TALN 2010, Montréal, 19–23 juillet 2010

Acquisition de grammaires locales pour l'extraction de relations entre entités nommées

Mani EZZAT[1, 2]
(1) Er-Tim, Inalco, 75343 Paris
(2) Arisem, Thales, 91300 Massy
mani.ezzat@arisem.com

Résumé. La constitution de ressources linguistiques est une tâche cruciale pour les systèmes d'extraction d'information fondés sur une approche symbolique. Ces systèmes reposent en effet sur des grammaires utilisant des informations issues de dictionnaires électroniques ou de réseaux sémantiques afin de décrire un phénomène linguistique précis à rechercher dans les textes. La création et la révision manuelle de telles ressources sont des tâches longues et coûteuses en milieu industriel. Nous présentons ici un nouvel algorithme produisant une grammaire d'extraction de relations entre entités nommées, de manière semi-automatique à partir d'un petit ensemble de phrases représentatives. Dans un premier temps, le linguiste repère un jeu de phrases pertinentes à partir d'une analyse des cooccurrences d'entités repérées automatiquement. Cet échantillon n'a pas forcément une taille importante. Puis, un algorithme permet de produire une grammaire en généralisant progressivement les éléments lexicaux exprimant la relation entre entités. L'originalité de l'approche repose sur trois aspects : une représentation riche du document initial permettant des généralisations pertinentes, la collaboration étroite entre les aspects automatiques et l'apport du linguiste et sur la volonté de contrôler le processus en ayant toujours affaire à des données lisibles par un humain.

Abstract. Building linguistics resources is a vital task for information extraction systems based on a symbolic approach : cascaded patterns use information from digital dictionaries or semantic networks to describe a precise linguistic phenomenon in texts. The manual elaboration and revision of such patterns is a long and costly process in an industrial environment. This work presents a semi-automatic method for creating patterns that detect relations between named entities in corpora. The process is made of two different phases. The result of the first phase is a collection of sentences containing the relevant relation. This collection isn't necessairly big. During the second phase, an algorithm automatically produces the recognition grammar by generalizing the actual content of the different relevant sentences. This method is original from three different points of view : it uses a rich description of the linguistic content to allow accurate generalizations, it is based on a close collaboration between an automatic process and a linguist and, lastly, the output of the acquisition process is always readable and modifiable by the end user.

Mots-clés : relation, entité nommée, grammaire.

Keywords: relation, named entity, pattern.

1 Introduction

Les *grammaires locales* (Silberztein, 1993) sont des ressources indispensables aux systèmes de recherche d'information fondés sur une approche symbolique. Elles permettent de formaliser un phénomène linguistique et de le détecter dans les textes. Leur principal avantage est la possibilité de révision : elles sont lisibles et manipulables par un linguiste, ce qui n'est pas le cas pour les systèmes à base d'apprentissage statistique. L'amélioration incrémentale des résultats est une contrainte importante dans un contexte industriel. La plupart des travaux similaires (section 3) produisent une liste importante de patrons, généralement lexico-syntaxiques, dans laquelle il est difficile de naviguer.

Afin de répondre à ces contraintes, nous proposons une méthode semi-automatique d'acquisition de grammaires pour l'extraction de relations en deux phases. La première consiste en la récolte de quelques segments de textes attestant la relation recherchée. Puis à partir de cette collection de segments, un algorithme génère une grammaire présentée sous la forme d'un unique transducteur à nombre fini d'états, facilitant alors sa maintenance par le linguiste. Enfin, nous présentons les résultats de l'expérience sur un corpus à travers un cas concret : la relation de *contact*. L'originalité de l'approche repose sur trois aspects : une représentation riche du document initial permettant des généralisations pertinentes, la collaboration étroite entre les aspects automatiques et l'apport du linguiste et sur la volonté de contrôler le processus en ayant toujours affaire à des données lisibles par un humain.

Dans la suite de cet article nous présentons en premier lieu ce que nous entendons par le terme *relation* (section 2), puis nous présenterons les travaux similaires, notamment ceux qui alimentent les systèmes fondés sur une approche symbolique (section 3). Après une description de notre méthodologie (section 4), nous présentons nos résultats d'expérience (section 5). Enfin, nous concluons par une discussion sur les résultats et les perspectives de recherche (section 6).

