Grammaire de dépendances et ontologies de haut niveau : vers un processus modulaire pour l'analyse sémantique

Amal Zouaq, Michel Gagnon, Benoit Ozell

Ecole Polytechnique de Montréal-Montréal-Qc-Canada

Résumé

L'analyse sémantique et l'étiquetage de rôles sémantiques sont des problématiques importantes pour les applications du traitement de la langue naturelle et du Web sémantique. Cette analyse s'appuie généralement sur des ressources (lexiques et rôles sémantiques) élaborées manuellement et qui sont bien souvent dédiées à un domaine particulier. Cela est peu compatible avec les besoins actuels de mise à échelle, de fiabilité et de mise à jour des connaissances extraites ainsi que d'adaptation des systèmes à divers domaines de connaissance. Par ailleurs, il existe divers typologies de rôles sémantiques sous des formats propriétaires et peu standardisés, ce qui pose des problèmes d'interopérabilité. Ce travail décrit une architecture modulaire pour l'extraction de connaissances qui répond à ces problématiques et qui comprend un analyseur syntaxique, un analyseur logique et un annotateur de rôles. L'analyseur logique utilise une grammaire basée sur les dépendances et ne nécessite pas de lexique sémantique intégré. Le système utilise également une ontologie de haut niveau, SUMO, qui permet d'associer des rôles formels et inter-échangeables aux éléments extraits. Une évaluation du système est également effectuée et démontre l'obtention de résultats intéressants par comparaison avec des modèles existants.

Abstract

Semantic analysis and semantic role labeling are crucial issues for natural language processing and Semantic Web applications. This analysis often relies on manually-built semantic resources (lexicons, grammars, etc.) that are generally devoted to a particular domain and that are costly to acquire. This is incompatible with current challenges of the field, which involve the portability and interoperability of semantic analyzers and their adaptability to various application domains. Moreover, semantic role labeling often relies on proprietary and non standard terminologies, which again hinders their reusability. This paper proposes a modular architecture pipeline for semantic analysis which describes a solution to these issues and which decomposes the process into a syntactic analysis, a logical analysis and semantic role labeling. The logical analysis is based on dependency grammars and does not require a lexicon. The system also involves the use of an upper-level ontology that describes formally the semantic roles. An evaluation of the system is presented which shows interesting results by comparison with existing systems and models.

Keywords : semantic analysis, dependency grammar, logical analyzer, upper-level ontology, SUMO

1. Introduction

L'analyse sémantique est une application importante des techniques du traitement automatique de la langue (TAL). Elle permet de donner un sens aux phrases et aux textes analysés. Traditionnellement, ce sens est associé aux textes au travers de l'annotation des phrases par des rôles sémantiques, ce qui nécessite des algorithmes de désambiguïsation. Une analyse syntaxique préalable est alors souvent utilisée comme étape initiale à l'annotation. Cette analyse constitue donc une étape cruciale dont dépend la performance des traitements réalisés

aux étapes subséquentes. Deux problématiques sont actuellement posées par la communauté de recherche. La première pose la question du formalisme à utiliser. Bien que les grammaires de constituants aient été traditionnellement utilisées et considérées comme plus performantes dans certaines études telles que Pradhan et al. (2005) et Swanson et Gordon (2006), d'autres travaux récents semblent contredire ces résultats à la lumière des récents développements concernant les grammaires de dépendances (Johansson et Nugues, 2008). La deuxième problématique est de nature architecturale. L'analyse sémantique actuelle repose essentiellement sur des caractéristiques lexicales et sur le mot comme pierre angulaire à l'analyse. C'est le cas par exemple des grammaires catégorielles (Steedman, 1998) ou HSPG (Sag et al., 2003) qui reposent sur des lexiques sémantiques pour mener à bien une analyse compositionnelle. Or ces lexiques sont eux-mêmes assez lourds à développer. La question qui se pose est donc si les grammaires de dépendances peuvent permettre de s'affranchir d'un tel lexique intégré et si une architecture modulaire d'analyse sémantique peut être développée, basée sur ces grammaires. Cette notion de modularité est importante car elle permet de ne pas se lier à des ressources particulières telles qu'un lexique, une typologie de rôles ou un analyseur syntaxique donnés. En effet, l'un des problèmes actuellement rencontrés par les systèmes d'annotation de rôles repose sur la typologie des rôles à choisir (Marquez et al., 2008). Des plus générales aux plus détaillées, ces typologies enferment le système dans une terminologie donnée et il est alors difficile de rendre de tels systèmes interopérables. Or, à l'ère du Web sémantique, une telle approche n'est plus possible. Les rôles doivent pouvoir être définis dans des structures formelles telles que les ontologies et de ce fait être interopérables et partageables.

Cet article se propose de décrire une démarche permettant de :

1. Définir un processus d'analyse sémantique modulaire reposant sur une analyse syntaxique, une analyse logique et enfin une annotation de rôles ;
2. Utiliser les grammaires de dépendances pour l'analyse logique, en se basant uniquement sur des patrons syntaxiques ;
3. Intégrer une ontologie pour la définition des rôles sémantiques ;
4. Intégrer un lexique sémantique seulement à l'étape de l'annotation des rôles.

