique ; (ii) les *caractéristiques syntaxiques* comprenant des étiquettes des tokens; mot de tête du syntagme nominal, prépositionnel ou verbal; (iii) les *caractéristiques sémantiques* comprennent les entités nommées reconnus par la REN, et les entités additionnelles mentionnées dans le corpus d'entrée, (iv) et enfin les *caractéristiques structurelles* concernent toutes les caractéristiques structurelles spécifiques au modèle de représentation de la phrase à base de graphes retenu (Lima, 2014). Ainsi dans la phase d'apprentissage d'OntoILPER, les exemples, les entités, les relations et tous les types de caractéristiques précédentes sont convertis en prédicats Prolog.

- Le composant "*ILP Rule Learning*". Il est basé sur le système de PLI nommé GILPS développé par (Santos, 2010). Ce système, programmé en Yap-6 Prolog, utilise différents algorithmes de base (ou stratégies) PLI comme TopLog, FuncLog et ProGolem. GIPS permet à OntoILPER d'induire des règles d'extraction dans la syntaxe Prolog, ceci à partir des exemples annotés. Ainsi une règle induite pour l'extraction de la relation binaire *part_whole* est la suivante :

#Literals = 4, Pos. Score = 90; Neg. Score = 1; P = 98.9%
part_whole(A,B):- t_gpos(A,nn), t_next(A,B), t_subtype(B,state-or-province).

Cette règle classifie une instance de la relation *part_whole*. Sa grande précision (P = 98,9) est liée au nombre élevé de phrases contenant deux tokens adjacents où la première (A) est un nom, et la seconde (B) est étiqueté, par rapport à l'ontologie de domaine, comme une instance de la classe "State-or-Province". Cette règle met en évidence que les lieux (A) comme « *ville* » sont situés, ou font partie soit d'un Etat ou d'une Province. Plus de détails sur ce composant dans (Lima et al., 2015).

Dans la phase d'extraction, on distingue trois composants qui seront développés dans la section suivante : (i) le composant "*Backgroud Knowledge Generation*" permet de prendre en compte les différentes annotations faites sur les textes et de générer la base de connaissances à priori (BK), (ii) le composant "*Information Extraction*" applique les règles d'extraction symboliques apprises en phase d'apprentissage sur la BK, et enfin (iii) le composant "*Domain Ontology Population*", qui à partir des instances extraites trouvées précédemment, peuple l'ontologie de domaine (Petasis et al., 2011).

4. Opérationnalisation de la phase d'extraction dans OntoILPER

Cette section présente comment la phase d'extraction d'OntoILPER a été opérationnalisée pour extraire de façon efficace des instances d'entités et de relations sur des corpus de textes importants en intégrant un triple-store. Rappelons qu'un triple store est une base de données spécialement conçue pour le stockage de triplets RDF, interrogeable par le langage SPARQL.

Nous avons retenu le triple-store Stardog (http://stardog.com/) pour ses performances notamment en raisonnement du fait qu'il utilise de façon optimale le raisonneur Pellet grâce à une mémoire virtuelle spécifique. Ce triple-store, développé en Java, a été utilisé pour stocker les ontologies d'annotation et de domaine, qui ont dû être traduites en triplets RDF/XML. Le triple-store appelle

ensuite le raisonneur Pellet, pour appliquer ces règles, et ensuite réalise le peuplement de l'ontologie de domaines par les instances extraites. Autour de ce triple-store, différents composants logiciels ont été développés (cf. Fig. 2).

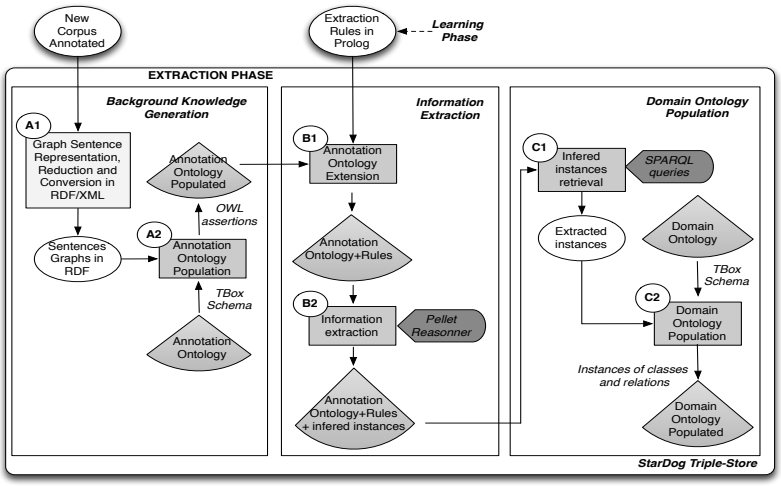

Figure 2. Mise en œuvre de la phase d'extraction dans OntoILPER.