2 Définition

L'extraction de relations entre entités nommées n'est pas un problème nouveau et a été formalisée officiellement pour la première fois en tant que tâche indépendante et réutilisable lors de la conférence *Message Understandig Conference* de 1998 (MUC, 1998). Le but est de détecter des relations entre entités nommées et de structurer les résultats afin d'alimenter une base de données. Plus tard, les travaux motivés par la campagne *Automatic Content Extraction* (ACE, 2004) ont fait émerger une définition dont nous nous inspirons ici. Nous appelons *relation*, un lien significatif entre entités nommées explicité dans un texte. Nous distinguons deux types de relations :

1. Les *relations statiques* ou *faits* représentent essentiellement des états. Ce qu'on appelle état se caractérise par l'absence de changement. Un état qui est vrai pour un intervalle donné est vrai pour tout point de cet intervalle. C'est donc un lien stable et avéré entre deux entités nommées.

   exemple : *Arisem* est une filiale du *Groupe Thales*. (1)

2. Les *évènements* peuvent être assimilés à une phrase d'action et mettent en cause plusieurs entités (l'acteur, la cible et l'évènement particulier qui est défini par le prédicat et ses arguments par exemple), qui apportent une information nouvelle sur les participants et qui peuvent avoir une localisation spatio-temporelle implicite ou non.

   exemple : Le *groupe Thales* a racheté *Arisem* en *Mars 2004*. (2)

Les relations entre entités nommées sont généralement *n-aires* et nous distinguons deux types de constituants : les *arguments* et les *circonstants*. Les arguments sont les entités nommées nécessaires à l'existence de chaque instance. Les circonstants, quant à eux, sont des éléments optionnels qui ne sont pas indispensables à la compréhension et à la complétude de l'énoncé et représentent généralement une localisation, une date ou encore une expression numérique. Par exemple dans (2), les entités nommées *groupe Thales* et *Arisem* sont ici les arguments de l'évènement sans lesquels la relation n'existerait pas, tandis que *Mars 2004* est un circonstant, qui n'est pas obligatoire pour l'instanciation de la relation.

3 Travaux similaires

Depuis MUC 1998, la plupart des travaux se sont concentrés autour des approches statistiques, car elles ne nécessitent pas de développement manuel de ressources et obtiennent de bons résultats. Différents modèles sont utilisés, des SVM (*Support Vector Machine*) (Zelenko *et al.*, 2003) (Zhao & Grishman, 2005) aux CRF (*Conditional Random Field*) (Zhang *et al.*, 2008). Cependant, toutes ces approches reposent sur la disponibilité de corpus annotés et elles apparaissent comme de véritables "boîtes noires" : l'intervention du linguiste dans l'analyse reste difficile et ce sont essentiellement certains paramètres qui peuvent être manipulés afin d'améliorer les résultats du système.

Parallèlement, du milieu des années 90 à aujourd'hui, des études traitent de la génération de grammaires pour l'extraction de relations. Certains travaux utilisent un algorithme de généralisation ascendante (Soderland *et al.*, 1995), (Califf & Mooney, 2003). Il s'agit de relâcher les contraintes des grammaires qui décrivent principalement des éléments lexicaux dans un premier temps. On généralise ensuite cette description à différents niveaux (morpho-syntaxiques, sémantiques) en unifiant les patrons. Une nouvelle règle est créée en fusionnant deux règles existantes.

A l'inverse, d'autres travaux utilisent des algorithmes descendants pour spécifier les patrons. Le système *AutoSlog* (Riloff, 1996) précise des schémas syntaxiques simples comme "*sujet - verbe à la voix passive*" et les instancie avec des éléments du domaine après analyse. Par exemple, ce schéma, dans un corpus sur le terrorisme, pourra devenir "<*victim*> *was murdered*". Les patrons fournis ne décrivent généralement pas de longues dépendances, comme c'est souvent le cas pour les relations entre entités nommées. D'une manière similaire, le composant LIEP (Huffman, 1996) extrait des patrons lexicaux à partir de schémas simples dont le coeur est une liste de mots clés, puis analyse le contexte afin de trouver les éléments en relation syntaxique.