L'article est organisé comme suit. La section 2 décrit l'état de l'art dans l'analyse sémantique et l'annotation de rôles. Dans la section 3, nous présentons notre prototype et décrivons une architecture modulaire comprenant une analyse syntaxique, une analyse logique et enfin une annotation de rôles sémantiques. La section 4 décrit l'ensemble des expérimentations que nous avons menées ainsi que les corpus utilisés. Pour finir, nous effectuons une analyse des résultats et présentons les travaux futurs envisagés.

2. État de l'art

La nature des rôles sémantiques est une question soulevée depuis les travaux de Fillmore (1968). Elle est généralement envisagée de deux manières (Marquez, 2009) : soit en utilisant une approche basée sur la syntaxe, telle l'approche de Levin (1993) et de VerbNet (Kipper et al., 2000), qui se focalisent sur les verbes, soit en utilisant une approche basée sur les situations ou cadres (frame) tel que proposées dans le projet FrameNet (Fillmore et al., 2004). Cette dernière nécessite un savoir encyclopédique et n'est pas adaptée à tous les contextes et domaines. De manière générale, ces deux approches nécessitent une description de patrons dits syntaxiques-sémantiques, que ce soient des verbes ou des cadres plus génériques. C'est également le cas des grammaires catégorielles (Steedman, 1998) ou des grammaires HPSG (Sag et al., 2003), qui nécessitent des lexiques sémantiques intégrés pour associer un sens aux différents arguments

et prédicats d'une phrase de manière compositionnelle. Les analyses syntaxique et sémantique sont alors envisagées comme un tout intégré. Or les conclusions de Marquez (2009) dans la conférence (*CoNLL-2009 shared task*) montrent que les meilleures performances sont toujours obtenues par des systèmes aux architectures modulaires comprenant une analyse syntaxique puis sémantique. Par ailleurs, à un niveau d'ingénierie logicielle, il est plus facile de mettre à jour et de maintenir des systèmes modulaires.

De plus, les performances de l'état de l'art, notamment en analyse par dépendances, montrent qu'un certain niveau de maturité est atteint par les analyseurs syntaxiques (Stevenson et Greenwood, 2009 ; Johansson et Nugues, 2008) et qu'il est alors possible de reposer sur de tels analyseurs pour une analyse sémantique. En effet, de plus en plus d'efforts sont effectués pour comparer les performances des systèmes d'analyse sémantique basés sur des formalismes différents. C'est le cas par exemple de Johansson et Nugues (2008), qui démontrent que les analyseurs basés sur les dépendances n'ont plus rien à envier aux systèmes basés sur les constituants, ou des expérimentations de Greenwood et Stevenson (2009) qui désignent l'analyseur de l'université de Stanford (Klein et Manning, 2003) comme étant parmi les plus performants.

Enfin, il est à noter que les conclusions de la conférence CoNLL-2009 (Marquez, 2009) démontrent que des problèmes de portabilité des analyseurs sémantiques sont toujours un des points faibles de la discipline. En effet, les analyseurs les plus performants reposent généralement sur des approches statistiques et d'apprentissage machine qui nécessitent des corpus d'entraînement. Or il suffit d'exécuter ces systèmes sur des domaines différents pour entraîner une dégradation notable des performances. Pour améliorer cette portabilité, il est primordial de viser une approche indépendante du domaine. De plus, Marquez et al. (2008) et Marquez (2009) font également état du choix des typologies de rôles comme obstacle à cette portabilité.

3. Une architecture modulaire pour l'analyse sémantique

Notre objectif est donc de proposer une architecture modulaire qui décompose le processus d'analyse linguistique en trois étapes : une analyse syntaxique, une analyse logique et une annotation sémantique. L'analyse syntaxique s'appuie sur une grammaire de dépendances (De Marneffe et al., 2006) issue de l'analyseur de Stanford. L'approche n'est pas statistique, du moins au niveau de l'analyse logique. Elle ouvre le champ à l'essai de plusieurs algorithmes de désambiguïsation lors de l'annotation par rôles sémantiques. Elle définit également formellement les rôles au moyen de l'ontologie de haut niveau SUMO (Pease et al., 2002), ce qui permet une interopérabilité au niveau des systèmes adoptant cette ontologie ou tout simplement des systèmes aptes à la comprendre, et offre une portabilité accrue avec la possibilité de changer cette ontologie. Étant de haut niveau, cette ontologie n'est pas dédiée à un domaine particulier, mais peut toutefois être étendue, si le besoin s'en fait sentir, par des concepts du domaine. La Fig. 1 décrit cette architecture. Les textes traités actuellement sont en anglais.

3.1. Analyse syntaxique

Le processus d'analyse sémantique commence donc par une analyse syntaxique par dépendances De Marneffe et al. (2006) ont développé une telle analyse en se greffant sur l'analyseur de Stanford (produisant initialement des analyses par constituants) et en définissant une hiérarchie de relations grammaticales. L'analyse d'une phrase produit donc une relation grammaticale entre chaque paire de mots dans la phrase. Les dépendances ont été largement utilisées dans la communauté de forage de textes, et suscitent depuis peu un regain d'intérêt dans la communauté de traitement linguistique. Cet article se situe dans la droite ligne de ces travaux.