Composant « Background Knowledge Generation ». Entièrement développé en Java, et assez similaire à celui de la phase d'apprentissage, il permet de constituer la BK à partir du corpus de documents annotés. Comme en phase d'apprentissage, il réalise un graphe à partir de toutes ces annotations, puis effectue une conversion directe de ce graphe en RDF/XML (A1) pour obtenir une représentation ontologique des annotations du corpus, et enfin peuple (A2) l'ontologie d'annotation par ces annotations de niveau lexical, syntaxique ou sémantique.

Composant « Information Extraction ». Ce composant permet tout d'abord l'intégration des règles d'extraction en SWRL dans l'ontologie d'annotation peuplée en RDF/XML (B1). Ensuite, en appelant du Triple-store le raisonneur Pellet, les règles d'extraction sont appliquées sur l'ontologie d'annotation (B2) et les instances de classes et de relations inférées sont ajoutées à l'ontologie d'annotation peuplée.

Composant « Domain Ontology Population ». Dans ce composant, grâce à des requêtes SPARQL, les instances d'entités et de relations extraites sont récupérées dans l'ontologie d'annotation (C1), elles sont utilisées pour peupler l'ontologie de domaine (C2), avec les classes et aux méthodes proposées par l'API du triple-store.

5. Expérimentations

Ces expérimentations ont été faites sur le corpus reACE 2004 sur un cluster avec des CPU à 6 cœurs et 12 Threads, fonctionnant à 2,5 Ghz, et sous Linux Debian 7 Wheezy. Cette section introduit ce corpus, les ontologies de domaine et d'annotation utilisées, ainsi que les règles d'extraction des entités et des relations obtenues lors de la phase d'apprentissage. Les différentes tâches du processus d'extraction réalisées

par les trois composants de la phase d'extraction sont décrites en détail, ainsi que les résultats obtenus selon une stratégie spécifique.

5.1 Le corpus ACE 2004

Pour évaluer les performances d'OntoILPER intégrant le triples-store Stardog, nous avons retenu la version révisée du corpus ACE 2004 (ACE, 2004), nommée reACE 2004, proposé par (Hachey et al, 2011). C'est une version normalisée de l'ACE originale 2004, qui constitue un corpus de référence pour l'extraction d'instances d'entités nommées et de relations entre ces dernières. Il comprend **348 documents** provenant de divers journaux, de dépêches d'information (NewsWire), et d'informations déjà diffusées, et il est composé de **1079 phrases.**

Types des relations	Sous-types de relations
Employ; Member; Subsidiary	*Employ-Staff* *Employ-Executive Member*
General-Affiliation	*Located; Citizen-Resident; Religion-Ethnic*
Part-Whole	*PartWhole; Subsidiary*
Personal- Social	*Business; Family*

Table 1. Distribution des relations dans le corpus reACE2004

Relations (TYPE.sous-type (arg1, arg2))	Exemples de phrases
PER-SOC.business (John,superiors)	*John'superiors*
EMP-ORG.employ-exec (investors, WallStreet)	*Investors on Wall Street*
EMP-ORG.employ-staff (ABC, John Martin)	*Here's ABC's John Martin*
GPL-AFF.citizen/resident (voters, Missouri)	*Some Missouri voters*

Table 2. Exemples de relations du corpus reACE2004

Ce corpus est annoté pour les entités nommées et les relations, et contient **1079** phrases avec **4352** entités de **4 types** : PER (person), ORG (organization), GPL (Geo-Political/Location) et FVW (Facility/Vehicle/Weapon). Il contient aussi comme représenté dans la Table 1, **4 types de relations binaires et leurs 9 sous-types**. La Table 2 représente quelques exemples de relations du corpus reACE 2004.

5.2 Ontologie de domaine

Cette ontologie, spécifique au corpus reACE traité, est représentée à la Fig. 3.