Enfin, le système *Sem+*[1] (Goujon, 2008) utilise un algorithme issu de la terminologie (Hearst, 1992). Il s'agit d'un processus itératif auquel on donne des couples d'entités nommées que l'ont sait en relation, afin d'extraire des phrases où cette relation apparaît. Après avoir inféré des patrons lexicaux à partir de ces phrases, le système les applique au corpus afin de trouver de nouveaux candidats qui servent de données d'entrée à une nouvelle itération. Le système produit un patron par phrase, contraint sur le lexique qui la compose. Il en résulte un nombre important de patrons difficiles à maintenir et dont le rappel peut chuter sur un autre corpus.

La plupart de ces travaux nécessitent que l'utilisateur fournisse en amont une liste de patrons. De plus, ces systèmes produisent des grammaires présentées sous la forme de liste importante de patrons qui restent très difficiles à maintenir.

[1] Sem+ est un outil de Thalès Research and Technology avec qui Arisem collabore

4 Méthodologie

Notre méthode génère une grammaire à partir de segments de textes extraits d'un corpus et attestant la relation recherchée. Elle se divise en deux phases distinctes. La première consiste en la sélection de segments représentatifs de la relation recherchée. Cette étape n'est pas entièrement automatique et requiert l'assistance du linguiste. Il s'agit de compter les cooccurrences entre entités nommées puis de présenter les résultats de manière lisible, afin que le linguiste puisse rapidement sélectionner une relation qui l'intéresse et les segments de texte qui en attestent. Cette collection de segments vient ensuite alimenter un algorithme, basé sur un principe de curseur. Cette technique consiste à lire les phrases de gauche à droite et à généraliser au fil de l'eau l'information contenue dans les phrases. Le processus produit un unique transducteur à nombre fini d'états, au fur et à mesure des résultats renvoyés par l'analyse de chaque segment.

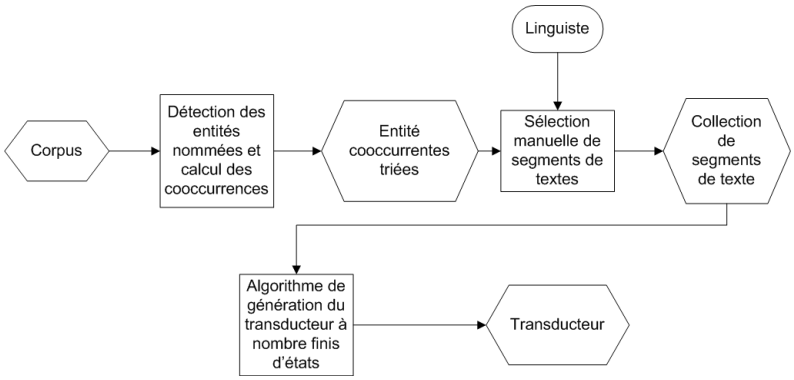


FIG. 1 – *Schéma général de la méthode*

4.1 Sélection des segments de textes

La première étape a pour but la sélection de segments de texte qui relèvent d'un évènement particulier. Après avoir détecté les entités nommées avec l'analyseur d'*Arisem*, nous comptons le nombre de phrases dans lesquelles le même groupe d'entités nommées apparaît. Ce processus ne se restreint pas à 2, mais à *n* entités présentes dans la même phrase.

Personne	***Personne***	***Lieu***	***Date***	***Nb***	***Liens***
Gnassingbé Eyadéma	Gbagbo	Lomé	Jeudi	11	Lien1, Lien2...
Olusegun Obasanjo	Guillaume Soro	Paris	30 Avril	11	Lien1, Lien2...
Carlo Azeglio Ciampi	Laurent Gbagbo	Vatican	mercredi	11	Lien1, Lien2...
Chirac	Kofi Annan			10	Lien1, Lien2...

TAB. 1 – *Echantillon de la sortie après calcul des cooccurrences*

Le résultat (tableau 1) est présenté sous la forme d'un tableau listant les ensembles d'entités nommées cooccurrentes, classés par leur nombre, avec des liens pointant sur les phrases où elles apparaissent. Le

linguiste peut alors explorer ce résultat afin de choisir rapidement une relation intéressante par rapport au corpus et sélectionner les segments de textes qui en attestent. Pour cette expérience, nous avons retenu la relation de *contact*, définie comme la rencontre explicite entre deux personnes dans un texte.