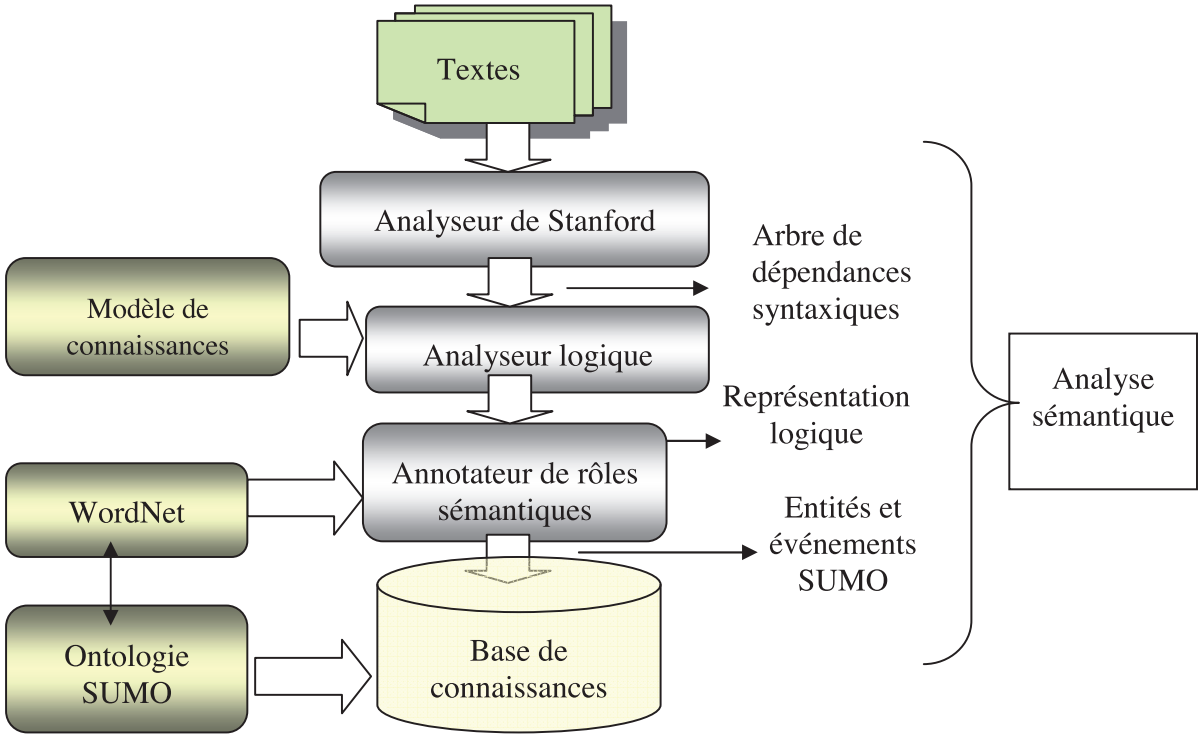

Figure 1 : Architecture modulaire pour l'analyse sémantique

Un exemple de sortie de l'analyseur de Stanford pour une phrase telle que « *Banners flap in the wind outside the walls of the city* » est présenté ci-dessous.

nsubj(flap-2, Banners-1);prep(flap-2, in-3);det(wind-5, the-4);pobj(in-3, wind-5);prep(flap-2, outside-6);det(walls-8, the-7);pobj(outside-6, walls-8);prep(walls-8, of-9);det(city-11, the-10);pobj(of-9, city-11).

L'analyseur permet également l'étiquetage des mots de la phrases selon leurs catégories grammaticales (*Parts-of-speech*) telles que dans l'exemple : *Banners/NNS flap/VBD in/IN the/DT wind/NN outside/IN the/DT walls/NNS of/IN the/DT city/NN./.*

Basés sur ces deux résultats, un transformateur produit un arbre de dépendances syntaxiques incluant les relations de dépendances et les catégories syntaxiques des mots. L'exemple suivant montre un tel arbre de dépendances.

```
root/tree(token(flap,2)/v,
 [nsubj/tree(token(banners,1)/n,[]),
  prep/tree(token(in,3)/prep,
   [pobj/tree(token(wind,5)/n,
    [det/tree(token(the,4)/d,[])])]),
  prep/tree(token(outside,6)/prep,
   [pobj/tree(token(walls,8)/n,
    [det/tree(token(the,7)/d,[]),
     prep/tree(token(of,9)/prep,
      [pobj/tree(token(city,11)/n,
       [det/tree(token(the,10)/d, [])])])])])]).
```

3.2. Analyse logique

L'arbre de dépendances syntaxiques est ensuite transmis à notre analyseur logique, implémenté en Prolog et inspiré de l'approche de Zouaq (2008). Cet analyseur produit des représentations en logique de premier ordre et s'appuie sur un modèle de connaissances minimal.