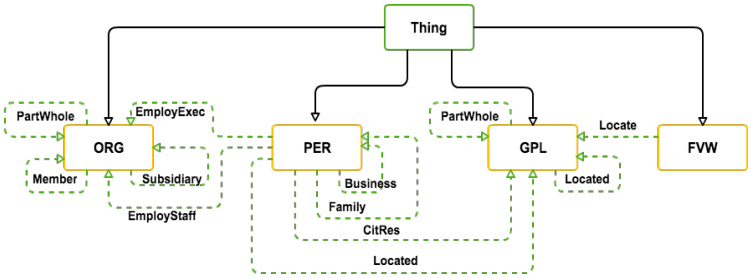

Figure 3. Ontologie de domaine pour le corpus reACE2004

Elle contient toutes les entités et relations à extraire sur ce corpus. Elle est composée de 4 classes qui représentent les quatre types des entités (ORG, PER, GPL et FVW) ; de 4 propriétés représentant les 4 types de relations, et de 9 sous-propriétés représentant les 9 sous-types de relations. Ces classes et ces sous-

propriétés sont les entités et les relations que nous devons extraire pour peupler l'ontologie de domaine par leurs instances.

5.3 L'ontologie d'annotation

Cette ontologie est une ontologie opérationnelle, apportant à OntoILPER plus de souplesse et de performance, n'est pas spécifique au corpus reACE : elle est générale et représente sous les annotations de toutes natures (lexicales, syntaxiques, sémantiques) apportées au corpus d'entrée. Cette ontologie d'annotation est composée de 16 classes et de 57 relations/propriétés. Ces propriétés expriment les prédicats Prolog définis dans la BK et forment le corps des règles d'extraction. La Fig. 4 illustre, par un modèle entité-association étendu cette ontologie d'annotation.

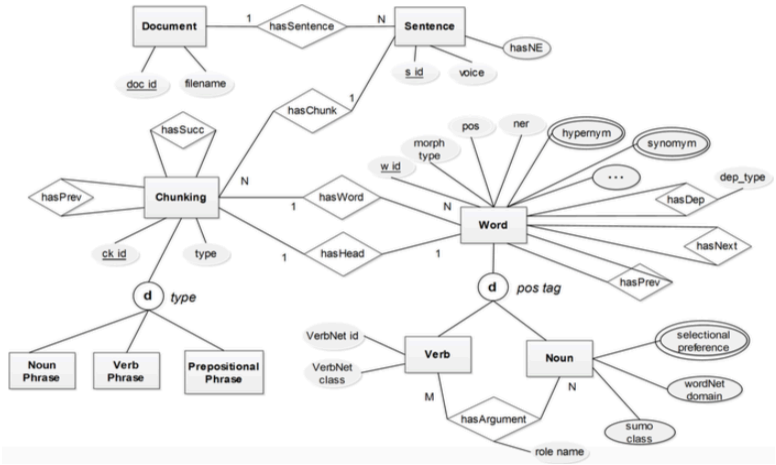

Figure 4. Ontologie d'annotation d'OntoILPER

5.4 Génération de la base de connaissances à priori (BK generation)

A partir du corpus reACE annoté, la génération de la BK, ou base de connaissance à priori est réalisée par les deux tâches A1 et A2.

La tâche A1 réalise tout d'abord une représentation sous forme de graphe des différentes annotations de toutes natures (tokens, sentences, POS tagging, Chunking (syntagmes), …) du corpus annoté, sur laquelle une réduction est effectuée pour des raisons d'optimisation. Ensuite cette tâche effectue une conversion directe entre les éléments graphiques et leurs éléments ontologiques correspondants dans l'ontologie d'annotation exprimée en RDF. Ainsi chaque nœud du graphe indiquant un mot ou token dans une phrase va être converti en une instance de la classe « Token » de l'ontologie d'annotation. Le processus de conversion est identique pour chaque document, phrase, et Chunks représenté comme nœuds dans le graphe. Ensuite, chaque arête du graphe reliant deux nœuds va être convertie en prédicats correspondants dans l'ontologie d'annotation.

Dans la tâche A2, l'ontologie d'annotation est peuplée de façon systématique par toutes les arêtes du graphe RDF précédent. Ce peuplement est assez lent (plusieurs

minutes), aussi est-il réalisé par incrément de 10% de corpus (tâche A2). Ainsi les premiers 10% du corpus sont traités et peuplent l'ontologie, soit des phrases de 1 à 107, ensuite les 10% suivant du corpus, soit des phrases de 108 à 216, peuplement qui se rajoute au peuplement précédent jusqu'à 100% du corpus.