Parmi les segments sélectionnés, nous avons gardé uniquement ceux basés sur un prédicat verbal. L'algorithme qui génère le transducteur obéit à des règles établies en fonction du type de construction syntaxique. Dans le cadre de cet article, nous nous sommes restreint aux prédicats verbaux car ils composent la majorité des segments de textes représentant une relation. Le nombre de segments sélectionnés ne doit pas être nécessairement important.

4.2 Description de l'algorithme de génération de la grammaire

L'algorithme de génération de la grammaire à partir des exemples procède en deux étapes. Les phrases sont tout d'abord "décorées" avec différentes annotations linguistiques (liés à la morphosyntaxe, à la syntaxe de surface et à la sémantique) grâce à un analyseur développé chez Arisem. Une généralisation est ensuite opérée à partir des annotations pour produire une grammaire à partir d'un algorithme de type *shift-reduce* (Aho & Ullman, 1972).

4.2.1 Description des annotation fournies par l'analyseur Arisem

Nous utilisons l'analyseur d'*Arisem*, fondé sur une approche symbolique, afin de généraliser le niveau de la description de la grammaire générée à partir des segments de texte issus de la première étape. L'analyseur décore le texte avec différents niveaux d'étiquettes, morphologiques, syntaxiques et sémantiques sur les mots et les syntagmes pertinents.

En complément des étiquettes morpho-syntaxiques apportées par les dictionnaires, nous avons utilisé des grammaires pour la détection des entités nommées. Nous avons également ajouté une grammaire de détections de groupes nominaux (déterminants exclus) ainsi qu'une grammaire de reconnaissance de locutions verbales ("*aller voir*" par exemple) et de verbes munis d'un auxiliaire. Cette analyse syntaxique de surface nous permet de reconnaître de longues séquences de mots, avec des contraintes suffisantes pour ne pas générer de bruit.

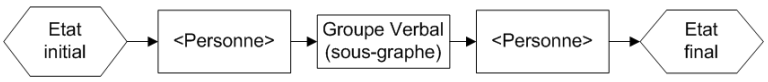
4.2.2 Description de l'algorithme de production de la grammaire

On peut assimiler l'algorithme de production de la grammaire à un algorithme de type *shift reduce* (Aho & Ullman, 1972; Soricut & Marcu, 2003) : les phrases sont examinées une à une, de gauche à droite. Les mots sont lus les uns après les autres (*shift*) et réduits à une étiquette donnée quand une règle de généralisation peut être appliquée (*reduce*). Un graphe (la grammaire résultat) est généré en parallèle à partir des généralisations ; cette génération repose sur deux opérations essentielles : unification du noeud en cours d'examen avec un noeud compatible déjà existant dans la grammaire générée, ou création d'un nouveau noeud si aucune unification n'est possible.

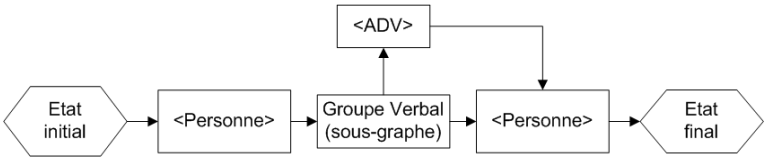
Exemples :

1. Soit la phrase "*Sarkozy a rencontré Chirac*"

L'analyseur fournit (parmi d'autres) les étiquettes suivantes :
– Sarkozy → `Personne`
– Chirac → `Personne`
– a rencontré → `Groupe Verbal`
S'il s'agit de la première phrase analysée, la grammaire générée est vide. A partir de cette phrase, l'algorithme va produire la grammaire suivante sous la forme d'un graphe :



2. Si l'analyseur reçoit à présent la phrase "*Sarkozy a rencontré hier Chirac*"
La même analyse se met en place ; *Sarkozy* peut s'unifier avec le premier noeud de la grammaire `<Personne>` ; idem pour le verbe. En revanche, *hier* ne peut s'unifier avec `<Personne>`. L'algorithme crée donc une nouvelle ramification avec un nouveau noeud portant l'étiquette `<ADV>` entre `<Groupe Verbal>` et `<Personne>`.