3.2.1. Le modèle de connaissances, une typologie universelle des rôles sémantiques

Le modèle de connaissances que nous introduisons à ce niveau définit des catégories génériques et universelles qui se retrouvent dans toutes les typologies de rôles sémantiques. Il comprend les catégories suivantes : ***Entity, Named Entity, Super-type, Event, Statement, Circumstance, Time, Number, Measure*** et ***Attribute.*** Chaque catégorie est déterminée par une étiquette grammaticale et par des relations de dépendance détectées dans l'arbre syntaxique, ainsi que le montre le Tab. 1.

Introduire un tel modèle n'est pas une nécessité. Il serait en effet possible de se contenter d'identifier les unités lexicales par des identificateurs. Toutefois, nous pensons qu'un tel modèle représente une étape intermédiaire utile à l'analyse sémantique subséquente.

Catégorie du modèle	*Catégorie Syntaxique*	*Exemple*
Entity	Nom (n)	The **cat** eats.
Event	Verbe (v)	The cat **eats**.
Statement	Tout patron comprenant une relation xcomp (clausal complement with external subject)	I **like** {to eat in the garden}$_{xcomp}$
Circumstance	Toute relation advcl (Adverbial clause) advcl	The accident happened {as the night was **falling**}$_{advcl}$
Time	Toute relation tmod (Temporal modifier)	He swam in the pool {last **night**}$_{tmod}$
Number	Toute relation num (Numeric Modifier)	**200** people came to the party
Attributes	1. Sujet nominal et copule 2. Toute relation acomp (Adjectival complement) 3. Toute relation amod (Adjectival modifier)	1. The cat is **big** 2. He looks **tired** 3. He is a **happy** man
Measure	Toute relation Measure	The director is 55 **years** old

Tableau 1 : Les catégories du modèle et les catégories syntaxiques applicables

3.2.2. Une analyse compositionnelle

L'analyseur effectue une analyse compositionnelle. À la différence des analyseurs compositionnels basés sur le lambda-calcul, qui reposent sur les unités lexicales pour leur analyse, notre analyseur se charge de détecter des patrons syntaxiques dans l'analyse par dépendances. Chaque composant d'un patron est ensuite logiquement analysé par un sous-patron et ainsi de suite jusqu'à l'unité lexicale. L'analyse à ce niveau se contente de générer un identificateur lorsque cela est requis, d'assigner une des catégories du modèle de connaissances à l'unité lexicale et enfin de déterminer si cette unité a déjà été rencontrée dans le discours.

Aucun trait sémantique n'est alors requis. Les résultats des sous-analyses sont ensuite combinés de manière à obtenir une représentation générale pour la phrase. La Fig. 2 montre ainsi l'analyse de la phrase simple : «*Banners flap in the wind*» constituée d'un ensemble de sous-analyses, représentées ici par des cercles. La figure montre également le résultat de ces sous-analyses.

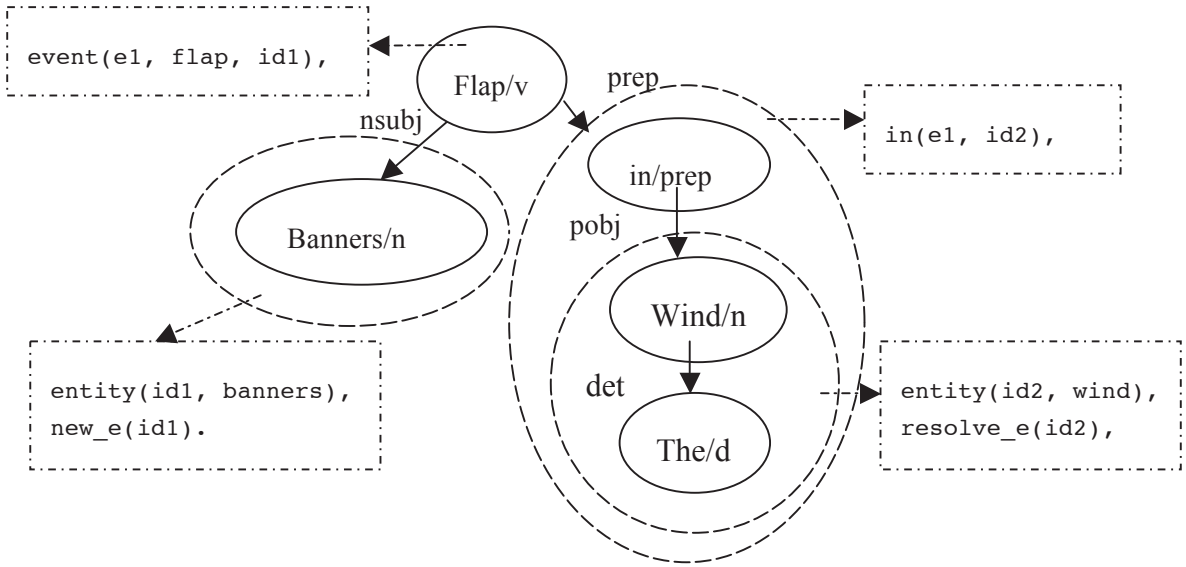

Figure 2 : Un exemple des différentes sous-analyses compositionnelles

Le code suivant illustre notre analyse compositionnelle pour le patron «*sujet nominal – verbe – complément d'objet direct*»

```
semparseMainPattern(tree(Node/v,Children),
                    tree(Node/v,Rest),
                    Id,
                    SemIn,
                    [event(Id,Node,IdAgent,IdObject)|SemOut]):-
    select(nsubj/tree(N1/_,C1),Children, R1),
    select(dobj/tree(N2/_,C2),R1, Rest),
    semparse(tree(N1/n,C1),_,IdAgent,SemIn,Sem1),
    semparse(tree(N2/n,C2),_,IdObject,Sem1,SemOut),
    gensym(e,Id).
```

Chaque sous-arbre *(nsubj* et *dobj*) est analysé séparément, le résultat de ces analyses est combiné et une catégorie *événement* est enfin créée.

