B. Espinasse, R. Lima, F. Freitas (2016), « Extraction automatique d'entités et de relations par ontologies et programmation logique inductive », in: Revue d'Intelligence Artificielle (RIA), Vol. 30 (n° 6/2016), dec 2016

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

Bernard Espinasse[1], Rinaldo Lima[2], Fred Freitas[3]

1. Aix-Marseille Université, LSIS UMR CNRS 6168
Domaine Universitaire de St Jerôme, F-13997, Marseille cedex 20, France
bernard.espinasse@lsis.org
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
3. Universidade Federal de Pernambuco, CIn - UFPE
Centro de Informática, Cx Postal 7851, 50372-970, Recife/PE, Brasil
fred@cin.ufpe.br

RESUME. *Face à la quantité croissante d'informations disponibles tant sur le web que dans les bibliothèques numériques, le développement de systèmes d'extraction d'information (EI) automatique, à la fois efficaces, robustes et adaptatifs, constitue un grand défi. Dans l'extraction d'information la reconnaissance d'entités nommées (REN) vise à extraire des instances nommées dans le texte, par exemple des noms de personnes, de lieux, tandis que l'extraction de relation (ER) consiste à extraire des relations entre ces entités nommées. Pour ces deux tâches, la plupart des méthodes d'apprentissage automatiques supervisées utilisées sont essentiellement statistiques avec de très bons résultats pour la REN, moins bons pour l'ER. Ces méthodes statistiques utilisent généralement un espace d'hypothèses propositionnelles pour la représentation des exemples (attribut-valeur) présentant certaines limitations notamment dans l'extraction de relations complexes exigeant des informations contextuelles sur les instances concernées, voire des ressources sémantiques. Dans cet article, nous présentons le système OntoILPER, permettant d'extraire automatiquement des instances d'entités et de relations de documents textuels en langue anglaise. Ce système tire profit des ressources sémantiques d'une ontologie de domaine, mais aussi d'un espace d'hypothèses relationnel plus riche, pour représenter les exemples et induire des règles d'extraction symboliques par programmation logique inductive (PLI), une technique d'apprentissage symbolique. Plusieurs expérimentations sur le corpus de référence TREC permettent de comparer ses performances encourageantes avec celles de systèmes d'extraction statistiques.*

ABSTRACT. *Faced with the growing amount of information available both on the Web and in digital libraries, the development of automatic Information Extraction (IE) systems, both effective, robust and adaptive, is a big challenge. In IE domain, Named Entity Recognition (NER) and Relation Extraction (RE) are two important tasks. The former aims at finding*

named instances, as people´s names, locations, among others, whereas the latter consists detecting and characterizing relations among such named entities in text. Most of the state-of-the-art supervised learning methods for NER and RE relies on statistical machine learning techniques with higher accurate results for NER than RE. These statistical machine learning techniques typically uses a propositional hypothesis space for representing examples, i.e., an attribute-value representation. Such representation presents some limitations particularly to the extraction of complex relations, which demand more semantic resources, and mainly contextual information about the involving instances. In this paper, we present an IE system, named OntoILPER, permitting to extract both entity and relation instances from textual document in english. This system, not only benefits from a domain ontology as semantic resource, but also takes advantage of a higher expressive relational hypothesis space for representing examples whose structure is relevant to the task at hand. OntoILPER induces extraction rules that subsume examples of entities and relation instances from a specific graph-based model of sentence representation. Moreover, the system enables the application of domain ontologies and further ground knowledge in the form of relational features. In addition, this paper presents several experiments with OntoILPER on NER and RE using the TREC reference corpus, and compare these results to other state-of-the-art IE systems.

MOTS-CLES : *extraction d'entités et de relations, apprentissage symbolique, extraction d'information à base d'ontologies, programmation logique inductive, peuplement d'ontologie.*

KEYWORDS: *entity and relation extraction, symbolic machine learning, ontology-based information extraction, inductive logic programming, ontology population.*

DOI:10.3166/RIA.30.1-38 © Lavoisier 2016

1. Introduction

Face à la quantité croissante d'informations disponibles tant sur le web que dans les bibliothèques numériques, le développement de systèmes d'extraction d'information (EI) automatique, à la fois précis, efficaces, robustes et adaptatifs, constitue un grand défi. Le développement de tels systèmes passe nécessairement par l'usage de techniques d'apprentissage machine (apprentissage artificiel), mais aussi par l'exploitation de ressources sémantiques notamment liées au domaine d'application.

L'EI consiste à reconnaître et à extraire certains types d'informations à partir de textes. Les deux sous-tâches majeures 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. L'apprentissage automatique supervisé est largement utilisé pour ces deux tâches, principalement en utilisant des méthodes statistiques. Pour la REN, la performance de tels systèmes utilisant ces méthodes statistiques est autour de 90 %. Cependant, l'ER entre entités est une tâche plus difficile et les performances obtenues avec ces méthodes sont nettement inférieures (Giuliano *et al.,* 2007).

Les *méthodes statistiques supervisées* s'appuient principalement sur la distribution des données, ce qui peut expliquer, dans une certaine mesure, pourquoi

elles fournissent des explications pauvres sur leurs résultats. En outre, comme l'a souligné (Bach et Badaskar, 2007), elles font face à des difficultés significatives dans l'ER, impliquant notamment plus de deux entités. Une alternative à l'approche supervisée statistique est *l'approche supervisée symboliqu*e. Dans cette approche, des exemples d'entités ou de relations cibles sont utilisées pour induire des règles d'extraction basées sur des prédicats logiques, en exploitant les informations structurelles des exemples (Muggleton *et al.,* 2009).

Cet article s'intéresse au développement de systèmes d'EI selon une approche originale, consistant à combiner d'une part, l'usage de ressources sémantiques dans le processus d'extraction d'information, et par là même le domaine des ontologies et du web sémantique, et, d'autre part, le domaine de l'apprentissage symbolique ou « relationnel », avec ici l'usage de la programmation logique inductive (PLI), une technique d'apprentissage symbolique (Lavrac et Dzeroski, 1994). Ces deux domaines, qui se situent à un même niveau sémantique de représentation (logique du premier ordre), ont suivi des développements spécifiques. Cependant, ils nous semblent actuellement pouvoir se rencontrer et se combiner afin de proposer des solutions à des problèmes actuels importants comme celui de l'EI.

Plus précisément cet article présente un système d'extraction d'information, nommé OntoILPER (Lima *et al.,* 2013, 2014, 2015), permettant d'extraire de façon automatique des instances d'entités et de relations à partir d'un document textuel exprimé en langue anglaise. Ce système tire profit des ressources sémantiques d'une ontologie de domaine pour représenter les exemples et induire des règles d'extraction symboliques par PLI.

Pour induire ces règles d'extraction symboliques, ce système utilise un espace d'hypothèses relationnelles (prédicats binaires) plus expressif que l'espace d'hypothèses propositionnelles utilisé par les méthodes statistiques pour la représentation des exemples (attribut-valeur). Ainsi, pour l'extraction de relations, plus délicate que celle d'entités, OntoILPER s'appuie sur une représentation relationnelle de la phrase du texte sous forme de graphe. Cette représentation peut incorporer des éléments issus de l'ontologie de domaine, ainsi que d'autres connaissances de base, sous la forme d'attributs (ou caractéristiques). Ces derniers peuvent améliorer significativement les performances du processus d'extraction.

La section 2 introduit différents concepts fondamentaux abordés dans cet article, notamment relatifs à l'extraction d'information (EI) avec la reconnaissance d'entités nommées (REN) et l'extraction de relations (ER), mais aussi relatifs à l'apprentissage « relationnel » et à la programmation logique inductive (PLI), aux ontologies, et à leurs usages dans l'extraction d'information automatique à partir de textes. La section 3 introduit la méthode d'extraction utilisée dans OntoILPER, une méthode basée sur l'usage de la PLI et d'une ontologie de domaine pour extraire des instances d'entités et de relations à partir de données textuelles. La section 4 présente l'architecture et les différents composants logiciels du système OntoILPER mettant en œuvre cette méthode. La section 5 présente et interprète les résultats de plusieurs expérimentations conduites avec OntoILPER pour différents modèles d'extraction, sur le corpus de référence TREC en extraction d'entités et de relations.

La section 6 compare, sur ce même corpus, les résultats d'OntoILPER avec ceux obtenus avec deux autres systèmes d'extraction utilisant des méthodes statistiques. Enfin, la section 7 présente plusieurs perspectives de recherche en conclusion.

2. Extraction d'information, ontologies et programmation logique inductive

Dans cette section, nous présentons brièvement les domaines abordés dans cette recherche. Tout d'abord ceux de la reconnaissance d'entité nommées (REN) et l'extraction de relations (ER) en extraction d'information (EI). Ensuite les domaines de la programmation logique inductive (PLI) relevant de l'apprentissage « relationnel », et celui des ontologies, avec pour chacun de ces domaines, leurs usages en extraction d'information.

2.1. Extraction d'information

Les deux tâches fondamentales de l'extraction d'informations sont : la reconnaissance d'entités nommées (REN) et l'extraction de relations (ER) entre ces entités nommées. Par exemple soit la phrase *Mary Kandel at the Newsdesk CNNfn in New York*. Dans cette phrase ont va chercher à extraire automatiquement par exemple les entités *Mary Kandel*, et *New York*. On cherchera ensuite à extraire des relations entre ces entités, par exemple la relation *located_in* entre les entités *Mary Kandel* et *New York*.

2.1.1. La reconnaissance d'entités nommées

L'objectif de la reconnaissance d'entités nommées (REN) est d'identifier les entités nommées d'un texte en langage naturel et de les classer en un ensemble de types prédéfinis tels que « personne », « organisation », « localisation ». La REN est la tâche la plus basique de l'EI, elle nécessite la mise en œuvre de méthodes de désambiguïsation, pouvant combiner des approches symboliques et distributionnelles (Ehrmann, 2008). L'extraction de structures plus complexes telles que de relations (ER) et d'événements, considère la REN comme une étape préliminaire.

Dans la REN, l'identification des entités nommées peut être considérée comme un problème d'étiquetage séquentiel (*Sequential Labelling*), dans lequel des séquences complètes de mots ou de « tokens » sont considérées plutôt que des mots isolés comme cela se fait dans la classification. Dans les méthodes supervisées utilisées en REN, dites méthodes d'étiquetage séquentiel (*Sequential Labelling Methods*), un document est considéré comme un séquence de tokens, et une séquence d'étiquettes est attribuée à chaque token pour indiquer la propriété du token (Tang *et al.,* 2007). Une des méthodes les plus utilisées en matière d'étiquetage séquentiel est le modèle de Markov caché (*Hidden Markov Model* - HMM) (Rabiner, 1989), consistant en un automate à états finis avec des transitions d'état stochastiques et des émissions de symboles. Un autre type de méthodes aussi utilisé en REN sont les CRF (*Conditional Random Field*), méthodes statistiques

aussi basées sur les séquences, qui ont souvent des performances supérieures aux HMM. Ces deux types de méthodes ont été notamment utilisés avec succès dans le domaine biomédical (Kinoshita et Cohen, 2005 ; Shen *et al.,* 2003 ; Okanohara *et al.,* 2006 ; Settles, 2004).

Ces méthodes d'étiquetage séquentiel présentent deux principaux inconvénients. Le premier est qu'elles nécessitent une grande quantité de données d'apprentissage, et plus on a de données d'apprentissage, plus les résultats sont bons. Le second est que le modèle sous-jacent à ces méthodes repose sur un modèle plat ne prenant pas en compte d'information structurelle. En conséquence, ces méthodes ne sont indiquées que lorsque les séquences étiquetées ne sont pas emboîtées et qu'il n'y a pas de relation explicite entre les séquences.

2.1.2. Extraction de relations

L'extraction de relations (ER) consiste à *détecter* et *caractériser* dans le texte des relations sémantiques entre entités. Dans la détection, il s'agit de déterminer s'il y a une relation entre deux entités, alors que dans la caractérisation, il s'agit d'assigner une étiquette de relation type à une instance de relation particulière (problème de classification). La plupart des travaux se concentre sur l'extraction de relations binaires, relations entre les deux entités, par exemple des relations *physiques* (une entité est physiquement près d'une autre entité), ou *d'affiliation* (une personne est employée par une organisation).

Différentes techniques d'apprentissage automatique, supervisées, semi-supervisées ou non supervisées sont utilisées en ER. Les plus utilisées sont les techniques supervisées, soit basées sur les caractéristiques (*Features-based*), soit basées sur les noyaux (*Kernel-based*).

Les techniques basées sur les *caractéristiques* (*Features-based*) reposent sur des modèles de classification construits en transformant des exemples de relations en vecteurs numériques correspondants représentant plusieurs types de caractéristiques. Elles appliquent ensuite une technique d'apprentissage machine, comme SVM (*Support Vector Machine)*, pour détecter et classer les exemples de relations dans des types prédéfinis de relations. Ces techniques donnent de bons résultats en employant un grand nombre de caractéristiques linguistiques dérivées de connaissances lexicales, d'informations relatives aux entités, d'arbres d'analyse syntaxique et de dépendances, et d'informations sémantiques (Kambhatla, 2004 ; (Zhou *et al.,* 2005 ; Jiang et Zhai, 2007). Cependant, ces techniques ont du mal à capturer efficacement l'information structurée d'arbres d'analyse (Zhou *et al.,* 2007), ce qui est essentiel pour une ER performante. Roth et Yih (2007) ont proposé un système d'extraction d'entités et de relations basé sur une inférence « globale », dans laquelle les facteurs prédictifs identifiant les entités et les relations, sont appris de l'information locale dans la phrase. Les contraintes induites par les dépendances entre les types d'entités et les relations constituent une structure relationnelle sur les résultats des prédicteurs et sont utilisées pour faire une inférence globale.