Règles de généralisation

Voici, de façon plus systématique, l'ensemble des généralisations opérées à partir des informations fournies par l'analyseur d'Arisem.

– Les entités nommées sont typées avec leur étiquette correspondante :"*Chirac*" → `<Personne>`, "*Microsoft*" → `<Organisation>` etc...
– Les groupes nominaux qui ne sont pas des entités nommées sont généralisés en tant que `<GN>` : "les rebelles ivoiriens" → `<GN>`.
– Les groupes verbaux ont un statut particulier. La sémantique de la relation étant portée par celui-ci, nous les généralisons en construisant un nouveau graphe en cascade, avec des noeuds représentant un retour au lemme de tous les éléments qui composent le segment original (figure 2). Ainsi le segment *ira voir* détecté en tant que groupe verbal, produira le sous-graphe `[<aller>] → [<voir>]`.
– Enfin, tous les mots restants ne rentrant pas dans ces catégories sont généralisés par leur étiquette morpho-syntaxique : "*et*" → `<CONJC>` (conjonction de coordination),"*pour*" → `<PREP>` (préposition),'*le*"→> `<DET>` (déterminant) etc...

4.3 Résultat et formalisation

Le résultat est assimilable à un graphe conceptuel (Sowa, 1984), où le prédicat est le centre et les principaux arguments sont liés par des relations typées. Les règles de généralisation opèrent sur des éléments de natures diverses sur la base de la compatibilité de type (on peut généraliser "*Chirac*" en `<Personne>` puis en `<Entité>` parce que ces différents éléments sont des généralisations compatibles, autrement dit

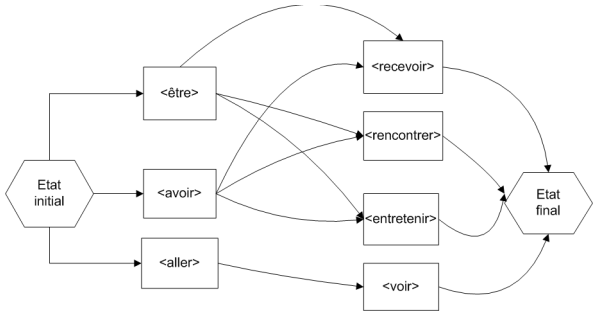


FIG. 2 – *Exemple de transducteur en cascade pour les groupes verbaux*

parce qu'on a affaire à une relation d'hyponymie/hyperonymie entre ces différents éléments, modulo le fait que *Chirac* est une instance et non un type).

De même, la génération de la grammaire est comparable à la jointure dans les graphes conceptuels (Sowa, 1984), où les noeuds compatibles sont fusionnés, et où les autres créent de nouvelles ramifications dans le graphe résultat. La différence essentielle réside dans l'analyse sous-jacente au moyen d'un algorithme de type *shift reduce* qui se fonde sur l'ordre linéaire de la phrase. Mais même dans les premières implémentations des graphes conceptuels (Fargues *et al.*, 1986), de tels mécanismes avaient dû être mis en place pour limiter le pouvoir expressif de la jointure souvent trop puissante pour des applications en langue naturelles (intuitivement, on souhaite tenir compte de l'ordre des éléments tout simplement parce que la syntaxe est déterminante pour la compréhension). Certaines jointures sont guidées ici de façon à ce que le graphe respecte la construction des phrases analysées, représentée ici par un verbe présent entre les deux arguments de la relation.

Nous autorisons les *retours-arrière* dans le graphe. Ils opèrent seulement dans une fenêtre précise, délimitée par les deux arguments et le verbe. Il y a donc 4 fenêtres définies (avant, après et entre les arguments et le verbe). Nous unifions les noeuds compatibles par un *retour-arrière*, si et seulement s'ils apparaissent dans la même fenêtre. Autrement dit, si deux déterminants sont présents dans la même fenêtre, ils seront unifiés par le même noeud.
Ces *retours-arrière* guidés rendent le graphe plus lisible car ils limitent le nombre de création de noeuds. Cependant, ils augmentent la combinatoire des chemins possibles dans le graphe et peuvent engendrer du bruit.

Nous partons donc d'une base d'analyse assez traditionnelle. Notre objectif principal est de l'opérationnaliser dans un cadre industriel avec les contraintes que nous avons mentionnées dans l'introduction : nécessité de garder un processus lisible, modifiable par un linguiste et facilement adaptable à un nouveau domaine.