3.2.3. Des patrons basés sur une grammaire de dépendances

Nous avons défini une grammaire constituée d'un ensemble de patrons couplés à des règles de transformation de ces structures syntaxiques en représentations en logique de premier ordre et utilisant le modèle de connaissances défini ci-dessus. Cette grammaire est divisée en ***patrons principaux*** (*core patterns)* décrivant les structures grammaticales principales de la langue (tels que le patron défini ci-dessus) et en ***patrons modificateurs***, qui apportent des précisions ou spécifient certaines parties de la phrase (modificateur temporel, causal, prépositionnel, etc.). Les patrons sont organisés en une hiérarchie où les patrons contenant le plus de relations syntaxiques figurent en premier et où leurs fils comprennent obligatoirement au moins une

relation grammaticale en commun avec le parent immédiat. Dans notre analyseur Prolog, cette hiérarchie est simplement représentée par un ensemble de règles où les parents sont placés avant leurs enfants. Plusieurs patrons peuvent être instanciés dans une même phrase incluant un ou plusieurs patrons principaux et zéro ou plusieurs modificateurs. Notons également que les représentations logiques peuvent indiquer des relations logiques non évidentes dans le texte. Ainsi par exemple, une relation de possessif comme dans «*the author's name*» est transformée en un prédicat *has_attr (author, name)*. Toutefois, ce prédicat ne contraint pas encore le sens d'une telle relation qui peut revêtir différentes significations (possession, partie-de, etc.). C'est dans l'étape de l'annotation sémantique que cette désambiguïsation est effectuée.

3.3. Annotation sémantique

L'annotation sémantique se base donc sur une formule logique pour effectuer la désambiguïsation et l'annotation selon des rôles sémantiques. Elle repose pour cela sur les catégories du modèle de connaissances pour orienter la désambiguïsation. Par exemple, une entité devra normalement être indexée par un concept SUMO.

3.3.1. Ressources

Pour rendre la typologie des rôles sémantiques réutilisable et partageable, notre objectif était de les définir sous forme d'ontologie. Pour ce faire, nous avons choisi l'ontologie SUMO (Pease et al., 2002), qui contient un grand nombre de concepts et de relations de haut niveau et qui permet le rattachement d'ontologies du domaine à ces catégories. Annoter des textes selon ces catégories de haut niveau était adéquat dans le cadre du corpus dont nous disposions, utilisant un vocabulaire usuel non technique et constitué d'histoires pour enfants. Toutefois, dans le contexte d'une plate-forme à base d'ontologies, rien n'interdit de greffer une ontologie du domaine pour décrire plus spécifiquement certains concepts. C'est d'ailleurs ce qui a déjà été fait puisque de nombreuses ontologies du domaine étendent actuellement SUMO. Par ailleurs, SUMO se distingue par un long cycle d'expérimentation qui assure sa fiabilité. Enfin, l'un des points forts de SUMO est que ses concepts et ses relations ont été appariés avec le lexique WordNet (Fellbaum, 1998) par Niles et Pease (2003), construisant ainsi une interface entre le vocabulaire utilisé dans les textes, les différents sens qu'il peut revêtir et les catégories conceptuelles de SUMO. Cet appariement est loin d'être parfait, car certains liens sont inconsistants. Il est ainsi possible de trouver une relation verbale indexée par un concept SUMO (ce qui peut porter à confusion) plutôt que par une relation SUMO (ce qui est logique). Nonobstant ces imperfections, cet appariement a au moins le mérite de montrer les liens qui peuvent être établis entre lexique et ontologies et notre article montre comment cet appariement peut être exploité. Enfin, à des fins d'évaluation, nous avons utilisé la ressource WordNet, qui est largement exploitée dans des compétitions telles que Senseval (Mihalcea et al., 2004), permettant de juger la performances d'algorithmes de désambiguïsation et d'étiquetage de rôles.

3.3.2. Algorithmes de désambiguïsation et contextes

L'intérêt de notre architecture modulaire se voit également à cette étape d'annotation et de désambiguïsation, où les rôles sémantiques sont pris en compte. En effet, l'étiquetage se base généralement sur deux étapes essentielles : l'identification des prédicats et des arguments dans le texte et ensuite leur catégorisation selon une certaine typologie de rôles. Dans notre cas, la première étape est déjà effectuée puisque la représentation logique identifie les entités, événements et autres relations de la phrase. À ce niveau, l'annotation se charge donc uniquement de la catégorisation de ces éléments.