Les techniques basées sur les *noyaux* (*Kernel-based*) reposent sur des fonctions à noyaux définissant le produit scalaire (*inner product)* de deux exemples observés,

représentés dans un espace vectoriel sous-jacent. Les fonctions noyau sont souvent considérées comme une mesure de similarité entre deux vecteurs d'entrée qui représentent des exemples dans un espace transformé en utilisant l'ensemble d'attributs initial. L'avantage majeur de l'utilisation de noyaux est que les cas observés, n'ont pas besoin d'être explicitement mis en correspondance sur l'espace vectoriel sous-jacent, afin que leurs produits scalaires définis par le noyau puissent être calculés (Jiang, 2012). Deux types de noyaux sont principalement utilisés en ER (Jiang, 2012) : les noyaux *à base d'arbres* (*Tree-based kernels*) utilisés dans (Culotta et Sorensen, 2004), et les noyaux *composites* (*Composite Kernels*) utilisés dans (Zhao et Grishman, 2005) et (Choi *et al.,* 2009). Ces méthodes à noyaux sont généralement construites par des techniques d'apprentissage, notamment SVM *(Support Vector Machine).*

2.2. Programmation logique inductive et extraction d'information

L'apprentissage automatique du premier ordre, appelé aussi « relationnel », trouve des modèles à partir de données stockées dans des structures de données complexes (graphes ou plusieurs tables). Ces modèles sont utilisés pour classer de nouveaux exemples en positif ou en négatif.

2.2.1. La programmation logique inductive

La programmation logique inductive (PLI) est une technique relevant de l'apprentissage relationnel qui utilise les *clauses de premier ordre* comme langage de représentation uniforme pour les *exemples*, les *connaissances de base* ou *BK (Background Knowledge)*, et des *hypothèses* (Lavrac et Dzeroski, 1994 ; De Reaedt, 2010). La PLI permet de traiter des données structurées et exprimer des connaissances dans le puissant langage qu'est la programmation logique, tant pour décrire des hypothèses que des modèles induits. Elle permet aussi le développement d'un apprentissage tenant compte de connaissances expertes disponibles afin d'augmenter l'expressivité de l'espace d'hypothèses. Muggleton (1991), un des pères de la PLI, décrit formellement la PLI ainsi :

Etant donné :

– un ensemble fini *E d'exemples*, divisé en un ensemble d'exemples positif E^+ et négatif E^-, tous deux exprimés par des ensembles non vides de *clauses sans variables (ground facts)*, et

– des *connaissances de base (BK - Background Knowledge),* consistant en un ensemble fini de clause de Horn[1] extensionnelles (sans variables) or intentionnelles (avec variables),

le *but* est alors d'induire une *hypothèse correcte H* (*ou théorie H*) composée des clauses de premier ordre telles que soient satisfaites les deux conditions :

- $\forall e \in E^+$: BK $\wedge$ $H \models e$ (*H* est *complète*), et

1. Les clauses de Horn sont des clauses de premier ordre ayant au plus un littéral positif.

- $\forall e \in E^-$: BK $\wedge$ $H \not\models e$ (H est *consistante*).

Dans la pratique, il n'est pas toujours possible de trouver une hypothèse correcte qui satisfasse strictement les deux conditions ci-dessus, aussi ces deux conditions sur la BK doivent être assouplies.

La PLI peut être mise en œuvre par la recherche, dans un espace d'hypothèses partiellement ordonné (Mitchel, 1982), des états correspondants aux descriptions de concept (hypothèse), le but est alors de trouver un ou plusieurs états satisfaisant certains critères de qualité. Structurer l'espace de recherche consiste à trier les hypothèses selon un ordre partiel permettant de déterminer, entre deux clauses, qu'elle est la clause la plus générale ou la plus spécifique. La plupart des stratégies de tri utilisées par les systèmes PLI sont basées sur la θ-subsomption[2], stipulant qu'étant données deux clauses C et D, C θ-subsume D, s'il existe une substitution θ, telle que $C\theta \subseteq D$. En d'autres termes, C est une généralisation de D, et D est une spécialisation de C, sous θ-subsomption (Plotkin, 1971).

Plusieurs modèles d'implémentation (ou d'algorithmes) de PLI basés sur une recherche descendante ont été proposés. Les plus importants étant *Golem* (Muggleton et Feng, 1990) et *Progol* (Muggleton, 1995). Plus récemment, le modèle *ProGolem* (Muggleton *et al.,* 2009), basé sur une approche ascendante a été proposé. Il combine la stratégie de construction de la « botton-clause » de *Progol* avec la stratégie de contrôle de *Golem* en utilisant des généralisations relatives minimales et asymétriques *(Asymmetric Relative Minimal Generalisations).* En raison de l'efficacité du modèle *ProGolem*, comparativement à d'autres systèmes génériques de PLI comme le système *ALEPH*[3], nous avons retenu le système *GILPS* (Santos, 2010) implémentant *ProGolem*, pour développer le module d'apprentissage d'OntoILPER.

Enfin, notons toute l'importance de la notion de *biais* en PLI. Un biais d'apprentissage est une méthode qui permet de sélectionner une hypothèse plutôt qu'une autre, en se basant sur des critères autres que leur cohérence vis-à-vis des exemples d'apprentissage. De nombreux types de biais existent (Nédellec *et al,* 1996), les principaux types étant les *biais déclaratifs* et les *biais de préférence. Les biais déclaratifs* sont un moyen de restreindre l'espace de recherche, ils définissent notamment les clauses dont la syntaxe est acceptable, par exemple pour un utilisateur. Les *biais de préférence* sont des heuristiques utilisés pour évaluer les clauses afin de les comparer entre elles. Les modèles *Golem*, *Progol* et *Progolem* définissent leurs propres biais.

---

2. Une substitution θ = {V1/t1, V2/t2, ..., Vn/tn} consiste à assigner des termes ti à des variables Vi.

3. A. Srinivasan. The Aleph Manual.
http://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html.

2.2.2. Extraction d'information à base de PLI

La PLI a été appliquée dans l'EI, couplée à des méthodes statistiques ou seule. Dans (Ramakrishnan *et al.,* 2008), la PLI construit des caractéristiques (*features*) pour l'EI en identifiant un grand nombre de caractéristiques pertinentes qui sont utilisées comme entrée d'un classificateur SVM d'entités nommées. Ces modèles SVM construits avec ces caractéristiques issues de la PLI obtiennent de meilleurs résultats que s'ils étaient construits avec des caractéristiques définies manuellement.

Patel *et al.* (2010) utilisent la PLI pour réduire les efforts d'un développeur de règles pour la REN. Ils ont utilisé deux techniques de PLI pour construire des règles d'extraction des instances de plusieurs classes d'entités nommées. Ils ont constaté que, par rapport à des règles développées par un expert linguiste humain, le temps de développement en utilisant la PLI a été réduit par un facteur de 240. De plus, par rapport à une approche avec des règles développées manuellement, les auteurs avancent qu'une méthode basée sur la PLI fournit une vue complète et cohérente de tous les patrons d'extraction significatifs dans les données au niveau d'abstraction spécifié par l'ingénieur de la connaissance. Ainsi, une approche fondée sur la PLI permet la découverte de règles qui pourraient être ignorées par l'expert du domaine, ce qui rend également possible le développement de telles règles avec des ensembles de données conséquents.

Nédellec *et al,* (2008) proposent le système Alvis permettant d'extraire les entités et les relations de corpus biologiques avec l'usage de PLI. Alvis fournit une analyse sémantique basée sur la plate-forme de traitement des langages naturels (TAL) Ogmios (Nazarenko *et al.,* 2006), qui effectue plusieurs tâches : reconnaissance d'entités biologiques, *POS tagging*, analyse terminologique assistée de dictionnaires terminologiques, analyse syntaxique, et de correspondance (*mapping*) sémantique à des ontologies du domaine biologique. Dans Alvis, une fois les unités sémantiques du texte identifiées, elles sont typées avec des concepts à grains fins qui sont associés par des relations spécifiques au domaine de l'ontologie. La composante apprentissage machine d'Alvis est basée sur la méthode *LP-Propal* utilisant l'algorithme supervisé PLI Propal (Alphonse et Rouveirol, 2000).

Horvath *et al,* (2009) examinent les deux arbres de dépendances que les structures relationnelles composées d'un seul prédicat binaire qui représente les bords de ces arbres. Dans leur travail, ils utilisent la plate-forme GATE et l'analyseur de Stanford pour le prétraitement du texte. Ils ont utilisé aussi WordNet comme ressource sémantique pour obtenir les relations hyperonymes. De plus, les auteurs supposent un ordre partiel sur l'ensemble des prédicats unaires qui sont définis par une hiérarchie entre les mots, par exemple, le prédicat unaire Personne(X) est plus général que le prédicat Physicien(X), le dernier étant dérivé de WordNet. En appliquant l'opérateur de généralisation, *Least General Generalization* (LGG) de Plotkin (Plotkin, 1971), ils génèrent un ensemble de règles exprimées en clauses de Horn non récursives satisfaisant certains critères de cohérence, par exemple, toutes les règles doivent couvrir un nombre minimum d'exemples positifs, tout en tenant compte d'un nombre d'exemples négatifs en même temps. Ils utilisent ensuite ces règles pour générer un vecteur binaire d'attributs pour chaque exemple,

et les vecteurs résultant pour l'apprentissage d'un classifieur SVM afin de séparer des exemples positifs à partir des exemples négatifs.

Seneviratne et Ranasinghe (2011) ont pour leur part proposé un système multi-agent d'extraction d'information qui s'appuie sur la PLI pour l'apprentissage de règles d'extraction de relations. Les auteurs exploitent la capacité d'apprentissage des agents pour former un agent apprenant des règles d'extraction à partir de la structure syntaxique de la phrase en langage naturel. Les dépendances typées des constituants syntaxiques de la phrase fournissent les informations de base pour l'espace de recherche, c'est à dire les constituants de base pour l'induction de règles. Dans ce système multi-agent, un agent PLI est responsable du processus d'apprentissage de la règle, tandis qu'un autre agent utilise les règles apprises afin d'identifier de nouvelles relations, ainsi que pour extraire des instances des relations prédéfinies. Toutes les relations dérivées sont exprimées en prédicats de deux arguments qui sont des entités. Les auteurs ont évalué leur système en l'appliquant sur seulement 13 pages web Wikipédia du domaine des oiseaux.

Smole *et al,* (2012) ont proposé un système de ER basé sur la PLI et capable d'apprendre les règles d'identification et d'extraction d'informations à partir de définitions d'entités géographiques dans des textes en langue Slovène. Les auteurs se concentrent sur l'extraction de cinq des plus fréquentes relations/propriétés, à savoir *isA, islocated, hasPurpose, isResultOf* et *hasParts,* présentes dans les définitions de 1 308 entités spatiales. Pour cela, ils ont retenu le système de PLI *Progol* (Muggleton, 1995), qui induit des clauses de Horn. Leur composant de traitement de langue est basé sur un POS tagger pour le Slovene, suivit par un outil de « chunking » développé spécifiquement en Slovène. Enfin, ils utilisent la version du texte annoté pour attribuer manuellement relations ou propriétés à tous les « chunks » des 1 308 définitions sélectionnées.

2.3. Ontologies et extraction d'information

Dans l'une des définitions les plus cités du concept d'ontologie, Gruber affirme qu'*une ontologie est une spécification explicite d'une conceptualisation* (Gruber, 1993). Ces ontologies sont de plus en plus utilisées dans les processus d'extraction d'information, d'où l'émergence des systèmes OBIES pour *Ontology Based Information Extraction Systems* (Wimalasuriya et Dou, 2009).

2.3.1. Ontologies

Les ontologies sont des représentations de connaissances formalisées pouvant être traitées par un ordinateur dans un grand nombre de tâches, notamment la communication et l'interopérabilité (en utilisant l'ontologie comme un vocabulaire commun), la communication et le raisonnement d'agents intelligents. En termes pratiques, les ontologies englobent des définitions de concepts, de propriétés, de relations, de contraintes, d'axiomes et d'instances sur un certain domaine ou un univers du discours. En outre, ils permettent la réutilisation de connaissances de

domaine, ce qui rend les hypothèses de domaine explicites, séparant ainsi la connaissance du domaine de l'opérationnel.

La spécification d'une telle conceptualisation nécessite d'utiliser des langages spécifiques. Le langage OWL (Hitzler *et al.,* 2009), issu de travaux de recherche du W3C dans le domaine du web sémantique, est actuellement le langage d'expression d'ontologies le plus répandu. Ce langage est basé sur les logiques de description (Baader *et al.,* 2008), dont la sémantique est formellement définie. Enfin les systèmes OBIES tirent profit des divers outils développés autour de OWL et du web sémantique, notamment les environnements d'éditions d'ontologies comme Protégé, les raisonneurs automatiques comme Pellet et les langages de règles comme SWRL.