5 Protocole d'expérimentation et résultats

Nous avons utilisé deux corpus pour l'expérimentation, un corpus d'acquisition et un corpus de test. Le corpus d'acquisition nous a servi pour inférer la grammaire, le second pour l'évaluation présentée dans cette section.

5.1 Élaboration de la grammaire à partir du corpus d'entraînement

Le corpus d'acquisition est composé d'environ 7 millions de mots répartis sur 13 500 dépêches AFP portant sur la crise en Côte d'Ivoire. Une dépêche ne dépasse généralement pas les 8 phrases et le genre textuel est journalistique. Comme expliqué *supra*, il faut d'abord sélectionner un ensemble de phrases pertinentes par rapport à la relation pour pouvoir ensuite lancer le processus de génération de la grammaire de reconnaissance.

Lors de la sélection de segments de textes, nous remarquons que les nombres de cooccurrences les plus importantes (jusqu'à 200 phrases contenant le même ensemble d'entités nommées) relèvent de relations statiques. Les cooccurrences avec un nombre moyen, situé entre 5 et 15, sont composées généralement d'évènements. Au dessous de 5, le nombre d'ensembles d'entités cooccurrentes très important rend difficile l'exploration. Nous avons observé que la relation de *contact* était la plus présente dans les scores moyens et nous avons sélectionné un ensemble de 98 segments qui en attestent. Parmi ces 98 segments, nous n'avons gardé que les 54 qui sont présentés sous la forme d'une construction autour d'un prédicat verbal.

L'algorithme de génération est produit automatiquement à partir de ces 54 phrases pertinentes. On remarquera la très faible taille de l'échantillon d'apprentissage, caractéristique de ce type d'applications en milieu industriel. De fait, les techniques d'apprentissage demandant de grosses masses de données pour l'apprentissage ne sont pas utilisables dans ce contexte.

5.2 Evaluation de la grammaire

Nous utilisons le second corpus pour évaluer la grammaire produite : il s'agit d'un échantillon de l'année 2007 du journal *Le Monde* composé d'environ 639 500 mots. 227 segments attestant la relation de *contact* ont été annotés manuellement ; nous avons volontairement utilisé un corpus d'un genre textuel proche du corpus d'acquisition pour l'évaluation (corpus de type journalistique) mais nous avons tenu à modifier la source afin de ne pas être trop proche du corpus original. Le reste de la section présente uniquement les résultats obtenus à partir du corpus de test.

Pour l'étape de sélection des segments de textes, nous n'utilisons pas de métriques car la sélection des segments de textes est effectuée manuellement (à partir de l'analyse automatique des cooccurrences toutefois). Pour le transducteur généré, nous utilisons les mesures de précision et rappel, calculé à partir des segments annotés dans le corpus d'évaluation.

Nous appliquons la grammaire générée sur le corpus de test et nous calculons le rappel (tableau 2) selon trois cas distincts.

1. En prenant en compte tous les segments annotés dans le corpus.
2. En ne gardant que les segments annotés construits autour d'un prédicat verbal.
3. En ne gardant qu'un sous-ensemble des segments annotés construits autour d'un prédicat verbal sans anaphores ou coréférences des entités nommées du type *personne*.

La précision ne varie pas car la grammaire reconnaît soit des segments qui correspondent au critère 3 (le plus petit des ensembles), soit des segments qui ne correspondent à aucun des trois critères. Si une petite partie du bruit généré provient des grammaires utilisées dans l'analyseur (mauvaise détection d'entités

Segments considérés	***Rappel***	***Précision***
Tous segments confondus	36.5%	84.6%
Prédicats verbaux	55.3%	84.6%
Prédicats verbaux sans coréférences	**83.8%**	84.6%

TAB. 2 – *Estimation du rappel et de la précision*

nommées, de groupes nominaux et verbaux), la majeure partie est due aux *retours arrière* du transducteur, notamment sur le noeud représentant un groupe nominal. En effet, celui-ci ne fait l'objet d'aucune règle de jointure, engendrant alors une forte augmentation de la combinatoire des séquences reconnues.