Pour effectuer le passage des textes aux sens de WordNet puis enfin aux concepts SUMO, nous avons implémenté divers algorithmes de désambigüisation utilisant des ressources linguistiques telles que WordNet. Ces algorithmes incluent l'algorithme simplifié de Lesk (Kilgarrif et Rosenzweig, 2000), l'algorithme de Banerjee et Pedersen (2003) qui étend la définition d'un sens par son réseau sémantique direct extrait de WordNet (incluant les descriptions, mots reliés, hyponymes, hypéronymes, etc.). Nous avons également implémenté l'algorithme du sens le plus fréquent qui, à partir d'un corpus d'entraînement, définit le sens le plus fréquemment utilisé pour un mot donné. Dans WordNet, et en l'absence de données d'entraînement, les sens sont ordonnés selon une fréquence extraite du concordancier SEMCOR. Enfin, nous nous sommes également intéressés à des algorithmes minimalement supervisés et reposant sur des données d'entraînement indiquant, pour chaque sens de WordNet, un ensemble de termes cooccurrents ainsi que leurs fréquences de cooccurrences. Dans ce cadre, un algorithme reposant sur les fréquences de cooccurrences des termes avec un sens donné dans le corpus d'entraînement a été implémenté, de même qu'une combinaison de ces cooccurrences avec l'algorithme de Banerjee et Pedersen. Dans chacun de ces cinq algorithmes, nous avons utilisé la même mesure de similarité, soit le nombre de termes partagés par un sens donné et par le contexte du mot à désambiguïser. Il est à noter que l'objectif de cet article n'est pas de proposer un nouvel algorithme de désambiguïsation mais de montrer comment les algorithmes existants peuvent être utilisés pour annoter notre représentation logique. Si l'on considère une telle représentation annotée par WordNet, on obtient alors la formule suivante pour notre exemple type :

outside(e1, id3), of(id3, id4), entity(id4, WN: city%1:15:00::), resolve_e(id4), entity(id3, WN: wall%1:06:01::), resolve_e(id3), in(e1, id2), entity(id2, WN: wind%1:04:01::), resolve_e(id2), event(e1, WN: flap%2:38:00::, id1), entity(id1, WN: banner%1:06:00::), new_e(id1).

En utilisant SUMO, cette même expression devient :

outside(e1, id3), of(id3, id4), entity(id4, SUMO:City), resolve_e(id4), entity(id3, SUMO: StationaryArtifact), resolve_e(id3), in(e1, id2), entity(id2, SUMO: Wind), resolve_e(id2), event(e1, SUMO: Motion, id1), entity(id1, SUMO: Fabric), new_e(id1).

Il est à souligner que seuls les entités et événements sont annotés actuellement mais que nos travaux subséquents devront permettre d'annoter également des relations telles que «*outside*» ou «*of*».

Par ailleurs, les algorithmes de désambiguïsation nécessitent généralement la définition d'un contexte, aussi bien pour le mot à désambiguïser que pour ses différents sens possibles. La démarche de désambiguïsation repose ensuite sur une mesure de similarité qui permet de mesurer la distance entre le contexte du mot à désambiguïser et le contexte de chacun de ses sens. Le sens le plus proche est celui qui est finalement choisi. Dans notre cas, nous avons testé un ensemble de contextes comprenant des fenêtres de mots, des fenêtres de phrases et enfin des fenêtres issues de notre analyse par dépendance (contexte syntaxique) et de notre analyse logique (contexte logico-sémantique). Par exemple, dans notre exemple type, un contexte syntaxique pour le terme *flap* est composé de [*banners, in, outside*] tandis que son contexte logico-sémantique est constitué de *[banners, wind, walls]*. Enfin, ces deux derniers contextes peuvent être globaux ou locaux. Un contexte local ne contient que les nœuds directement reliés au terme à désambiguïser tant qu'un contexte global se construit de manière incrémentale en incluant toutes les contextes locaux précédents ainsi que le sens finalement choisi pour le terme. Cela accroît la taille des vecteurs qui représentent le texte et augmente donc les chances de retrouver des termes partagés avec les contextes des sens. Toutefois, un contexte global n'a de sens que dans le cadre de la désambiguïsation d'un texte dans son ensemble.

4. Evaluation

L'évaluation d'une telle architecture modulaire doit permettre de mesurer la performance de chacun de ses composants. Dans cet article, nous nous concentrons essentiellement sur l'évaluation de l'analyseur logique et de la démarche de désambiguïsation.

4.1. Première expérimentation

Notre première expérimentation porte sur un petit corpus de 185 phrases tirées d'histoires pour enfants. Cette expérimentation vise à tester aussi bien l'analyseur logique que la désambiguïsation. Au niveau de l'analyse logique, l'objectif était de vérifier si les entités et les événements étaient correctement identifiés. Au niveau de l'annotateur de rôles, l'objectif était de vérifier dans quelle mesure l'assignation des sens SUMO était correcte. Ces deux évaluations ont été effectuées au moyen de métriques standard de précision et de rappel:

Précision *= items (entités et événements) corrects / nombre total d'items générés*
Rappel *= items (entités et événements) corrects / nombre total d'items que le système aurait dû générer*

Pour ce faire, nous avons manuellement annoté ce corpus par les étiquettes «entité» et «événement» ainsi que par leur sens SUMO. Le Tab. 2 donne les résultats obtenus lors de l'analyse logique. Ces résultats démontrent que notre analyseur logique détecte les principales structures syntaxiques correctement. Il doit toutefois être étendu de manière à couvrir plus de patrons ainsi que le démontre le taux de rappel.