2.3.2. Extraction d'information basée sur les ontologies

Face au challenge d'une extraction d'information de plus en plus délicate, notamment l'extraction de relations (relations binaires) entre entités ou d'événements entre n entités (relations n-aires), les systèmes d'EI automatiques se doivent d'exploiter de plus en plus de ressources sémantiques disponibles, notamment des ontologies (Nédellec et Nazarenko, 2005). L'intérêt de l'utilisation d'ontologies dans le processus d'EI a été démontrée par plusieurs chercheurs notamment (Nedellec et Nazarenko, 2005 ; Karkalesis *et al,* 2011 ; Wimalasuriya et Dou, 2009) : les ontologies non seulement capturent les connaissances sur un domaine considéré, elles peuvent aussi être utilisées dans des applications pour traiter du contenu informationnel, ainsi que pour raisonner sur ce contenu.

Ainsi l'EI basée sur des ontologies (*Ontology-Based Information Extraction - OBIE*) a récemment émergée comme un sous-domaine de EI. L'OBIE peut être définie comme le processus d'identification dans le texte, de concepts, de propriétés et de relations exprimés dans une ontologie (Saggion *et al.,* 2007).

Les systèmes d'IE exploitant des ontologies ou OBIES (*Ontology Based Information Extraction System*) tels que définis par (Wimalasuriya et Dou, 2009), utilisent une ontologie de domaine pour le domaine d'extraction visé, ainsi qu'une technique d'extraction d'information spécifique pour extraire dans le texte, les individus des classes et les valeurs des propriétés de cette ontologie. Dans les systèmes OBIES les ontologies sont utilisées tant dans le processus d'EI même, que dans sa sortie assimilée au peuplement d'une ontologie.

L'usage d'ontologie concoure à rendre les systèmes d'EI plus adaptatifs du fait qu'une partie des connaissances qu'ils utilisent sont externes (dans l'ontologie), et qu'un même système d'EI peut ainsi plus facilement être adapté à un nouveau domaine d'extraction en exploitant une nouvelle ontologie liée à ce nouveau domaine. Pour être plus rapidement développés et plus facilement adaptables à d'autres domaines d'application, de tels systèmes d'EI doivent aussi utiliser des techniques d'apprentissage automatique.

2.4. Conclusion

Dans le développent de systèmes d'extraction automatique d'information, les méthodes classiques d'extraction d'information supervisées statistiques, soit basées sur les *caractéristiques* (*Features-based*), soit sur les *noyaux* (*Kernels-based*), sont largement utilisées. Ces méthodes sont utilisées avec de très bonnes performances en reconnaissance d'entités nommées (REN), et en extraction de relations (ER) avec des performances moindres. Ces méthodes s'appuient principalement sur l'aspect distributionnel des données, et fournissent des explications pauvres sur leurs résultats. Selon une étude récente (Choi *et al.,* 2013), ces méthodes présentent plusieurs limitations. La première concerne plus particulièrement les *méthodes basées sur les caractéristiques (Features-based)*. Celles-ci nécessitent beaucoup d'efforts pour la sélection et l'extraction de caractéristiques, en effet, il leur est difficile de capturer efficacement l'information structurée sur des arbres (Zhou et al 2005), ce qui est essentiel pour améliorer la performance de l'extraction de relations. La deuxième limite concerne les *méthodes à base d'arbres de noyau (Tree-based Kernels)* qui font un usage limité de l'information structurelle, très utile pour l'extraction de relations. La troisième limite concerne les *fonctions noyaux basées sur les arbres de dépendances* qui nécessitent un calcul lourd de similarité. Ainsi comme le souligne (Bach et Badaskar, 2007), ces méthodes statistiques, très efficaces dans la reconnaissance d'entités nommées, le sont moins dans l'extraction de relations.

Une alternative à ces méthodes d'apprentissage supervisées statistiques et leur limites, notamment dans l'extraction de relations, est d'adopter une approche relevant de l'apprentissage supervisé symbolique ou « relationnel », dans laquelle les exemples des entités ou des relations cibles sont utilisées comme entrées dans la construction de prédicats logiques comme règles d'extraction (Muggleton *et al.,* 2009). En d'autres termes dans cette extraction d'information supervisée, il s'agit d'opter pour des techniques d'apprentissage symboliques, exploitant les informations structurelles des exemples, afin d'induire des règles d'extraction symboliques. Contrairement aux méthodes statistiques, ces méthodes symboliques, comme la PLI, utilisent une représentation symbolique et déclarative, et les hypothèses produites sont alors compréhensibles et interprétables par l'homme. La *BK (Background Knowledge) ou connaissances de base* et les *exemples* sont exprimés à un *même niveau symbolique*. Ceci permet d'enrichir le processus d'extraction d'information en intégrant des ressources sémantiques supplémentaires, tels que thésaurus ou ontologies, ceci sans modifier le noyau du processus d'extraction d'information. Ainsi toute contrainte au problème d'extraction peut être exprimée en définissant des prédicats auxiliaires, pouvant être fournis par l'utilisateur, et complétant la BK. Enfin la PLI surmonte les limitations de la représentation valeur-attribut (propositionnelle) des systèmes d'apprentissage statistiques, en utilisant une représentation relationnelle avec des prédicats binaires.

Pour conclure, le développement de systèmes d'EI en combinant l'usage de ressources sémantiques dans le processus d'extraction d'information, et l'apprentissage symbolique ou « relationnel », avec l'usage de la programmation

logique inductive (PLI) apparaît judicieux. Dans la suite de l'article nous présentons le système OntoILPER, un système d'EI permettant d'extraire de façon automatique des instances d'entités et de relations d'un document textuel en langue anglaise. Ce système tire profit des ressources sémantiques d'une ontologie de domaine pour représenter les exemples, puis induit des règles d'extraction symboliques par PLI, effectue ensuite l'extraction d'instances d'entités et de relations en appliquant ces règles apprises, et enfin peuple de ces instances extraites l'ontologie de domaine. Les sections suivantes présentent les fondements méthodologiques de ce système et ensuite son architecture et ses composants logiciels.

3. Une méthode d'extraction d'information symbolique

L'extraction d'instance d'entités et de relations entre ces entités dans le système OntoILPER extraction est réalisée par l'application de règles symboliques induites par un apprentissage symbolique supervisé. L'apprentissage de ces règles à partir d'un corpus de textes annotés, utilise la PLI et exploite une ontologie de domaine. Une fois l'extraction d'information réalisée, cette ontologie de domaine est peuplée par ces instances d'entités et de relations extraites.

Le processus général d'extraction réalisé par OntoILPER, illustré dans la figure 1, s'articule en quatre grandes étapes :

1. Dans l'étape de *Traitement du langage naturel et d'annotation du texte* du processus, les documents textuels annotés (en XML) sont traités et annotés par des outils du TAL (annotation morphosyntaxique et sémantique).
2. Ensuite à partir de ce corpus de textes annotés en XML, OntoILPER génère *la BK (Background Knowledge – connaissances de base)*, selon un modèle de phrase spécifique, et en exploitant l'ontologie de domaine. Ainsi les axiomes de sa TBox et les assertions de sa ABox fournissent de précieuses annotations complémentaires.
3. Ensuite, à partir de cette BK, dans l'étape *d'Apprentissage des règles d'extraction* utilisant le système général GILPS de PLI, OntoILPER induit les règles d'extraction symboliques des instances d'entités et de relations exprimées comme un ensemble de programmes logiques (ou une théorie).
4. Enfin, l'étape *d'Extraction des instances et peuplement de l'ontologie* applique ces règles d'extraction induites dans l'étape précédente sur des exemples de textes annotés, les instances de classes et de relations extraites sont ensuite utilisées pour peupler l'ontologie de domaine.

Le cœur du système OntoILPER est le processus d'induction des règles d'extraction par PLI exploitant l'ontologie de domaine à partir d'une BK. Dans cette section nous présentons la spécificité de ce processus d'induction. Tout d'abord nous présentons les différentes hypothèses de travail qui ont été faites dans le développement de OntoILPER. Ensuite nous présentons comment l'ontologie de domaine est utilisée dans ce système. Puis nous présentons quelles caractéristiques ou attributs retenus dans OntoILPER pour générer l'espace des hypothèses pour le

processus d'apprentissage. Enfin nous présentons en détail le modèle de phrase sous la forme de graphe retenu dans OntoILPER pour constituer cet espace d'hypothèses.

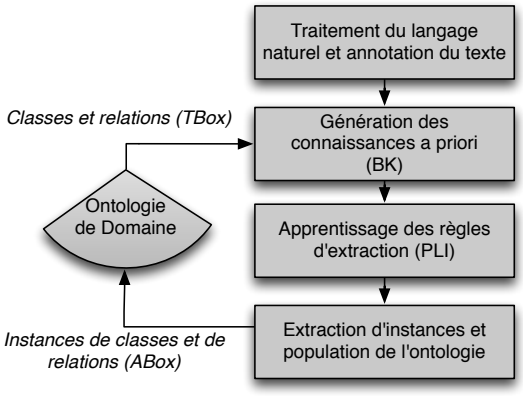

Figure 1. Processus général d'extraction dans OntoILPER

3.1. Hypothèses de travail

Étant donné une phrase S formée par une séquence ordonnée de mots w et les entités e_i $\{e_1, e_2, ..., e_n\}$ de S, et une relation binaire entre une paire d'entités contenues dans S, $R_{ij} = (e_i, e_j)$, où e_i et e_j sont le premier et le deuxième argument de la relation R_{ij}, la tâche d'extraction de la relation consiste à correctement assigner une étiquette $t_i \in T_R$ à l'ensemble de toutes les relations distinctes $\{ R_{ij} \}$ mentionnées dans S. Les principales hypothèses de travail qui ont été faites dans le développement de ce système sont les suivantes :

– *Première hypothèse* : on limite dans OntoILPER l'ensemble des étiquettes ou types d'entités et de relations prédéfinies à extraire et les étiquettes à respectivement T_E et T_R. De plus, on ajoute à l'ensemble T_R, le type spécial *no_rel* pour spécifier l'absence d'une relation type. Il est à noter que les instances de la relation R_{ij} sont généralement orientées, c'est à dire que $R_{ij} \neq R_{ji}$, du fait que les entités e_i et e_j jouent généralement des rôles différents dans la même phrase S.

– *Deuxième hypothèse* : on supposera qu'une ontologie de domaine existe déjà. Cette ontologie spécifie les concepts et les relations pertinentes du domaine d'application.

– *Troisième hypothèse* : on ne considère que les relations entre les entités au sein d'une même phrase, ce qui est le cas de nombreux corpus de référence pour l'évaluation de systèmes d'extraction d'information dans plusieurs campagnes ACE (Automatic Content Extraction), dans lesquels les relations impliquant des entités de phrase différentes ne sont pas considérées (on ne traite pas la coréférence).

– *Quatrième hypothèse* : on ne considère pas les relations réflexives, c'est à dire de type R_{ii}.

Notons que les entités d'une phrase peuvent être données dans le corpus d'entrée, ou elles peuvent être reconnues dans la phase lors de l'étape de prétraitement. Sinon, une classification précoce des entités ou des instances de classe, doit être effectuée. On considérera ici qu'une entité est constituée d'un seul mot ou de deux mots ou plus consécutifs, avec une limite prédéfinie. Dans ce dernier cas, on peut supposer que des *groupes (chunks) nominaux*, avec leur mot de tête correspondant, caractérisent une entité multimot (*multi-word entity*).

3.2. Rôle de l'ontologie de domaine dans OntoILPER

Dans OntoILPER, l'ontologie de domaine guide le processus de génération de la BK en définissant tout d'abord le niveau d'abstraction (classes et super-classes) des prédicats de la BK à partir desquels les règles d'extraction seront induites. Ainsi les axiomes de la TBox de cette ontologie (étiquettes de classes et de propriétés, des propriétés des données ou des objets, relations *is-a*, et le *domain/range* de relations non taxonomiques) sont pris en compte lors de l'étape de génération de la BK dans OntoILPER. Cette utilisation de l'ontologie de domaine dans OntoILPER est conforme aux trois premiers niveaux de connaissance ontologique utilisés par la plupart des systèmes OBIE (Karkeletis *et al.,* 2011) :

– Au *premier niveau*, les ressources ontologiques exploitées sont les entités de domaine (par exemple, personne, lieu), ainsi que leurs synonymes ou co-référents, et les classes de mots (mots-clés, termes, descripteurs d'entités). Ces ressources sont principalement utilisées dans OntoILPER pour l'extraction d'entités nommées;

– Au *deuxième niveau*, les principales ressources sémantiques exploitées sont les entités de domaine organisées en hiérarchies conceptuelles. Ces ressources sont exploitées dans OntoILPER pour induire les règles de généralisation/spécialisation pour l'extraction d'entités nommées;

– Au *troisième niveau*, sont exploitées les propriétés de concepts et/ou les relations entre les concepts de l'ontologie. Ces ressources sont principalement utilisées pour l'induire les règles pour l'extraction de relations.