5.5 Extraction d'information par application des règles induites

Cette extraction consiste à appliquer les règles induites dans la phase d'apprentissage à la BK présente dans l'ontologie d'annotation peuplée. Pour le corpus reACE, nous avons obtenu par la phase d'apprentissage de OntoILPER 4 règles pour extraire les instances d'entités et 220 règles pour les 9 sous-types de relations, qui se répartissent pour chaque sous-type de relation selon la table 3. Ces règles sont exprimées en langage Prolog. Ainsi les deux règles Prolog suivantes permet d'extraire la relation « *located* » selon différents prédicats d'annotation :

(1) located(A,B):- t_ner(A,loc),t_next(B,C),t_next(C,A),t_ner(B,loc)

Cette règle identifie les instances de la relation located(A, B) si les tokens A et B sont reconnus comme des instances de l'entité LOC, et s'il existe un token C entre eux.

(2) located(A,B):- t_next(A,C), t_next(C,D), t_next(D,B), t_isHeadNP(A), t_orth(B,upperinitial)

Cette règle identifie les instances de cette relation s'il existe deux tokens C et D entre les tokens A et B, et si A est un token de tête d'un syntagme nominal et B commence par une majuscule.

Relation	Nb. de règles induites
Business	11
CitRes	31
Familly	8
EmpExec	45
EmpStaff	5
Located	44
Member	21
PartWhole	28
Subsid	11

Table 3. Répartition des règles d'extraction de relations

Cette extraction est réalisée par les deux taches B1 et B2. Dans la tâche B1, l'ontologie d'annotation peuplée est chargée dans le triple-store, puis elle est enrichie par les règles d'extraction induites traduites en SWRL. Dans la tâche B2, en appelant le raisonneur Pellet à partir du triple-store, il s'agit d'appliquer les règles d'extraction sur l'ontologie d'annotation, qui est ensuite enrichie des instances de classes et de relations inférées par ces règles. Cette tâche d'extraction est très rapide : pour le corpus reACE complet : 5 secondes pour 1079 phrases.

5.6 Peuplement de l'ontologie de domaine

A partir de l'ontologie d'annotation augmentée des instances extraites, il s'agit de peupler l'ontologie de domaine. Les 2 tâches C1 puis C2 sont réalisées.

La tâche C1 récupère les instances extraites de l'ontologie d'annotation grâce à 13 requêtes SPARQL permettant de récupérer toutes les instances inférées de classes et de relations (9 requêtes pour récupérer les instances des relations et 4 pour récupérer les instances des entités). Voici un exemple de requête SPARQL pour récupérer les instances inférées de la relation « Located » :

PREFIX AnnotOnto : http://www.lsis.org/AnnotOnto/AnnotOnto#

SELECT ?s ?x WHERE { ?s AnnotOnto:Located ?s}

La tâche C2 réalise le peuplement proprement dit de l'ontologie de domaine par les instances d'entités et de relations extraites. Pour cela on utilise des classes pour la gestion des graphes spécifique au triple-store (*Graph* pour Stardog) proposées par son API, et la méthode « *Write* » pour écrire et stocker les graphes en RDF.

5.7 Stratégie utilisée

Pour réaliser les tâches d'extraction d'information et de population de l'ontologie de domaine de façon performante, nous avons obtenu les meilleurs temps en parallélisant l'exécution des composants d'extraction d'information et de peuplement de l'ontologie de domaine. Ainsi l'exécution de ces deux composants est réalisé en parallèle pour chaque entité et chaque relation à extraire, puis en effectuant une fusion des ontologies de domaines peuplées pour chacune d'elles, pour obtenir une ontologie de domaine complètement peuplée. Ceci en peuplant d'abord les instances d'entités puis les instances de relations.

Cela revient à considérer 10 ontologies d'annotation peuplées, dont une concerne les annotations relatives aux entités et 9 les annotations relatives à chacune des relations à extraire. Ces 10 ontologies d'annotation sont chacune enrichies par les règles d'extraction SWRL spécifiques à la relation à extraire ou aux entités à extraire. L'usage du raisonneur Pellet sur chacune de ces ontologies d'annotation enrichies génère les instances inférées et peuple chaque ontologie d'annotation. De chacune de ces ontologies d'annotation peuplées sont récupérées les instances inférées, qui seront utilisées ensuite pour peupler l'ontologie de domaine par les instances d'entités ou de relations spécifiques extraites. On obtient ainsi 10 ontologies de domaines partiellement peuplées. Enfin il s'agit de fusionner, grâce aux méthodes proposées par l'API du triple-store, ces 10 ontologies de domaines partiellement peuplées pour obtenir l'ontologie de domaine complètement peuplée.