	Précision %	*Rappel %*
Entités	95.09	80.16
Evénements	94.87	85.27

Tableau 2 : Taux de précision et de rappel pour les entités et événements lors de l'analyse logique

Il aurait été intéressant de tester notre analyseur logique avec les données de Senseval pour les formes logiques (Rus, 2004). Toutefois, notre formalisme logique n'était pas exactement identique à celui des données. Des travaux futurs devront permettre d'effectuer une telle démarche sur un corpus plus étendu.

Au niveau de l'annotation sémantique (Tab. 3), les résultats sont décomposés selon les trois textes du corpus. Nous avons constaté que pour chaque texte et chaque élément, des algorithmes différents offrent les meilleures performances. Ainsi, par exemple, les entités du texte A sont mieux annotées via l'algorithme de Banerjee et Pedersen couplé aux fréquences de cooccurrences tandis que le texte C est mieux annoté seulement avec les fréquences de cooccurrences. Il reste à déterminer quelles sont les caractéristiques qui diffèrent d'un texte à l'autre et qui donc mènent à ces différences.

	Entité SUMO		*Événement SUMO*	
%	*Précision*	*Rappel*	*Précision*	*Rappel*
Texte A	91.23	67.97	86.57	69.88
Texte B	91.62	87.93	84.61	84.61
Texte C	77.51	64.28	52.43	47.37
Moyenne	86.78	73.39	75.54	67.29

Tableau 3 : Meilleurs résultats obtenus lors de la l'annotation par rôles sémantiques

De manière générale, les résultats du Tab. 3 sont intéressants même si le rappel n'est pas satisfaisant. Ce rappel peut être influencé par une mauvaise analyse logique ne contenant pas toutes les entités et événements mais également par une mauvaise désambiguïsation. Plus spécifiquement, l'un des résultats intéressants de cette expérimentation concerne le contexte qui a mené à ces performances, à savoir les contextes dits syntaxiques et logico-sémantique par comparaison à des fenêtres de phrases et de mots (Tab. 4).

Contexte	*Entité*	*Evénement*
Texte A	Contexte logico-sémantique global	Contexte syntaxique global
Texte B	Contexte syntaxique global	Contexte logico-sémantique local
Texte C	Contexte syntaxique local	Contexte syntaxique global

Tableau 4 : Contextes menant aux meilleures performances

4.2. Deuxième expérimentation

Afin de réaliser une expérimentation à plus large échelle pour l'annotation sémantique, nous avons effectué un deuxième test avec les données de la compétition Senseval (Mihalcea et al., 2004) dans la tâche *« English lexical sample task »*. Le Tab. 5 retrace les meilleures performances obtenues avec les différents algorithmes de désambiguïsation.

Algorithme	*Résultat (fine-grained)*	*Résultat (coarse-grained)*
Sens le plus fréquent	55.2	61.2
Banerjee et Pedersen	55.1	61.1
Lesk simplifié	55.1	61.2
Fréquences de cooccurrences	63.7	68.9
Banerjee et Pedersen couplé aux fréquences de cooccurrences	**64.1**	**69.1**

Tableau 5 : Les meilleures performances obtenues avec divers algorithmes de désambiguïsation

Les performances de notre système de désambiguïsation nous placent juste derrière le meilleur système non supervisé (Ramakrishnan et al., 2004) de la compétition. Nous considérons toutefois notre approche comme minimalement supervisée puisqu'elle utilise des fréquences de cooccurrences extraites des données d'entraînement de Senseval (c'est également le cas de Ramakrishnan et al., 2004). Par ailleurs, nous n'avons pas été en mesure de confirmer l'intérêt des contextes syntaxiques et logico-sémantiques avec les données de Senseval, notamment à cause des multiples erreurs de ponctuation, d'analyse syntaxique et d'analyse logique. Cela pose la question des corpus à mettre en place pour la communauté de recherche utilisant des analyses linguistiques profondes ainsi que cela a été proposé dans cet article. Cette expérimentation ainsi que la mesure des performances de notre analyseur logique sur des corpus plus importants seront considérés dans des travaux futurs.