3.3. L'espace d'hypothèses dans OntoILPER

Différents travaux en EI ont montré que le choix des caractéristiques obtenues par les outils de TAL peut fortement influencer les résultats de l'extraction. Les caractéristiques utiles pour l'extraction de relations incluent le mot, le type d'entité, les *chunks*, les arbres d'analyse syntaxique, et les relations de dépendance (Zhou *et al,* 2005 ; Jiang, 2007 ; Zhang et Zhou, 2008).

En s'appuyant sur ces travaux ainsi que sur les limites des méthodes basées sur les noyaux (*Kernel-based methods*) mentionnées précédemment, nous avons défini un ensemble d'attributs (ou caractéristiques) conduisant à définir un espace d'hypothèses adapté au processus d'extraction. Cet espace intègre des attributs tant morphosyntaxiques que sémantiques du texte, et repose sur une représentation spécifique des phrases du texte sous forme de graphe (cf. section 3.4.). Par rapport

aux méthodes basées sur les noyaux (cf. section 2), OntoILPER apporte les améliorations importantes suivantes :

– La sélection de caractéristiques est basée dans OntoILPER sur une étude des caractéristiques les plus utiles et efficaces pour l'extraction d'information. Chaque caractéristique doit avoir une signification claire, c'est-à-dire, facilement comprise par un expert humain du domaine. Ainsi des combinaisons complexes de caractéristiques et de ratios statistiques n'ont pas été retenues dans notre processus de sélection, principalement en raison des mauvais résultats obtenus dans des travaux existants.

– Comme indiqué précédemment, les méthodes à arbres de noyau (kernel-based methods) ne sont pas en mesure d'exploiter pleinement l'information structurelle, de même avec les méthodes à arbres de noyau basées sur le plus court chemin des dépendances (Shortest-Path dependency tree kernel methods). Dans OntoILPER l'espace des hypothèses est structuré en trois niveaux d'information structurelle : *niveau séquentiel, niveau des chuncks syntaxiques, et niveau des arbres de dépendances syntaxiques.*

– Pour réduire le temps d'apprentissage et la redondance de caractéristiques, l'espace d'hypothèses d'OntoILPER se restreint à un ensemble compact de caractéristiques informatives et pertinentes, plutôt que des centaines, voire des milliers, dans la plupart des méthodes basées sur les noyaux.

3.4. Une représentation des phrases sous forme de graphe

L'espace d'hypothèses d'OntoILPER repose sur une représentation « relationnelle » spécifique des phrases du texte, un modèle de phrase sous forme de graphe orienté dans lequel les relations sont des arcs et les entités des nœuds. Ce modèle de phrase est basé à la fois sur des caractéristiques structurelles et sur des propriétés décrivant des entités et des relations. Ces caractéristiques sont ensuite formalisées par des prédicats logiques, qui exploités par PLI, permettent l'induction de règles d'extraction symboliques à partir d'exemples. Ce modèle s'appuie principalement sur deux types de techniques de TAL : le *chunking* (technique peu profonde), et l'analyse des *dépendances* (technique profonde).

L'analyse de Chunking, utile pour définir les frontières de l'entité, et la tête des constituants des *phrases* nominales, verbales et prépositionnelles. Pour la même phrase, la figure 2a montre les *tokens de tête* et les *chunks* issus d'une analyse de *Chunking*. Généralement, les *phrases* verbales sont des candidats possibles pour les relations, et des valeurs nominales des candidats pour une entité ou une instance de classe.

L'analyse de dépendances génère des dépendances typées de la phrase et produisant un graphe de dépendances (Marneffe et Manning, 2008). Ce graphe orienté est le résultat d'un algorithme d'analyse tout-chemin (*all-path parsing algorithm*) basé sur une grammaire de dépendance (Kruijff, 2002) dans laquelle la structure syntaxique est exprimée en termes de relations de dépendance entre les paires de mots, une tête (*head*) et un modificateur (*modifier*). Toutes les

dépendances dérivées d'une phrase définissent un graphe de dépendances dont la racine est un mot qui ne dépend d'aucun autre mot. Nous avons adopté les dépendances typées proposées dans (Marneffe et Manning, 2008), dites *dépendances de Stanford*. Différentes représentations des dépendances de Stanford sont disponibles dans l'analyseur « Stanford parser »[4] utilisé ici. Nous avons retenu la représentation *collapsed tree* dans laquelle les dépendances impliquant des prépositions, conjonctions, ainsi que des informations sur le référent de clauses relatives, sont collapsées pour obtenir des dépendances directes entre les mots. Cette représentation a l'avantage de réduire le nombre de dépendances typées dans un graphe de dépendance donné en réduisant certaines dépendances très fréquentes (Marneffe et Manning, 2008), ce qui simplifie le processus d'extraction de relations. La figure 2b présente un graphe de dépendance obtenu par une analyse de dépendances pour la phrase *Myron Kandel at the CNNfn Newsdesk in New York.*

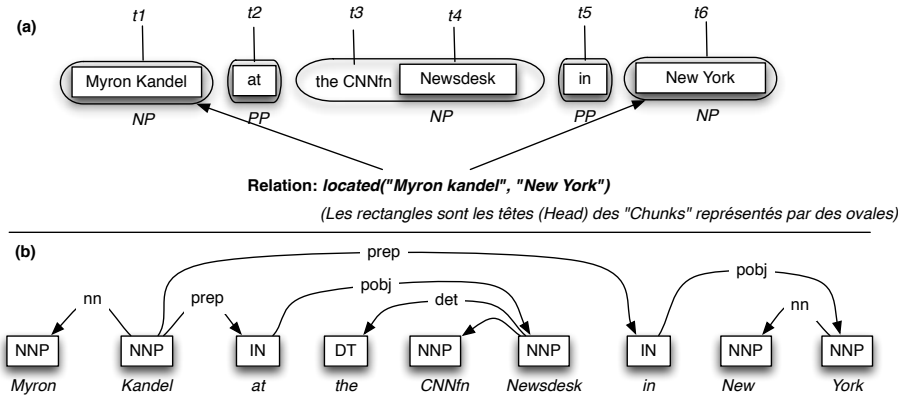

Figure 2. a) Analyse de Chunking de cette même phrase. b) Graphe de dépendances de la phrase Mary Kandel at the CNNfn Newsdesk in New York

La figure 3 présente le modèle de phrase mettant en évidence au travers des arcs du graphe : (i) une *analyse de dépendance* (dépendances collapsées - arcs *prep_on*), (ii) une analyse de *Chunking* (tokens de tête en gras), (iii) un séquençage de tokens dans une phrase (arcs *NextToken*). Ce modèle fait aussi apparaître les caractéristiques morpho-syntaxiques comme des nœuds attributs (flèches de couleur grise), et les attributs sémantiques, comme les entités nommées (en gras).

Dans ce modèle de phrase les arcs du graphe sont considérés comme des « caractéristiques relationnelles » pouvant être exploitées dans l'induction en PLI des règles d'extraction. Enfin les *tokens* ou des constituants de *chunking* d'une phrase référencent potentiellement des concepts ou des relations définies dans l'ontologie de domaine.

4. Stanford NLP Group. The Stanford Parser: A statistical Parser. http://nlp.stanford.edu/software/lex-parser.shtml

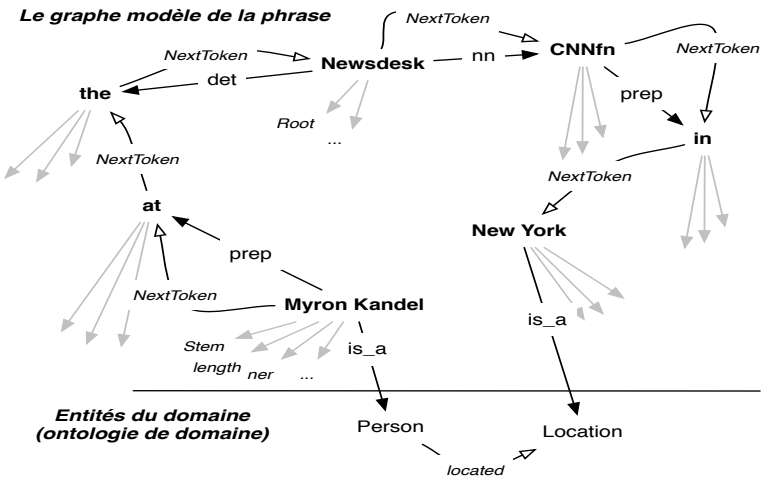

Figure 3. Modèle de la phrase Myron Kandel at the Newsdesk CNNfn in New York

Le modèle de phrase de la figure 3 peut être ensuite exprimé par un ensemble de relations binaires assimilables à des prédicats. Voici une liste de quelques relations binaires possibles : *det(Newsdesk, the), nn(Newsdesk, CNNfn), prep(Myron-Kandel, at), prep(CNNfn, in), NextToken(The, Newsdesk).* Les arguments de ces relations binaires peuvent être enrichis avec des contraintes supplémentaires sur les types d'arguments, de même des relations binaires supplémentaires peuvent être ajoutées et utilisées par le composant PLI d'apprentissage par induction pour lier les termes de la phrase avec des classes et relations de l'ontologie de domaine. Par exemple, si le prédicat à apprendre est *read(X, Y),* ou en termes ontologiques la propriété d'objet *read(X, Y),* alors le premier argument X devrait être une instance de la classe de « personne », alors que le second Y devrait être une instance de la classe « publication » dans l'ontologie de domaine. Ainsi des instances de classes et de relations peuvent être considérées respectivement, comme des nœuds et des arrêtes dans notre modèle, et chaque nœud pouvant avoir plusieurs attributs, par exemple, l'étiquette de la classe de l'ontologie à laquelle il appartient.

Dans le processus d'extraction, l'identification les étiquettes des instances candidates de classes et de relations sont définies comme des prédicats cibles dans le problème d'apprentissage. Plus concrètement, il s'agit d'apprendre ces prédicats cibles comme une combinaison de plusieurs éléments de la phrase donnés par le modèle de phrase. La plupart des travaux existant dans l'extraction d'entités et de relations ont seulement considérés des caractéristiques attribut-valeur, ou des caractéristiques propositionnel dérivés des données du texte d'entrée (Finn 2006 ; Giuliano 2007 ; Roth et Yih, 2007 ; Kambhala, 2004 ; Zhou *et al.,* 2005). On s'appuie ici sur une représentation des exemples en logique du premier ordre, une représentation beaucoup plus riche pour mener la tâche de classification (Fürnkranz *et al.,* 2012).

3.5. Caractéristiques relationnelles linguistiques et structurelles retenues

Les caractéristiques relationnelles linguistiques et structurelles fournies principalement par les analyses de dépendances et de chunking constituent l'espace d'hypothèses décrivant chaque unité sémantique du corpus, et constituent les principaux éléments de la BK d'OntoILPER. On distingue quatre principales catégories de caractéristiques :

– Les *caractéristiques lexicales* concernent le mot lui-même, son lemme, la longueur et le type morphologique général.

– Les *caractéristiques syntaxiques* constituées d'étiquettes POS de mots (*Part of Speach*) : mot de tête de chunks nominaux, de chunks prépositionnels ou verbaux; et sa forme généralisée qui ne considère ni pluriel pour les noms, ni les temps pour les verbes. Il est également prévu deux bigrammes et trigrammes d'étiquettes POS de mots consécutifs tels qu'ils apparaissent dans la phrase. Les caractéristiques de *Chunking* sont aussi dans cette catégorie. Les *chunks* segmentent les phrases en groupes nominaux, prépositionnels, et verbaux. L'analyse de ces *chunks* fournit des informations très utiles : leur *type* (nominal, verbal ou prépositionnel), leur *mot de tête*, leur *position relative* au verbe principal de la phrase. Ces caractéristiques décrivent la distance des arguments du verbe, dans le rôle de sujet ou d'objet, et sont très utiles dans le processus d'extraction.

– Les *caractéristiques sémantiques* comprennent les entités nommées reconnus par la REN, et les entités additionnelles mentionnées dans le corpus d'entrée. Par exemple, dans le corpus TREC, chaque entité annotée dans ce corpus est mentionnée (une personne (PER), une organisation (ORG) ou une localisation (LOC)).

– Les *caractéristiques structurelles* structurent toutes les autres caractéristiques du modèle de phrase proposé (section 3.4). Tout d'abord, le séquençage de tokens spécifiant l'ordre des tokens tels qu'ils apparaissent dans la phrase. Ensuite, la relation partie-tout (*part-whole*) entre les tokens et le chunk qui les contient est spécifiée. De même, le séquençage des chunks est spécifié par leurs arcs entre les tokens de tête. La dernière caractéristique structurelle concerne la dépendance grammaticale entre deux tokens dans une phrase, selon le type de dépendances entre les mots donné par l'analyseur de dépendances de Stanford. Cela inclut le nom de chaque dépendance typée entre les têtes de deux tokens dans une phrase.

4. Le système OntoILPER

Dans cette section, nous présentons quelques détails d'implémentation du système OntoILPER mettant en œuvre la méthode d'extraction présentée dans la section précédente. Nous présentons d'abord l'architecture générale d'OntoILPER et ensuite chacun de ses composants en mettant plus particulièrement l'accent sur les composants de génération de la BK et d'induction des règles d'extraction en PLI.