Le silence est en partie dû aux différences textuelles des corpus. Le corpus d'évaluation contient des phrases et des textes généralement plus longs qui peuvent contenir des incises. D'autre part, on observe également que la longueur des textes du corpus d'évaluation force l'emploi d'anaphores et de coréférences, qui composent près d'un tiers des segments annotés manuellement. Leur résolution peut améliorer de manière importante le rappel (environ 30%).

On remarque également que le rappel est relativement correct si on considère la petite taille de l'échantillon de segments de textes (au nombre de 54) à partir duquel est produite la grammaire. Les règles de généralisation et de jointure relâchent suffisamment les contraintes sans pour autant augmenter le bruit d'une manière importante.

6 Discussion et conclusion

Nous avons présenté une méthode semi-automatique pour la création de grammaires de détection de relations entre entités nommées. Si les performances sont honorables en terme de précision et rappel, il est en revanche difficile d'objectiver le réel gain apporté. En effet, notre méthode a un double objectif. D'une part, elle doit faciliter l'édition de la grammaire par un linguiste en la rendant plus lisible à travers un unique transducteur à nombre fini d'états, et d'autre part, elle doit également permettre un gain de temps important par rapport à la constitution manuelle d'une telle ressource. Ces deux critères, récurrents dans les systèmes d'extraction d'informations fondés sur une approche symbolique, sont particulièrement difficiles à évaluer. Par ailleurs, la manière de fusionner les noeuds du graphe est guidée par le type de construction syntaxique des phrases analysées. Ce qui implique un changement de règles selon ce type. Nous avons seulement expérimenté l'algorithme dans le cas de relations entre entités nommées basées sur un prédicat verbal. Mais il faut définir autant de règles qu'il existe de constructions syntaxiques exprimant une relation entre entités nommées. Celles-ci sont potentiellement nombreuses.

Remerciements

Je remercie mon directeur de recherche, Thierry Poibeau, pour son suivi et ses conseils avisés. Je remercie également Nicolas Dessaigne, directeur technique de l'entreprise qui m'emploie, ainsi que mon collègue Gaël Patin pour son aide apportée. Ces recherches ont été en partie effectuées dans le cadre du projet CAHORS financé par l'ANR (appel à projet CSOSG 2008).

Références

ACE (2004). *Automatic Content Extraction, English Annotation Guidelines for Relations*. ACE Consortium, Linguistic Data.

AHO A. & ULLMAN J. (1972). *The Theory of Parsing, Translation and Compiling*. Prentice Hall.

CALIFF M.-E. & MOONEY R. J. (2003). Relational learning of pattern matching rules for information extraction. In *The Journal of Machine Learning Research*.

FARGUES J., DUGOURD M.-C. L. A. & CATACH L. (1986). Conceptual graphs for semantics and knowledge processing. *IBM Journal of Research and Development archive*, **30**(1), 70–79.

GOUJON B. (2008). Relation extraction in an intelligence context. In *LangTech 2008*.

HEARST M. (1992). Automatic acquisition of hyponyms from large text corpora. *Conference On Computational Linguistics (COLING)*, p. 539–545.

HUFFMAN S. B. (1996). Learning information extraction patterns from examples. In *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*.

MUC (1998). *Proceedings of the Seventh Message Understanding Conference*. MUC-7.

RILOFF E. (1996). Automatically generating extraction patterns from untagged text. In *Thirteenth National Conference on Artificial Intelligence (AAAI-96)*.

SILBERZTEIN M. (1993). *Dictionnaires électroniques et analyse automatique de textes - Le système Intex*. Masson.

SODERLAND S., FISHER D., ASELTINE J. & LEHNERT W. (1995). Crystal : Inducing a conceptual dictionary. In *Fourteenth International Joint Conference on Artificial Intelligence*.

SORICUT R. & MARCU D. (2003). Sentence level discourse parsing using syntactic and lexical information. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

SOWA J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley.

ZELENKO D., AONE C. & RICHARDELLA A. (2003). Kernel method for relation extraction. In *Journal of Machine Learning Research*.

ZHANG S., ZHANG S. & GAO G. (2008). Automatic entity relation extraction based on conditional random fields. *Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08*.

ZHAO S. & GRISHMAN R. (2005). Extracting relations with integrated information using kernel methods. In *ACL 2005*, Dourdan.