5. Conclusion

Cet article propose une architecture modulaire pour l'analyse sémantique qui se décompose en analyse syntaxique, analyse logique et annotation de rôles. L'annotateur logique utilise une grammaire de patrons syntaxiques basée sur les dépendances et extrait des représentations en logique de premier ordre. Cela permet d'une part une identification des prédicats et des

arguments d'une phrase, étape nécessaire à l'étiquetage sémantique. D'autre part, l'utilisation d'une ontologie de haut niveau garantit une interopérabilité entre les systèmes et permet d'effectuer des raisonnements sur ces structures. Elle offre une certaine indépendance vis-à-vis d'un domaine donné et une portabilité accrue. La modularité de l'approche permet également de changer les différents éléments du système (analyseur syntaxique, ontologie utilisée) moyennant des modifications raisonnables. Le choix d'une grammaire de dépendances est justifié par la maturité technologique de telles grammaires et basé sur leurs performances pour l'analyse sémantique, ainsi que démontré par de nombreux travaux récents. Cette architecture permet donc de répondre à différentes problématiques soulevées par la communauté en analyse linguistique. Dans nos travaux futurs, nous envisageons d'enrichir nos patrons syntaxiques, ainsi que de procéder à des évaluations plus approfondies.

Références

Banerjee, S. and Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. In Gottlob, G. and Walsh, T., editors, *Proc. of the 18th IJCAI,* pp. 805-810, Morgan Kaufmann.

De Marneffe, M-C, MacCartney, B. and Manning. C.D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J. and Tapias, D., editors, *Proc. of LREC*, pp. 449-454, ELRA.

Fellbaum, C. (1998). WordNet : An Electronic Lexical Database. MIT Press.

Fillmore, C. J., Ruppenhofer, J., & Baker, C. F. (2004). Framenet and representing the link between semantic and syntactic relations. In Huang, C. and Lenders, W., editors, Frontiers in linguistics, pp. 19–59. Taipei : Institute of Linguistics, Academia Sinica.

Fillmore, C. (1968). The case for case. In E. Bach and R. T. Harms (Eds.), *Universals in linguistic theory*, pp. 1–88. Holt, Rinehart & Winston.

Johansson, R. and Nugues, P. (2008). The effect of syntactic representation on semantic role labeling. In Scott, D. and Uszkoreit, H., editors, *Proc. of COLING,* pp. 393-400, ACL.

Kilgarriff, A. and Rosenzweig, R. (2000). Framework and Results for English SENSEVAL. *Computers and the Humanities* 34 :15-48, Springer.

Kipper, K., Trang Dang, H., Palmer, M. (2000). Class-Based Construction of a Verb Lexicon. In *Proc. of 17th National Conference on Artificial Intelligence*, pp. 691-696, AAAI Press.

Klein, D. and Manning, C.D. (2003). Accurate Unlexicalized Parsing, *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 423-430, ACL.

Levin, B. (1993). *English Verb Classes and Alternations : A Preliminary Investigation*, University of Chicago Press.

Marquez, L. (2009). Semantic Role Labeling : Past, Present and Future. Tutorial at ACL-IJCNLP 2009, retrieved from http://www.lsi.upc.edu/~lluism/tmp/SRL-tutorial-ACL-IJCNLP-2009.pdf

Màrquez, L., Carreras, X. Litkowski, K.C. and Stevenson, S. (2008). Semantic Role Labeling : An Introduction to the Special Issue, *Computational Linguistics*, 34(2) : 145-159.

Mihalcea, R., Chklovski, T. And Kilgarriff, A. (2004). The Senseval-3 English Lexical Sample Task, in Mihalcea, R. and Edmonds, P., editors, *Proc. of Senseval*, pp. 25-28, ACL.

Niles, I., and Pease, A. (2003). Linking Lexicons and Ontologies : Mapping WordNet to the Suggested Upper Merged Ontology, in Arabnia, H. R., editor, *Proc. of IKE*, pp 412-416, CSREA Press.

Pease, A., Niles, I., and Li, J. (2002). The Suggested Upper Merged Ontology : A Large Ontology for the Semantic Web and its Applications. In *Workshop on Ontologies and the Semantic Web*, AAAI.

Pradhan, S., Wayne W., Hacioglu, K., Martin, J. and Jurafsky, D. (2005). Semantic role labeling using different syntactic views. In *Proc. of ACL,* pp. 581 - 588, ACL.

Ramakrishnan, G., Prithviraj, B. and Bhattacharyya, P. (2004). A Gloss Centered Algorithm for Word Sense Disambiguation. In Mihalcea, R. and Edmonds, P., editors, *Proc. of Senseval*, pp. 217--221, ACL.

Rus, V. (2004). A First Evaluation of Logic Form Identification Systems, In Mihalcea, R. and Edmonds, P., editors, *Proc. of Senseval*, pp. 37-40, ACL.

Sag, I., Wasow, T. and Bender, E. (2003). Syntactic Theory : A Formal Introduction, CSLI.

Steedman, M. (1998). Categorial Grammar, in Rob Wilson and Frank Keil (eds.), *The MIT Encyclopedia of Cognitive Sciences*, pp. 101-104, MIT Press.

Reid, S. and Gordon, A.S. (2006). A Comparison of Alternative Parse Tree Paths for Labeling Semantic Roles. In *Proceedings of COLING/ACL*, pp. 811-818, ACL.

Stevenson, M. and Greenwood, M.A. (2009). Dependency Pattern Models for Information Extraction, *Research on Language & Computation*, 7(1) : 13-39, Springer.

Zouaq, A. (2008). Une approche d'ingénierie ontologique pour l'acquisition et l'exploitation des connaissances à partir de documents textuels, *Thèse de doctorat*, Université de Montréal.