4.1. Architecture générale du système OntoILPER

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 figure 4.

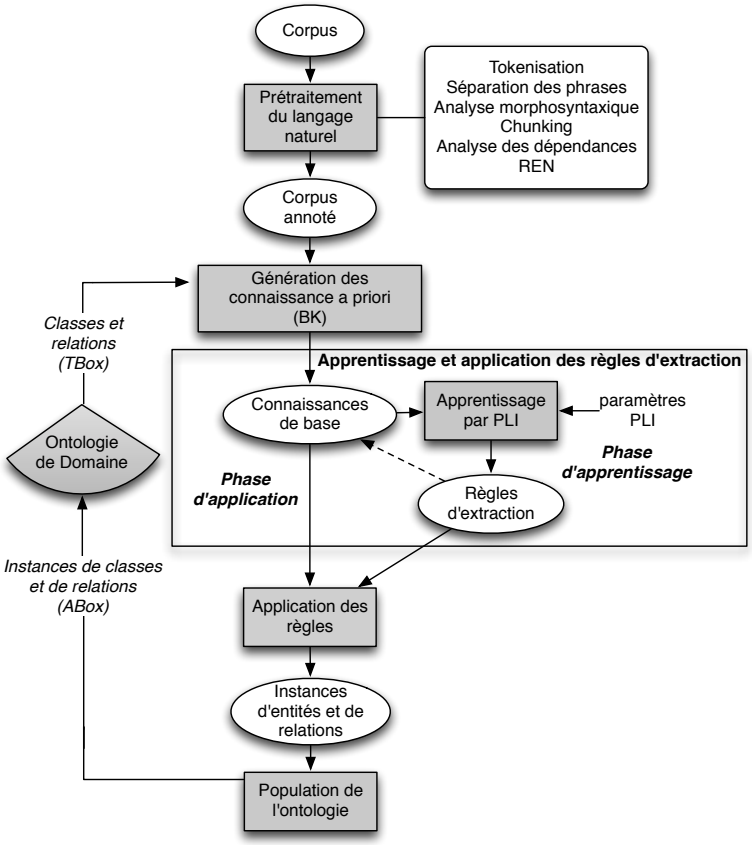

Figure 4. Architecture générale du système OntoILPER

La première phase est la *phase d'apprentissage* dans laquelle, à partir d'un corpus d'exemples annotés, d'une ontologie de domaine, est induite par apprentissage en PLI une théorie, correspondant à un ensemble de règles d'extraction (cf. figure 4). La seconde phase est la *phase d'application*, dans laquelle l'ensemble final de règles d'extraction induites est appliqué à des documents, 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 prétraitement des documents textuels mettant en œuvre différentes techniques de TAL, ainsi qu'un traitement permettant la génération de la BK à partir de ces exemples annotés.

4.2. Composant de prétraitement du langage naturel

Ce composant réalise une annotation automatique du corpus de document textuels au moyen d'outils de TAL. Cette annotation doit être réalisée avec soin car elle spécifie tous les aspects morphosyntaxiques et sémantiques présents dans le texte qui serviront à l'élaboration du modèle de phrase permettant de générer la BK. Dans OntoILPER nous nous intéressons essentiellement à des documents en langue anglaise, et ce composant intègre les outils Stanford CoreNLP[5] et OpenNLP[6]. Ces outils sont intégrés de façon à réaliser séquentiellement les sous-tâches suivantes : fractionnement des phrases, tokénisation, étiquetage morpho-syntaxique, lemmatisation, Chunking, et analyse de dépendances.

Cette annotation utilise généralement différents formats de d'entrée et de sortie. Pour cela nous avons défini un modèle d'annotation linguistique efficace, accessible et lisible par des humains pour vérification ou correction. Ce modèle (XML hybride) est basé sur la combinaison de deux normes différentes d'annotation linguistique : PAULA (Erk et Pado, 2004) et TIGGER/SALSA (Dipper *et al.,* 2007). Du premier nous avons adopté la représentation en ligne permettant la lisibilité et l'efficacité sur les requêtes au modèle, du second nous avons retenu un codage flexible des arbres et graphe représentant les annotations syntaxiques.

4.3. Composant de génération des connaissances de base (BK)

Dans l'apprentissage *relationnel*, et dans OntoILPER, chaque exemple est représenté indépendamment des autres. Ainsi, le problème de la dispersion des données pour représenter les exemples est fortement réduit (Fürnkranz *et al.,* 2012). Les limitations liées à la représentation vectorielle de l'apprentissage propositionnel ci-dessus sont ainsi surmontées par l'usage d'un formalisme relevant de la logique du premier ordre, clauses Prolog, pour la représentation tant de la BK que des exemples. Cette BK repose sur un modèle sous forme de graphe des phrases et des exemples. La figure 5 présente le métamodèle en formalisme entité-relation (E-R) de ce modèle de phrase. Dans ce métamodèle les attributs des entités (E) du métamodèle désignent les prédicats définissant des propriétés, alors que les relations (R) entre les entités (E) correspondent à des prédicats structurels.

Ce composant de OntoILPER génère la BK en convertissant les entités de domaine, relations, et toutes les caractéristiques linguistiques et structurelles des phrases mentionnées précédemment en prédicats Prolog spécifiques. La base de connaissances factuelles ainsi obtenue peut être utilisée à la fois pour restreindre l'espace d'hypothèses et guider la recherche. Le tableau 1 illustre une telle conversion sur la phrase exemple précédente, en présentant les prédicats logiques caractérisant l'instance candidate de la classe « Personne » : l'instance « Myron ».

---

5. Stanford CoreNLP Tools. http://nlp.stanford.edu/software/corenlp.shtml.
6. Apache OpenNLP. The Apache Software Foundation. http://opennlp.apache.org

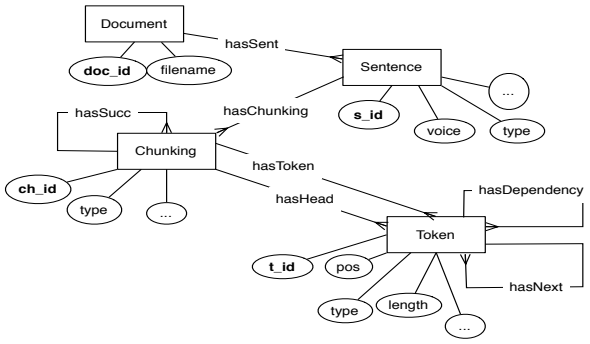

Figure 5. Métamodèle Entité-Relation du modèle de phrase d'OntoILPER

Tableau 1. Prédicats Prolog décrivant le token « Myron » (t_1)

Groupe	Prédicats Prolog	Signification
Corpus entities	*doc(d_1)*	*d_1* is a document identifier
	sent(s_1)	*s_1* is a sentence identifier
	chunk(ck_1)	*ck_1* is a chunk identifier
	token (t_1)	*t_1* is the token identifier
Lexical features	*t_stem (t_1, "Myron")*	token *t_1* stemming is "Myron"
	t_length (t_1, 5)	token *t_1* has length of 5 characters
	t_orth (t_1, upperInit)	token *t_1* begins with an initial uppercase letter
	t_morph_type(t_1, word)	token *t_1* is a word
Syntactical features		
POS and POS n-grams	*t_pos (t_1, nnp)*	token *t_1* is a singular proper noun
	t_gpos(t_1,nn)	token *t_1* is a canonical noun (no plurals)
	t_bigPosBef (t_1, ...)	POS tag bigram before token *t_1*
	t_bigPosAft (t_1, vbz-vbg)	POS tag bigram after token *t_1*
	t_trigPosBef (t_1, ...)	POS tag trigram before token *t_1*
	t_trigPosAft (t_1 vbz-vbg-dt)	POS tag trigram after token *t_1*
Chunking analysis	*ck_hasHead(ck_1, t_1)*	*ck_1* has *t_1* as its token head
	ck_hasType(ck_1, np)	*ck_1* is a nominal chunk
	t_isHeadNP(t_1)	*t_1* is the head token of a nominal chunk
	ck_dist_to_root(ck_n, near)	*ck_n* is near the main verb of the sentence
	t_ck_tag_type(t_1, np)	token *t_1* has the chunking type *np*
Semantic features	*t_ner(t_1, person)*	*t_1* was annotated by the NER as a Person entity
Predefined corpus annotation types	*t_type(t_1, person)*	*t_1* has the PERSON type
	t_subtype(t_1, none)	*t_1* has no subtype
	t_mtype(t_1, name)	*t_1* is a named proper noun
Structural features	*t_next (t_1, t_2)*	token *t_1* is followed by the token *t_2*
	t_next_head (t_1, t_3)	head token *t_1* is followed by head token *t_3*
	ck_hasToken(ck_1, t_1)	*t_1* is one the tokens in *ck_1*
	ck_hasSucc(ck_1, ck_2)	*ck_1* is followed by the chunk *ck_2*
	t_hasDep (nn, t_2, t_1)	*t_1* has a multi-word dependency with *t_2*
	t_root (t_n)	*t_n* is the root (main verb) of the dependency tree

La représentation en logique du premier ordre et en Prolog des caractéristiques est assez simple. Un prédicat unaire correspond à une entité, tandis que les prédicats binaires correspondent à des attributs (attribut, valeur) ou des relations binaires (rel (arg1, arg2)). Contrairement aux autres approches d'apprentissage automatique qui utilisent des vecteurs de caractéristiques pour représenter les fenêtres contextuelles (N tokens à droite/gauche d'un mot donné w dans une phrase), nous utilisons le prédicat t_next(t_1, t_2) qui concerne un token à son successeur immédiat dans un phrase (cf. tableau 1).

Notons que dans OntoILPER, l'utilisateur peut spécifier des connaissances déclaratives supplémentaires afin d'aider le processus d'induction, en ajoutant des prédicats intentionnels spécifiques, pouvant par exemple, discrétiser des caractéristiques numériques. Ainsi dans la figure 6 le premier prédicat (Tok_lengh) catégorise la longueur du token le plus court, de taille moyenne ou longue, et le second prédicat (Ck_dist_to_root) discrétise la distance en nombre de tokens au verbe principal (root) de la phrase. Ce type de prédicats définis par l'utilisateur permet une meilleure généralisation d´une règle.

```
% Token length type definition
length_type(short). length_type(medium). length_type(long).
tok_length(T, short) :- token(T), t_length(T, X), X =< 5.
tok_length(T, medium):- token(T), t_length(T, X), X > 5, X =< 15.
tok_length(T, long) :- token(T), t_length(T, X), X > 15.

% Chunking distance to the main verb
ck_dist_root(CK, near):- ck_posRelPred(CK, X), X >= -3, X =< 3.
ck_dist_root(CK, far) :- ck_posRelPred(CK, X), ( ( X >= -8, X < -3) ;
        (X > 3, X =< 8) ).
ck_dist_root(CK, very_far):- ck_posRelPred(CK, X),(( X < -8); (X > 8)).
```

Figure 6. Prédicats intentionnels spécifiques ajoutés à la BK originale

4.4. Composant d'apprentissage des règles d'extraction

Ce composant d'apprentissage d'OntoILPER est basé sur la PLI mis en œuvre au travers du système général de GIPS avec le modèle Progolem. Ce composant repose sur le paramétrage prédictif de la PLI pour induire des règles d'extraction (théories) conduisant à construire des classificateurs symboliques capables de faire la distinction entre les exemples positifs et négatifs. Par conséquent, il est nécessaire d'adapter cette technique d'apprentissage supervisée au problème de l'extraction d'entités et de relations, notamment d'imposer les restrictions suivantes aux règles d'extraction induites : (i) elles doivent tenir compte de la BK en termes de structure et de propriétés des caractéristiques définies par notre modèle de phrase, (ii) elles doivent être bien formées par rapport à la *linkedness* de la variable dans la clause, c'est à dire, il y ait une chaîne de littéraux reliant les variables d'entrée de la tête aux variables de sortie de la tête (Santos, 2010), enfin (iii) leurs aspects qualitatifs,

exprimées par des motifs linguistiques pertinents doivent être facilement compréhensibles par l'expert de domaine.

4.4.1. Scénarios d'apprentissage

Une caractéristique intéressante de la composante d'apprentissage d'OntoILPER est sa capacité à utiliser des règles déjà apprises dans une étape d'apprentissage (itération i), comme des prédicats additionnels de la BK dans l'étape d'apprentissage suivante (itération *i + 1*), cette capacité est appelée *pipeline method* (Roth et Yih, 2007). La figure 7 représente le flux d'informations échangées entre le composant de génération de la BK et le composant d'apprentissage des règles, qui permettent de distinguer deux grands scénarii d'apprentissage.

Dans le *premier scénario d'apprentissage*, la reconnaissance d'une instance d'entité ou de relation est faite dans le *composant apprentissage de règles*, à partir d'une BK au moyen de règles d'extraction déjà apprises. Cela correspond à la plupart des scénarios d'apprentissage des campagnes d'ER, comme ACE RDC[7] dans laquelle toutes les connaissances de base appropriées sont disponibles dans les exemples d'apprentissage, comme les annotations sur les instances d'entités. En d'autres termes, une paire d'entités déjà identifiées, qui représente les deux arguments d'une instance de relation cible, est donnée au classificateur de relations. Ce scenario d'apprentissage « facilité » peut ne pas correspondre à un réel besoin d'extraction.