Pour le peuplement de l'ontologie de domaine avec le triple-store StarDog, la Table 4 donne les temps en secondes obtenus. Ces temps sont les sommes de 3 temps : T1 = temps d'exécution de la requête SPARQL la plus lente pour la récupération des instances de chacune des relations (parallélisme) ; T2 = temps d'exécution des 4 requêtes SPARQL pour récupérer les instances d'entités ; et T3 = temps de la fusion de ces 10 ontologies.

Peuplement en %	T1(s)	T2(s)	T3(s)	Total
10%	12.13	27.74	26	39.87
50%	73.68	60.42	60	134.10
100%	180.01	101.33	100	281.34

Table 4. Temps de population de l'ontologie de domaine en secondes

6. Conclusion

Ce papier traite de l'opérationnalisation de la phase d'extraction, du système OntoILPER. Celle-ci utilise une ontologie d'annotation pour exploiter les annotations faites au préalable sur le corpus de documents, et le triple-store StarDog. Ce triple-store est utilisé pour appliquer sur cette ontologie les règles d'extraction en utilisant le raisonneur Pellet, ainsi que pour réaliser le peuplement de l'ontologie de

domaine par les instances extraites. En utilisant une stratégie de parallélisation/distribution des extractions/peuplements de chacune des relations et entités, les résultats obtenus sont intéressants. Cependant le peuplement de l'ontologie d'annotation doit être amélioré. On envisage pour cela utiliser des API spécifiques à StarDog plutôt que l'OWL API actuellement utilisé, voir d'utiliser aussi une stratégie de parallélisation/distribution.

Bibliographie

ACE 2004. http://www.itl.nist.gov/iad/mig/tests/ace/2004/doc/ace04-evalplan-v7.pdf

Bach, N. and Badaskar, S. (2007). *A Survey on Relation Extraction*. Language Technologies Institute, Carnegie Mellon University, 2007.

De Marneffe, M. C. & Manning, C. D. (2008). *Stanford Dependencies Manual*, 2008.

Giuliano, C., A. Lavelli & L. Romano. (2007). Relation Extraction and the Influence of Automatic NER, *ACM Transactions on Speech and Language Processing*, vol 5, no.1, ACM, 2007.

Hachey, B., Grover, C., & Tobin, R. (2012). Datasets for generic relation extraction. Natural Language Engineering, 18, 21–59.

Lavrac, N. & Dzeroski S. (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, New York, 1994.

Lima, R. (2014). OntoILPER: an Ontology and Inductive Logic Programming-based method to extract instances of Entities and Relations from Texts, Phd. Thesis. UFPE.

Lima R, Espinasse B, Freitas F (2013) Information Extraction from the Web: An Ontology–Based Method using Inductive Logic Programming. IEEE International Conference on Tools with Artificial Intelligence, IEEE-ICTAI 2013, Washington DC, USA.

Lima R, Espinasse B, Freitas F (2015) Relation Extraction from Texts with Symbolic Rules Induced by Inductive Logic Programming. IEEE International Conference on Tools with Artificial Intelligence, IEEE-ICTAI 2015, Vietri sul Mar, Italy.

Maynard, D., Li, Y., and Peters, W. (2008). NLP techniques for term extraction and ontology population. In Bridging the Gap between Text and Knowledge - Selected Contributions to Ontology Learning and Population from Text, IOS Press, 2008.

Muggleton, S. (1991). Inductive Logic Programming. New Generation Computing 8 (4): 29.

Nédellec, C. & Nazarenko, A. (2005). Ontologies and Information Extraction. http://arxiv.org/abs/cs.AI/0609137

Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A. & Zavitsanos, E. (2011). Ontology Population and Enrichment: State of the Art, in G. Paliouras et al. (Eds.): *Multimedia Information Extraction*, LNAI 6050 (pp. 134–166), 2011.

Popov B., A. Kiryakov, D. Ognyanoff, D. Manov and A. Kirilov, KIM - a semantic platform for information extraction and retrieval, Journal of Natural Language Engineering 10(3-4) (2004) 375-392, 2004.

Santos, J. (2010). Efficient Learning and Evaluation of Complex Concepts in Inductive Logic Programming, Ph.D. Thesis, Imperial College, 2010.

Wimalasuriya, D. C., Dou, D. (2010). Components for Information Extraction: Ontology-Based Information Extractors and Generic Platforms. *CIKM'10*, October 26–30, 2010, Toronto, Ontario, Canada.

Wu, F., and Weld, D. (2008) Automatically refining the Wikipedia infobox ontology. In Proceedings of the 17th International Conference on World Wide Web, New York, USA, 2008.