Dans le *second scénario d'apprentissage*, plus réaliste, le classificateur de relations ne connaît pas les étiquettes de ses arguments d'entités. Ainsi le *composant apprentissage de règles* doit d'abord identifier les classes ou étiquettes des entités argument, ce qui signifie générer les règles d'extraction pour classer les deux entités impliquées comme argument d'une relation. Ensuite, les étiquettes des nouvelles classes des entités arguments sont traitées comme BK complémentaire dans le *composant de génération de la BK*, puis la nouvelle BK est transmise au *composant apprentissage de règles* qui tente de découvrir une relation entre ces classes d'entités.

Ainsi les deux composants *Génération de la BK* et *Apprentissage des règles* peuvent être exécutés en boucle un certain nombre de fois, les nouvelles règles découvertes dans une itération i, peuvent être utilisées ou faire partie des nouvelles règles d'extraction dans une itération i + 1.

4.4.2. Modèles d'extraction de classes et de relations

Dans OntoILPER, comme le proposent (Roth et Yih, 2007), il y a trois grands types de modèles d'extraction d'instances de classes et de relations possibles : les modèles « Séparé ou Disjoint », « Pipeline » et « Omniscient » illustrés à la figure 7.

7. ACE (2004). Automatic Content Extraction 2004 Evaluation. http://www.itl.nist.gov/iad/mig/tests/ace/2004

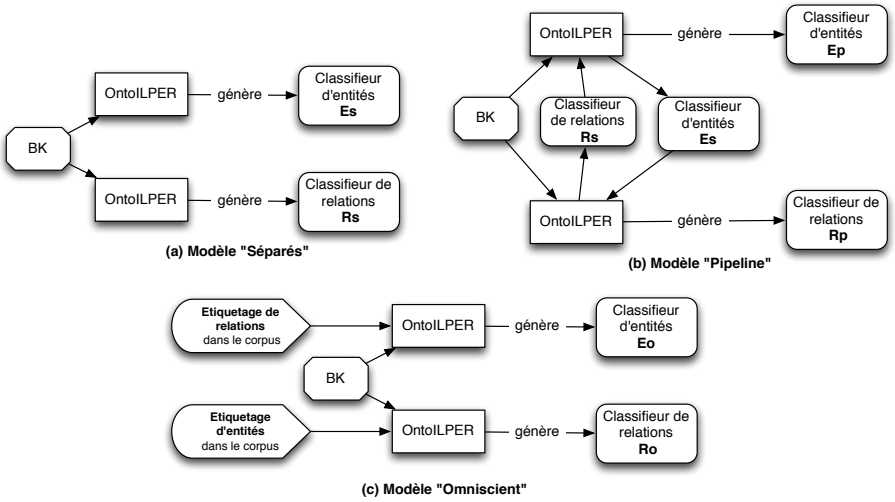

Figure 7. Types de modèles d'extraction d'entités et de relations

– *Modèle « Séparé ou Disjoint »*. Dans ce modèle, les classifieurs d'entités E_S et de relations R_S, sont construits de façon indépendante dans OntoILPER à partir de la BK. Ainsi le classifieur d'entités E_S est construit sans connaître les étiquettes des relations dans la phrase, de même que le classifieur de relations R_S est construit sans avoir la connaissance des étiquettes de leurs entités arguments. (cf. figure 7a).

– *Modèle « Pipeline »*. Dans ce modèle, le classifieur final d'entités, noté E_P, est obtenu après un premier apprentissage à partir de la BK dans OntoILPER d'un classifieur de relations noté R_S, qui est ensuite utilisé comme caractéristiques supplémentaires dans l'apprentissage du classifieur d'entités E_P. De même, le classifieur de relations final, noté R_P, est obtenu en utilisant les prédictions sur les deux entités arguments des relations, données par le classifieur d'entités E_S, comme caractéristiques supplémentaires dans son processus d'apprentissage (cf. figure 7b).

– *Modèle « Omniscient »*. Dans ce modèle, l'apprentissage dans OntoILPER à partir de la BK, des classifieurs d'entités Eo utilisent comme caractéristiques supplémentaires des étiquettes des relations sur corpus annoté, et de même, l'apprentissage dans OntoILPER des classifieurs de relations Ro utilise comme caractéristiques supplémentaires des étiquettes d'entités sur corpus annoté (cf. figure 7c).

4.4.3. Génération des modèles de règles

La génération des modèles des règles dans OntoILPER est faite avec le système GILPS selon le modèle ProGolem. Pendant l'apprentissage, la recherche dans l'espace d'hypothèse de règles utiles que ProGolem doit réaliser est la tâche la plus coûteuse du processus inductif. Une recherche exhaustive de toutes les hypothèses possibles avec un test de couverture sur l'espace d'hypothèses complet n'est pas

réalisable. De plus, pour trouver la meilleure hypothèse à partir d'un exemple donné, il est nécessaire de tester chaque hypothèse trouvée par rapport aux exemples positifs et négatifs. Aussi il est nécessaire de parcourir intelligemment cet espace d'hypothèses, en tirant parti de sa structure particulière, et de ne pas poursuivre l'exploration de parties dans cet espace qui ne contiennent pas de solutions intéressantes. Heureusement, cet espace d'hypothèses peut être structuré par une relation d'ordre partiel entre deux hypothèses permettant une navigation efficace dans cet espace (Muggleton *et al.,* 2009). Des biais supplémentaires sont également utilisés non seulement pour réduire l'espace des hypothèses, mais aussi pour assurer un processus d'induction efficace. L'apprenant de règles *ProGolem* fait principalement usage de *déclarations de mode* pour délimiter espace de recherche d'hypothèses, et de *paramètres spécifiques PLI* pour modifier son processus de construction de la théorie par défaut.

4.4.4. Exemple de règles induites

Considérons une théorie complète induite pour la relation *part_whole* obtenue avec les paramètres suivants : *theory construction* = global*, i (depth)* = 3, *minimum precision* = 0.0, *minimum positive examples* = 5, et *noise* = 20 %, en laissant les autres paramètres avec leurs valeurs par défaut. Cette théorie est composée de seulement deux règles suivantes :

Règle 1:
#Literals = 4, Positive Score = 90; Negative Score = 1; Precision = 98.9%
part_whole(A,B):- t_gpos(A,nn), t_next(A,B), t_subtype(B,state-or-province).

Règle 2:
#Literals = 5, Positive Score = 31; Negative Score = 7; Precision = 77.4%
part_whole(A,B):- t_next(A,B), t_pos(A,nnp), t_ne_type(B,gpl), t_subtype(A,pop-center).

Les règles de cette théorie qui classifient des instances de la relation *part_whole*, sont caractérisées par le nombre de littéraux impliqués, le nombre d'exemples positifs et négatifs couverts (positive and negative scores), et leurs précisions (P). Les règles sont aussi évaluées en utilisant leur taux de compression (*compression ratio*) : *(exemples positifs - exemples négatifs) / longueur de la règle*.

La Règle 1 a une bonne précision (P = 98,9) du au nombre élevé de phrases contenant deux tokens adjacents A et B (*t_next(A,B)*), où le premier token A est un nom (*t_gpos(A,nn)*), et le second B est étiqueté grâce à l'ontologie de domaine, comme un exemple de sous-type de la classe « state-or-province » (*t_subtype(B,state-or-province)*). Cette règle met ainsi en avant que les *lieux* (A) comme « *villes* » sont *localisées*, ou font partie d'un *état* ou d'une *province*.

La Règle 2, de précision moindre (P = 77,4), similaire à la règle précédente, implique aussi deux tokens adjacents A et B. Le token A est un nom propre *(t_pos(A,nnp))* et est associé à la sous-classe *Population-Center* dans l'ontologie *(t_subtype(A,pop-center))*. Le token B est une instance d'entité nommée de lieu géographique politique (*t_ne_type(B,gpl))*.

4.5. Composants d'application des règles d'extraction et de peuplement de l'ontologie de domaine

Le composant Application des règles d'extraction applique l'ensemble final des règles induites sur la base de connaissances Prolog générée à partir de nouveaux documents similaires à ceux utilisés dans la phase d'apprentissage. En conséquence, de nouvelles instances de classes et de relations sont extraites, et enfin peuvent être intégrées dans l'ontologie de domaine. Une autre possibilité consiste simplement à enregistrer les instances extraites dans d'autres formats, par exemple, des tables relationnelles et des fichiers XML. Voici quelques instances extraites de la phrase mentionnée dans la section 3, *Myron Kandel at the CNNfn Newsdesk in New York :*

Person("Myron Kandel") // "Myron Kandel" est une instance de la classe "Person"
Location("New York") // "New York" est une instance de la classe "Location".
Is Located("Myron Kandel", "New York"). // "Myron Kandel" est localisée à New

Pour finir, le composant *Peuplement de l'ontologie de domaine*, les prédicats Prolog sont convertis en axiomes OWL et sont utilisés pour peupler l'ontologie de domaine. Le raisonner comme Pellet peut être utilisé pour effectuer diverses vérifications, notamment afin d'éviter une redondance d'instances dans l'ontologie.

5. Évaluation expérimentale d'OntoILPER

Dans cette section, nous présentons et discutons de résultats expérimentaux obtenus par OntoILPER dans l'extraction d'instances d'entités et de relations sur un corpus de référence, le corpus TREC. Dans un premier temps, nous présentons ce corpus, la génération des exemples négatifs, le paramétrage utilisé pour l'expérimentation, et l'évaluation métrique et la validation croisée utilisées. Ensuite nous présentons les résultats obtenus par OntoILPER sur le corpus TREC.

5.1. Le corpus TREC

Le corpus TREC (*Text Retrieval Conference*)[8] proposé au départ par (Roth et Yih, 2004) est composé d'articles du WSJ (Wall Street Journal) et constitue un corpus de référence pour l'extraction d'instances d'entités nommées et de relations entre ces dernières. Il est annoté pour les entités nommées et les relations, et contient 1441 phrases avec 5349 entités, à savoir, 1691 personnes, 1968 emplacements, 984 organisations et 706 noms divers. Chacune des 1441 phrases a au moins une relation à extraire, sachant que les relations à extraire sont des instances des 5 relations précisées dans le tableau 2 : *located_in*, *work_for*, *based_in*, *live_in* et *kill*. Parmi ces phrases, il y a 19080 relations binaires possibles avec la distribution des exemples positifs précisée dans le tableau 2.

8. http://cogcomp.cs.illinois.edu/Data/ER/conll04.corp

Tableau 2. Relations binaires à extraire avec leurs arguments

Relation	Arg-1	Arg-2	Example	# of relations
located_in	LOC	LOC	(Toledo, Ohio)	405
work_for	PER	ORG	(Winter, Court)	401
based_in	ORG	LOC	(HP, Palo Alto)	452
live_in	PER	LOC	(Tvazir, Israel)	521
kill	PER	PER	(Oswald, JFK)	268

Ce tableau présente aussi un exemple de chaque relation et les contraintes à l'égard de ses deux arguments. La plupart des relations binaires candidates n'ont pas de relation active du tout, du fait d'une répartition non équilibrée entre exemples positifs et exemples négatifs.

La figure 9 représente l'ontologie de domaine créée pour extraire et stocker ensuite les instances extraites par OntoILPER sur ce corpus.

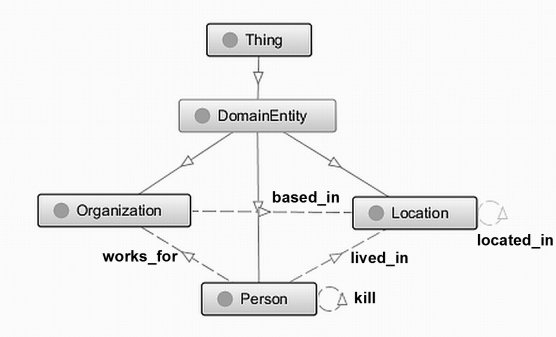

Figure 9. Ontologie sous-jacente au corpus TREC avec ses entités et relations

5.2. Génération des exemples négatifs

L'apprentissage de règles d'extraction dans OntoILPER nécessite des exemples positifs et négatifs Le corpus TREC ne comportant pas d'exemples négatifs, il est nécessaire d'en créer, en tant que compléments d'exemples positives, selon la technique *« one vs. all class binarization »*. En bref, il s'agit de produire plusieurs jeux de données « 2-classe » en discriminant chaque classe contre l'union de toutes les autres classes. Ainsi, compte tenu de l'ensemble des *N* classes possibles d'entités C_i, $i = 1..N$, pour chaque instance c_i positive d'une classe C_i donnée dans l'ensemble d'apprentissage, un exemple négatif est créé pour chacune des $N - 1$ autres classes. Un problème d'apprentissage multi-classe est ainsi réduit à plusieurs problèmes d'apprentissage binaires, un pour chaque classe.

Pour l'extraction de relations, la génération d'exemples négatifs est différente. On utilise pour cela la technique proposée dans (Airola, 2008), qui considère l'extraction de relations comme un problème de classification binaire, où les paires d'arguments devant interagir sont les exemples positifs, et les autres paires d'entités concomitantes dans la même phrase sont les exemples négatifs. En conséquence, pour chaque phrase on créée $C_{n\ ,\ 2} = 2! / 2 * (n - 2)!$ exemples, où *n* est le nombre total d'entités dans une phrase donnée.

5.3. Résultats expérimentaux

L'évaluation expérimentale d'OntoILPER sera faite avec une validation croisée (5-fold), car elle garantit l'utilisation maximale des données disponibles, et permet également de se comparer à des expérimentations menées sur le même corpus avec d'autres systèmes d'extraction (cf. section suivante). L'évaluation de la performance est faite selon les mesures classiques en recherche d'information : la précision *P*, le rappel *R* et la *F1-mesure* (Baeza-Yates, 1999).

5.3.1. Évaluation des modèles d'apprentissage

Plusieurs expérimentations ont été faites pour évaluer les trois différents grands types de modèles d'extraction évoqués : modèles *Séparé* (E_S), *Pipeline* (E_P), et *Omniscient* (E_O) (cf. 4.4.2). Les tableaux 3 et 4 donnent les résultats de classification obtenus par ces trois modèles.

On constate dans le tableau 3 que pour la classification des entités Location (LOC), Organization (ORG) et Person (PER), les trois modèles E_S, E_P et E_O. ont obtenus une précision élevée (de 93,5 à 98,7). Le rappel obtenu (de 74,4 à 92,4) montre que plusieurs instances n'ont pas été considérées lors de la classification, que le nombre de cas avec les prévisions de la classe fausse de l'entité. Le tableau 3a montre aussi l'équilibre entre la précision et le rappel dans les trois modèles de classification pour les entités LOC et PER. Au contraire, l'entité ORG a obtenu la plus grande précision que les deux autres entités, mais avec le plus faible rappel.

L'extraction de relations est une tâche plus délicate que l'extraction d'entités, ce que confirment les résultats du tableau 4 obtenus pour les trois modèles d'extraction R_S, R_P et R_O. Comme pour les modèles d'extraction d'entités les modèles d'extraction de relations ont plus de précision (de 85,7 à 93,1) que de rappel (72,1 à 86,1).

Bien que les résultats des tableaux 3 et 4 suggèrent que OntoILPER a de meilleurs résultats en précision plutôt qu'en rappel, en fait OntoILPER peut utiliser d'autres fonctions évaluation qui lui donneraient plus de rappel que de précision, telles que la fonction d'évaluation de rappel (Santos, 2009).

Les résultats du tableau 3 montrent qu'en extraction d'entités les modèles E_O et E_P, plus riches que le modèle E_S, ont des scores légèrement supérieurs en précision et en rappel. Ces résultats étaient attendus car les modèles E_O et E_P sont plus informés que le modèle E_S. Pour l'entité PER cependant, dans Eo les étiquettes

d'entités ont entraîné une baisse de 1,3 % de la précision par rapport à Es, ce qui pourrait être lié à la présence d'exemples bruités dans le corpus TREC.

Tableau 3. Résultats en classification d'entités tous modèles

NER Model	LOC			ORG			PER		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
E_S	**96.0**	88.4	92.0	97.0	74.4	84.3	**94.8**	87.5	91.0
E_P	95.2	92.0	93.5	97.5	76.5	85.7	93.5	89.0	91.3
E_O	95.9	**92.4**	**94.1**	**98.7**	**79.2**	**87.8**	93.7	**91.2**	**92.4**

Tableau 4. Résultats en classification de relations tous modèles

(a) RE Model	located_in			work_for			based_in		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
R_S	**91.2**	75.9	82.6	**93.1**	72.9	81.7	88.4	77.0	82.2
R_P	91.1	78.0	83.9	87.2	80.8	83.8	**91.5**	**84.0**	**87.5**
R_O	90.5	**78.6**	**84.0**	85.7	**86.1**	**85.8**	88.7	82.5	85.4

(b) RE Model	live_in			kill		
	P	*R*	*F1*	*P*	*R*	*F1*
R_S	**92.5**	67.4	78.0	**97.5**	73.7	83.8
R_P	85.7	72.1	78.2	91.5	77.6	83.9
R_O	87.4	**76.9**	**81.7**	92.3	**78.0**	**84.3**

Les résultats des tableaux 4a et 4b montrent qu'en extraction de relations, sauf pour la relation *based_in*, les étiquettes d'entités fournies au modèle Omniscient R_O, diminue la précision des classificateurs, mais contribuent à améliorer les scores de rappel de tous les classificateurs de relations. Cela peut s'expliquer par le fait que l'information bruitée sur les entités dans le corpus peut être atténuée par d'autres indices sur les classificateurs. Des étiquettes d'entités correctes permettent aux classificateurs de couvrir plus d'exemples.

Enfin les résultats des tableaux 3 et 4, indiquent que les modèles *Pipeline* (R_P, E_P) ont été plus performants que les modèles *Séparés* (R_S, E_S) sur les deux tâches de REN et de ER. Pour les modèles d'extraction de relation, les résultats globaux en F1-mesure ont montré une différence statistiquement significative entre les modèles de R_P et R_S. Ainsi la capacité d'OntoILPER à utiliser des règles apprises dans une phase d'apprentissage précédente comme prédicats additionnels de la BK dans une

étape d'apprentissage suivante (modèle *Pipeline*), s'avère intéressante, comme le suggèrent ces résultats sur le corpus TREC.

5.3.2. Un exemple de règle d'extraction

La règle suivante, induite à partir du modèle de R_P pour la relation *located_in,* est spécifiée par le nombre de littéraux, d'exemples positifs couverts, d'exemples négatifs couverts, et sa précision P :

Règle:
#Literals=4, PosScore = 187, NegScore = 19, Prec = 90.8%
located_in(A,B):- t_class(A,loc), t_next(A,B), t_class(B,loc).

La précision élevée de cette règle est principalement due à la présence de plusieurs phrases similaires à « Perugia, Italy » dans le corpus d'apprentissage, indiquant que le premier argument (A) « Perugia » est suivi par (prédicat *t_next*) le second argument (B) « Italy », sans considérer le symbole de ponctuation entre eux.

6. Évaluation comparative

Dans cette section, nous comparons les résultats obtenus par OntoILPER sur le corpus TREC avec ceux obtenus sur le même corpus par deux autres systèmes existants d'extraction d'instances d'entités et de relations mettant en œuvre des méthodes statistiques et n'utilisant pas d'ontologie. Ces deux systèmes, considérés comme les meilleurs systèmes, sont présentés dans (Roth et Yih, 2007) et (Giuliano *et al.,* 2007). A notre connaissance, ce sont les seuls travaux qui utilisent ce corpus pour évaluer des performances en extraction à la fois d'entités et de relations.

6.1 Modèles d'extraction comparés

Pour la comparaison de l'extraction d'entités, nous avons choisi les deux meilleurs modèles de classification trouvés dans (Giuliano *et al.,* 2007) et (Roth et Yih, 2007) :

– Le modèle $\boldsymbol{M_C}$ de (Giuliano *et al.,* 2007) : il suppose que les frontières de l'entité ont déjà été déterminée. Ce modèle correspond au modèle d'apprentissage $\boldsymbol{E_S}$ d'OntoILPER.

– Le modèle ***Separate w/inf*** (Separate with Inference) proposé par (Roth et Yih, 2007) : il utilise une procédure d'inférence globale supplémentaire pour produire la décision finale. Ce modèle correspond aussi au modèle d'apprentissage E_S d'OntoILPER.

– Pour OntoILPER, nous utilisons le modèle ***Es*** comme modèle d'extraction de relation (cf. 5.4.1).

La deuxième comparaison concerne les meilleurs classificateurs de relations présentés dans les mêmes travaux :

– Le modèle $M_O|K_{SL}$ (Giuliano *et al.,* 2007) : il correspond de façon similaire à notre modèle R_O, en ce sens qu'il utilise également les étiquettes d'entité du corpus pendant l'apprentissage.

– Le modèle ***Omniscient w/Inf*** (Roth et Yih, 2007) : c'est le meilleur classificateur de relations informé. Ce classificateur utilise aussi une procédure d'inférence globale pour produire sa décision finale sur la classification de la relation.

– Pour OntoILPER, nous utilisons le modèle R_O qui est son meilleur modèle d'extraction de relation (cf. 5.3.1).

6.2 Résultats obtenus et discussion

Les résultats des classificateurs d'entités sont présentés dans le tableau 5, tandis que les résultats des classificateurs de relations sont présentés dans les tableaux 6a et 6b.

Tableau 5. Résultats comparatifs pour la classification d'entités (modèles séparés)

NER Model	LOC			ORG			PER		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
MC	94.2	**94.4**	**94.3**	91.9	**88.5**	**90.2**	**94.8**	**96.6**	**95.7**
Separate w/Inf	91.8	88.6	90.1	91.2	71.0	79.4	90.6	90.5	90.4
OntoILPER - E_S	**96.0**	88.4	92.0	**97.0**	74.4	84.3	**94.8**	87.5	91.0

Tableau 6. Résultats comparatifs pour la classification de relations (meilleurs modèles)

(a) RE Model	located_in			work_for			orgBased_in		
	P	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
MO|KSL	79.6	76.0	77.8	76.8	80.0	78.4	74.3	77.2	75.7
Omniscient w/Inf	61.9	62.9	59.1	79.2	50.3	61.4	81.7	50.9	62.5
OntoILPER - R_O	**90.5**	**78.6**	**84.0**	**85.7**	**86.1**	**85.8**	**88.7**	**82.5**	**85.4**

(b) RE Model	live_in			kill		
	P	*R*	*F1*	*P*	*R*	*F1*
MO|KSL	78.0	65.8	71.4	82.8	81.0	81.9
Omniscient w/Inf	63.9	57.3	59.9	79.9	81.4	79.9
OntoILPER - R_O	**87.4**	**76.9**	**81.7**	**92.3**	**78.0**	**84.3**

Les résultats sur l'extraction d'instances d'entités (NER) présentés dans le tableau 5 suggèrent que le modèle $\boldsymbol{M_C}$ offre des performances supérieures en termes de score F1. Cependant, les tests de signification statistique – *paired Student t-test* pour la différence entre les scores F1 du modèle $\boldsymbol{E_S}$ et du modèle $\boldsymbol{M_C}$ a révélé qu'il n'y a pas de différence significative à $\alpha = 0{,}05$ (95 % intervalle de confiance) entre eux. Le même résultat se produit lorsque l'on compare le modèle $\boldsymbol{E_S}$ avec le modèle ***w/Inf***. Le tableau 5 montre que le modèle ***OntoILPER-Es*** est plus précis que les autres, mais qu'il a une performance inférieure en rappel. Pour les applications dans lesquelles la précision est plus souhaitable que le rappel, le modèle $\boldsymbol{E_S}$ pourrait être une bonne alternative, car elle évite de surcharger les utilisateurs finaux avec trop exemples faux positifs.

Les résultats du tableau 6 (6a et 6b) sur l'extraction d'instances de relation (ER) montrent que le modèle ***OntoILPER-R_O*** surpasse de façon nette les deux autres. La raison principale est qu'il s'appuie sur un meilleur modèle de représentation de la phrase. Dans notre modèle de phrase sous forme de graphes, tous les types de relations entre les termes dans une phrase sont représentés en utilisant une représentation relationnelle (logique du premier ordre), plus expressif que la représentation basée sur les caractéristiques, de type attribut-valeur, utilisées par les deux autres modèles statistiques considérés. Ainsi les modèles statistiques d'extraction ne peuvent pas saisir efficacement l'information structurelle dérivée de phrases analysées (Zhou *et al,* 2005), contrairement à ce que fait OntoILPER avec son modèle de représentation relationnelle. Comme l'ont montré les expérimentations, cette information structurelle est essentielle pour la performance dans l'extraction de relations.

Ainsi, les résultats expérimentaux comparatifs sur ce corpus TREC tendent à conclure que OntoILPER est plus efficace que les deux autres systèmes statistiques pour l'extraction de relations que d'entités. En fait, OntoILPER repose sur l'outil Stanford NER, dont la performance a été reconnue par rapport à d'autres systèmes d'extraction d'entités (Dlugolinsky *et al.,* 2013). Cependant, même sans utiliser Stanford NER dans son étape de prétraitement sur le corpus TREC, OntoILPER surpasse le modèle d'extraction d'entités basé sur les caractéristiques proposé par Roth et Yih (2007). Le système d'extraction d'entités, proposé par Giuliano *et al.* (2007) et basé sur les CRF, utilise un répertoire toponymique (*gazetter*) de lieux (locations), de noms et des organisations de personnes, dans sa phase de prétraitement, ce qui a certainement un impact positif sur ses performances.

Cependant, des tests de signification statistique ont montré que OntoILPER est comparable au système proposé par Giuliano *et al.* (2007).

7. Conclusion

Le développement de systèmes d'extraction d'information constitue toujours un grand défi, notamment dans le contexte du web. Pour être plus précis, de tels systèmes d'IE doivent exploiter de plus en plus de ressources sémantiques disponibles, notamment des ontologies. Pour être plus rapidement développés et adaptables à d'autres domaines d'application, ils doivent utiliser des techniques d'apprentissage automatique.

La plupart des techniques d'extraction d'entités et de relations actuellement utilisées sont basées sur les méthodes d'apprentissage automatique statistique. L'approche retenue dans cette recherche a été d'utiliser plutôt des méthodes d'apprentissage automatique symbolique. Le premier intérêt de cette approche est de placer les processus tant d'extraction d'information que de traitement de cette dernière (notamment le peuplement d'une ontologie) sur un même niveau sémantique, qui est le niveau symbolique de la logique du premier ordre, mais aussi celui du web sémantique avec toutes ses ressources sémantiques et ses mécanismes de raisonnement automatique disponibles. Le deuxième intérêt de cette approche et qu'elle est ouverte, extensible à la prise en compte de nouvelles ressources sémantiques, et par là même adaptative à tout domaine applicatif nouveau. Le troisième intérêt et non le moindre, est que les règles d'extraction induites par cette approche sont compréhensibles et modifiables par des experts humains.

Dans ce contexte nous avons présenté un système d'EI, nommé OntoILPER, permettant d'extraire des instances d'entités et de relations à partir de données textuelles, ceci par l'usage d'ontologies et de programmation logique inductive (PLI), une technique d'apprentissage machine symbolique. Ce système utilise un espace d'hypothèses relationnelles (prédicats binaires) plus expressif, et exploite une ontologie de domaine, ainsi que d'autres connaissances complémentaires représentées également de façon relationnelle, pour représenter les exemples et induire des règles d'extraction symboliques.

Plusieurs expérimentations d'OntoILPER sur le corpus de référence TREC ont permis de comparer ses performances avec celles de systèmes d'extraction statistiques, et montrer sa supériorité dans l'extraction de relations. Afin d'évaluer plus complètement OntoILPER, plusieurs expériences en utilisant d'autres corpus de référence sont en cours. Comparés à d'autres systèmes d'extraction d'information, les résultats obtenus avec OntoILPER sont dès à présent très encourageants. Il reste cependant à améliorer les performances du système en agissant sur chacun de ses composants.

Concernant le composant de prétraitement des textes, le système actuel repose sur une analyse syntaxique de surface des phrases, et ne tient pas compte des aspects sémantiques relatifs aux entités de verbes. Aussi nous voulons intégrer davantage de connaissances de base (ressources sémantiques) dans OntoILPER étape de

prétraitement, comme les synonymes, les hyperonymes/hyponymes, l'étiquetage de rôle sémantique (*Semantic Role Labelling*), et la désambiguïsation. Pour le composant d'apprentissage (PLI), en raison d'une répartition inégale entre le nombre d'exemples positifs et négatifs trouvés dans la majorité des corpus utilisés dans les expérimentations, nous voulons étudier l'impact des techniques d'échantillonnage principalement des techniques *undersampling* qui permettraient d'accélérer la tâche d'apprentissage dans OntoILPER en réduisant le nombre d'exemples négatifs générés. Une autre perspective est de d'utiliser OntoILPER pour faire de l'extraction d'événements, c'est-à-dire l'extraction de relations non plus binaires mais n-aires.

Enfin, nous souhaitons pouvoir utiliser OntoILPER avec d'autres langues que l'anglais pour laquelle d'importantes ressources de traitement existent, notamment le français et le portugais. Le composant de prétraitement de texte est alors à reconsidérer avec de nouvelles ressources de TAL disponibles à identifier et intégrer.

Remerciements

Les auteurs remercient le Conseil national de développement scientifique et technologique du Brésil (CNPq) pour son soutien financier (Grant N°140791/2010-8).

Bibliographie

Airola A., Pyysalo S., Björne J., Pahikkala T., Ginter F., Salakoski T. (2008). All-paths graph kernel for protein–protein interaction extraction with evaluation of cross corpus learning, *BMC Bioinformatics*, 9:S2.

Alphonse E., Rouveirol C. (2000). Lazy propositionalisation for relational learning. In Horn W. (ed.). *14th European Conference on Artificial Intelligence (ECAI'2000),* Berlin, Germany, pp. 256-260, IOS Press.

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

Baeza-Yates R., Ribeiro-Neto B. (1999). *Modern Information Retrieval*. Addison-Wesley.

Choi S-P, Jeong C-H, Choi Y-S, Myaeng S-H (2009). Relation extraction based on extended composite kernel using flat lexical features. *JKIISE: Software Application*, 36:8.

Choi S. P., Lee. S., Jung H., Song S. (2013). An intensive case study on kernel-based relation extraction. Proceedings of *Multimedia Tools and Applications*, Springer, US, p. 1-27.

Culotta A., Sorensen J. (2004). Dependency tree kernels for relation extraction. *ACL'2004*, p. 423-429. 21-26 July 2004. Barcelona, Spain.

De Raedt L. (2010). Inductive Logic Programming. *Encyclopedia of Machine Learning,* p. 529-537.

Dipper S., Götze M., Küssner U., and Stede M. (2007). Representing and querying standoff XML. Proceedings of the *GLDV-Frühjahrstagung 2007*, Tübingen, Germany.

Dlugolinský S., Ciglan M., Laclavík M. (2013). Evaluation of Named Entity Recognition Tools on Microposts (2013). *INES 2013, 17th IEEE International Conference on Intelligent Engineering Systems*. Budapest p. 197-202.

Erk K., Pado S. (2004). A Powerful and Versatile XML Format for Representing Role-semantic Annotation. *LREC 2004*, Lisbon, Portugal.

Ehrmann M. (2008). *Les entités nommées, de la linguistique au TAL : statut théorique et méthodes de désambiguïsation*, Thèse de Doctorat, Université Paris 7 – Denis Diderot.

Finn A. (2006). *Multi-Level Boundary Classification Approach to Information Extraction*, Phd thesis, University College Dublin.

Fürnkranz J., Gamberger D., Lavrac N. (2012). *Foundations of Rule Learning*, Springer-Verlag.

Giuliano C., Lavelli A., Romano L. (2007). Relation Extraction and the Influence of Automatic NER, *ACM Transactions on Speech and Language Processing*, vol. 5, n° 1, ACM.

Gruber T. (1993). Towards Principles for the Design of Ontologies used for Knowledge Sharing. Int. *Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Deventer, The Netherlands.

Hitzler P., Krötzsch M., Parsia B., Patel-Schneider P.F., Rudolph S. (editors) (2009). OWL 2 Web Ontology Language Primer. W3C Working Draft, http://www.w3.org/TR/owl2-primer/

Horváth T., Paass G., Reichartz F., Wrobel S. (2009). A Logic-based Approach to Relation Extraction from Texts. *ILP 2009*: 34-48, Leuven, Belgium.

Jiang J. (2012). Information Extraction from Text, in C.C. Aggarwal and C.X. Zhai (eds), *Mining Text data*, chap. 2, p. 11-41.

Jiang J., Zhai C. X. (2007). A systematic exploration of the feature space for relation extraction. *Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT'2007*, Rochester, NY, USA.

Kambhatla N. (2004). Combining lexical, syntactic and semantic features with Maximum Entropy models for extracting relations. *ACL'2004* (Poster), 21-26 July 2004, Barcelona, Spain, p. 178-181.

Karkaletsis V., Fragkou P., Petasis G., and Iosif E. (2011). Ontology Based Information Extraction from Text. Paliouras G. et al. (Eds.) *Multimedia Information Extraction*, LNAI 6050, p. 89-109.

Kinoshita S., Cohen K. B., Ogren P., and Hunter L. (2005). BioCreAtIvEtask 1A: Entity identification with a stochastic tagger. *BMC Bioinformatics,* 6(Suppl 1):S4.

Kruijff G. J. M. (2002). *Formal and Computational Aspects of Dependency Grammar: History and Development of DG*, Tech. report, ESSLLI.

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

Lima R., Batista J., Ferreira R., Freitas F., Lins R., Simske S., Riss M. (2014). Improving Relation Extraction through the Simplification of Graph-based Representations of

Sentences. *Proceedings of the 14th ACM Symposium on Document Engineering* (DocEng 2014), September 16-19, Denver, Colorado, USA.

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

Lima R., Espinasse B., Oliveira H., Freitas F. (2014). Ontology Population from the Web: an Inductive Logic Programming-Based Approach. *Proceedings of the 11th International Conference on Information Technology:* New Generations, ITNG 2014, Las Vegas, Nevada, USA.

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

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, ICTAI 2015*, Vietri sul Mar, Italy.

De Marneffe M-C., Manning C. D. (2008). *Stanford Dependencies Manual*. http://nlp.stanford.edu/software/stanford-dependencies.shtml

Mitchel T. (1982). Generalization as Search. *Artificial Intelligence* 18, p. 203-226.

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

Muggleton S. (1995). Inverse entailment and Progol. *New Generation Computing*, 13, p. 245-286).

Muggleton S., Fen C. (1990). Efficient induction of logic programs. *1st Conference on Algorithmic Learning Theory* (pp. 368-381), Tokyo, Japan.

Muggleton S., Santos J., Tamaddoni-Nezhad A. (2009). ProGolem: a system based on relative minimal generalisation. *19th International Conference on ILP,* Springer, p. 131-148, Leuven, Belgium.

Nazarenko A., Nédellec C., Alphonse E., Aubin S., Hamon T., and Manine A.-P. (2006). Semantic annotation in the Alvis project. In W. Buntine and H. Tirri, editors, Proceedings of the *International Workshop on Intelligent Information Access*, pages 40–54, Helsinki, Finlande.

Nédellec C., Rouveirol C., Adé H., Bergadano F. et Tausend B (1996). Declarative Bias in ILP. *In Advances in Inductive Logic Programming*, p. 82-103, De Raedt L. (Ed.), IOS Press.

Nédellec C., Nazarenko A. (2005). *Ontologies and Information Extraction*. LIPN Internal Report.

Nédellec C., Nazarenko A., Bossy R. (2008). Information Extraction. In: Staab, S., Studer, R. (editors). *Ontology Handbook.* Springer, Heidelberg.

Okanohara D., Miyao Y., Tsuruoka Y., and Tsujii J. (2006). Improving the scalability of semi-Markov conditional random fields for named entity recognition. Proc. of the *21st International Conf. on Computational Linguistics and the 44th Annual Meeting of the ACL*, p. 465-472.

Patel A., Ramakrishnan G., Bhattacharya P. (2010). Incorporating Linguistic Expertise Using ILP for Named Entity Recognition in *Data Hungry Indian Languages*, LNCS, vol. 5989, p. 178-185, Springer Berlin Heidelberg.

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, p. 134-166.

Plotkin G. (1971). A note on inductive generalization. *Machine Intelligence* 5, p. 153-163.

Rabiner L. R. (1089). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77, n° 2, p. 257-287.

Ramakrishnan G., Joshi S., Balakrishnan S., Srinivasan A. (2008). Using ILP to Construct Features for Information Extraction from Semi-structured Text. In Proceedings of the *17th International Conference on Inductive Logic Programming*, LNAI 4894, p. 211-224, Berlin, Springer.

Roth D., Yih W. (2007). Global Inference for entity and relation identification via a linear programming formulation. In Introd. to *Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press.

Roth D., Yih W. (2004). A Linear Programming Formulation for Global Inference in Natural Language Tasks. Proc. of *CoNLL-2004*, Boston, MA, USA.

Saggion H., Funk A., Maynard D., Bontcheva, K. (2007). Ontology-based Information Extraction for Business Intelligence, *ISWC'07/ASWC'07*, Busan.

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

Seneviratne M. D. S., Ranasinghe D. N. (2011). Inductive Logic Programming in an Agent System for Ontological Relation Extraction, *International Journal of Machine Learning and Computing*, vol. 1, n° 4, p. 344-352.

Shen D., Zhang J., Zhou G., Su J., and Tan C.-L. (2003). Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In Proc. of the *ACL 2003 Workshop on NLP in Biomedicine*, vol. 13, p. 49-56, Sapporo, Japan.

Smole D., Ceh M. and Podobnikar T. (2011). Evaluation of inductive logic programming for information extraction from natural language texts to support spatial data recommendation services. *International Journal of Geographical Information Science*, 25, p.1809-1827.

Tang J., Hong M., Zhang D., Liang B., and Li J. (2007). Information Extraction: Methodologies and Applications. In *The book of Emerging Technologies of Text Mining: Techniques and Applications*, Hercules A. Prado and Edilson Ferneda (Ed.), Idea Group Inc., Hershey, USA, p. 1-33.

Wimalasuriya D. C., Dou D. (2009). Ontology-Based Information Extraction: An Introduction and a Survey of Current Approaches, *Journal of Information Science*, JIS-0987-v4, 2009, p. 1-20.

Zhang M., Zhou G.D., Aw A.T. (2008). Exploring syntactic structured features over parse trees for relation extraction using kernel methods, *Information Processing and Management*, 44, p. 687-701.

Zhao S. B., Grisman R. (2005). Extracting Relations with Integrated Information using Kernel Methods. *ACL'2005*, 25–30 June 2005, Ann Arbor, USA, p. 419-426.

Zhou G. D., Su J., Zhang J., Zhang M. (2005). Exploring various knowledge in relation extraction, in: Proceedings of the *Annual Meeting of the Association for Computational Linguistics - ACL'2005,* 25-30 June 2005, Ann Arbor, Michigan, USA.

Zhou G., Zhang M., Ji D-H., Zhu Q. (2007). Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague.