Titre: Title:	Comparaison d'une approche à base de règles avec une approche utilisant de l'apprentissage machine pour l'analyse sémantique
Auteur: Author:	François-Xavier Desmarais
Date:	2012
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Desmarais, F.-X. (2012). Comparaison d'une approche à base de règles avec une approche utilisant de l'apprentissage machine pour l'analyse sémantique [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1049/

UNIVERSITÉ DE MONTRÉAL

COMPARAISON D'UNE APPROCHE À BASE DE RÈGLES AVEC UNE APPROCHE UTILISANT DE L'APPRENTISSAGE MACHINE POUR L'ANALYSE SÉMANTIQUE

FRANÇOIS-XAVIER DESMARAIS

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INFORMATIQUE)

OCTOBRE 2012

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

COMPARAISON D'UNE APPROCHE À BASE DE RÈGLES AVEC UNE APPROCHE UTILISANT DE L'APPRENTISSAGE MACHINE POUR L'ANALYSE SÉMANTIQUE

présenté par : DESMARAIS François-Xavier
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. ROBILLARD Pierre-N., Ph. D., président
M. GAGNON Michel, Ph. D., membre et directeur de recherche
Mme ZOUAQ Amal, Ph. D., membre et codirectrice de recherche
M. OZELL Benoît, Ph. D., membre et codirecteur de recherche
Mme DA SYLVA Lyne, Ph. D. membre

DÉDICACE

À tous ceux qui m'ont aidé,

Merci!

REMERCIEMENTS

J'aimerais remercier mon directeur Michel Gagnon et mes codirecteurs Amal Zouaq et Benoît Ozell pour leur support, à la fois technique, moral et financier, tout au long de ma maîtrise. Je voudrais aussi remercier mes collègues Guillaume Bouilly, Paul Gédéon, Jonathan Tardif et l'associé de recherche Éric Charton pour leurs nombreux conseils. Finalement, j'aimerais exprimer toute ma gratitude à ma famille, qui m'a soutenu tout au long de mon cheminement, et spécialement ma copine qui m'a beaucoup aidé lors de la rédaction de mon mémoire.

RÉSUMÉ

L'analyse sémantique est une importante partie du traitement des langues naturelles qui repose souvent sur des modèles statistiques et des approches d'apprentissage machine supervisé. Cependant, ces approches nécessitent des ressources qui sont souvent coûteuses à acquérir. Ce mémoire décrit nos expériences afin de comparer Anasem, un analyseur sémantique en Prolog, avec le meilleur système de la tâche partagée (« Shared Task ») de la « *Conference on Natural Language Learning* » (CoNLL) sur l'analyse sémantique. Le meilleur système de CoNLL et Anasem sont basés sur des analyses de dépendance, mais leur différence majeure se situe au niveau des techniques d'extraction des structures sémantiques (à base de règles, par opposition à l'apprentissage machine). Nos résultats montrent qu'une approche fondée sur des règles est une solution capable de rivaliser avec les systèmes d'apprentissage machine sous certaines conditions.

ABSTRACT

Semantic analysis is a very important part of natural language processing that often relies on statistical models and supervised machine learning approaches. However, these approaches require resources that are costly to acquire. This paper describes our experiments to compare Anasem, a Prolog rule-based semantic analyzer, with the best system of the Conference on Natural Language Learning (CoNLL) shared task on semantic analysis. Both CoNLL best system and Anasem are based on a dependency representation, but the major difference is how the two systems extract their semantic structures (rules versus machine learning). Our results show that a rule-based approach might still be a promising solution able to compete with a machine learning system under certain conditions.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

CoNLL	« *Conference on Natural Language Learning* »
DRT	« *Discourse Representation Theory* »
DRS	« *Discourse Representation Structure* »
FOL	« *First Order Logic* »
OWL	« *Web Ontology Language* »
PoS	« *Part of Speech*»
SRL	« *Semantic role labeling* » ou Étiquetage des Rôles Sémantiques
SIGNLL	« *Special Interest Group on Natural Language Learning* »
TAL	Traitement Automatique des Langues
WSJ	« *Wall Street Journal* »

LISTE DES ANNEXES

INTRODUCTION

L'analyse sémantique est une tâche difficile dans le domaine du traitement de la langue. Celle-ci consiste à identifier les structures sémantiques dans des phrases ou des textes. Automatiser une telle tâche peut s'avérer être un défi de taille. Il existe deux méthodes majeures pour affronter ce problème : une approche à base de règles et une approche statistique avec apprentissage machine.

La première méthode consiste à définir des règles (ou patrons) qui s'appliquent sur une pré-analyse. La pré-analyse inclut tous les traitements qui sont faits avant l'analyse sémantique (par exemple : analyse syntaxique, identification des catégories grammaticales, etc.). La reconnaissance de ces patrons nous permet d'extraire une représentation logique de la phrase. Cette représentation peut prendre plusieurs formes, par exemple des formules en logique du premier ordre ou une structure de représentation du discours (« *Discourse Representation Structure* ») (DRS) (Blackburn & Bos, 2005). Les méthodes à base de règles permettent de traiter plusieurs phénomènes linguistiques tels que la résolution des coréférences, le traitement de la négation et l'identification des dépendances de longue distance. Par contre, elles sont difficiles à construire et à maintenir puisqu'elles peuvent contenir un grand nombre de règles définies à la main.

Pour la seconde méthode (basée sur l'apprentissage machine), il est nécessaire de décomposer l'analyse sémantique en différentes sous-tâches indépendantes plus spécifiques, telles que l'attribution des rôles sémantiques, la résolution de coréférences et l'extraction d'entités nommées, chacune de ces sous-tâches étant traitée généralement de manière indépendante. L'approche par apprentissage machine affiche de bons résultats lorsqu'elle se concentre sur ces sous-tâches. Par contre, elle possède certaines limites, telles que le besoin d'un corpus d'entrainement (apprentissage supervisé) ainsi que le fait que les modèles peuvent difficilement être utilisés avec des textes qui ne sont pas du même domaine que ceux utilisés pour l'entraînement.

On peut donc se demander laquelle de ces deux approches est la moins coûteuse au niveau du temps et laquelle donne de meilleures performances. Au cours de ce mémoire, nous tenterons de répondre à la seconde partie de cette question. Nous explorerons la question de recherche

suivante : est-ce qu'un analyseur à base de règles peut atteindre les mêmes performances qu'un analyseur basé sur l'apprentissage machine?

Malgré la popularité grandissante de l'analyse sémantique, il existe peu de travaux sur la comparaison de ces méthodes. De plus, les chercheurs ne semblent pas s'entendre sur un format standard pour exprimer leurs résultats, ce qui complique la tâche de comparaison des approches existantes.

Pour nous permettre de trouver une réponse à notre question de recherche, nous posons l'hypothèse suivante:

> *Les résultats obtenus par un analyseur à base de règles sont comparables à ceux obtenus par un analyseur basé sur l'apprentissage machine.*

Afin de vérifier cette hypothèse, nous avons défini une série d'étapes à accomplir pour confirmer ou infirmer notre hypothèse initiale :

1. Trouver deux analyseurs sémantiques, un à base de règles et un qui utilise de l'apprentissage machine.
2. Vérifier que les deux analyseurs utilisent le même format d'entrée et modifier l'un des analyseurs si ce n'est pas le cas.
3. Trouver un jeu de données utilisable par les deux analyseurs.
4. Évaluer les deux analyseurs en utilisant le jeu de données prédéfinies et des métriques précises.
5. Étudier les différences entre les résultats des deux approches en comparant leurs résultats.

Ces étapes sont présentées en ordre chronologique. Lorsque nous voulons comparer des analyseurs, il est crucial qu'ils aient accès aux mêmes informations initiales. Les trois premières étapes consistent à préparer le terrain pour la comparaison ultérieure des résultats. La quatrième étape nous permet de vérifier notre hypothèse. Finalement, la dernière étape a pour but de nuancer la réponse obtenue à la quatrième étape, et de comprendre pourquoi nous obtenons de tels résultats.

Ce mémoire est divisé en cinq chapitres. Dans le premier chapitre, les concepts de base de l'analyse sémantique sont présentés, ainsi qu'une revue de littérature sur ce domaine. Le

deuxième chapitre présente l'analyseur sémantique à base de règles, Anasem, et ses particularités. Ensuite, dans le troisième chapitre est expliquée la méthodologie utilisée pour comparer les analyseurs. Le quatrième chapitre expose l'ensemble des résultats obtenus lors des comparaisons. Finalement, le dernier chapitre discute des aspects intéressants des résultats ainsi que les limites de notre approche.

CHAPITRE 1 ÉTAT DE L'ART: MÉTHODES D'ANALYSE SÉMANTIQUE ET D'EVALUATION

La sémantique est une branche de la linguistique qui se consacre à l'étude du sens dans la langue[1]. Elle consiste à étudier les relations entre les signifiés, i.e. les concepts, et les signifiants, i.e. la représentation réelle (les termes signifié et signifiant ont été introduits par Ferdinand de Saussure (Tognotti, 1997)). L'objectif principal de la sémantique est de permettre la compréhension des énoncés dans un langage donné. C'est pourquoi l'analyse sémantique est très importante. Dans ce chapitre, nous nous concentrons sur la sémantique informatique, qui est le traitement de la sémantique par des ordinateurs, en présentant d'abord l'analyse sémantique comme telle. Nous poursuivons en explorant les représentations les plus communes pour l'analyse sémantique. Ensuite, nous décrivons en détail la campagne d'évaluation de CoNLL 2008. Puis, nous examinons les deux méthodes majeures qui ont été brièvement mentionnées dans l'introduction, soit l'analyse à base de règles et l'analyse basée sur l'apprentissage machine. Nous terminons en présentant des méthodes actuelles d'évaluation des analyses sémantiques.

1.1 Analyse sémantique

En sémantique informatique, la définition d'analyse sémantique est plutôt large. En fait, celle-ci est composée de plusieurs sous-tâches telles que a) l'étiquetage des rôles sémantiques (SRL) (Johansson & Nugues, 2008b), dont le but est de donner des rôles sémantiques à certains mots dans une phrase, b) la résolution des coréférences (Soon, et al., 2001) soit relier les différentes mentions d'un concept dans un texte, qui peut prendre la forme d'une désignation propre (par exemple : Paris, la capitale de la France, la Ville lumière, etc.) ou d'une expression anaphorique, comme un pronom, c) l'extraction d'entités nommées (Baluja, et al., 2000), soit identifier les noms de personnes, de lieux, d'organisation, etc., ou encore d) la désambigüisation du sens des mots (Yarowsky, 1992), qui sert à définir quel sens du mot doit être utilisé dans une situation particulière.

[1] http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=8369344

Il y a deux approches majeures, selon l'état de l'art, pour s'attaquer à ces problèmes : l'approche statistique avec l'utilisation d'apprentissage machine et l'approche à base de règles. Il est intéressant de noter que l'analyse syntaxique, une tâche qui est souvent cruciale à l'analyse sémantique, semble actuellement être l'exclusivité des méthodes d'apprentissage machine (De Marneffe, et al., 2006). Par contre, cela n'empêche pas l'approche à base de règles de rester compétitive pour d'autres tâches. Un exemple de ceci est la « *Shared Task* » de CoNLL 2011[2], qui portait sur la résolution de coréférences, et qui a été remportée par un système à base de règles (Lee, et al., 2011). Un autre exemple est la « *Shared Task* » de STEP 2008 (Bos, 2008b) à laquelle sept systèmes ont participé, incluant Boxer (Bos, 2008a), qui utilise un système à base de règles. Le but de cette tâche était de comparer des représentations sémantiques.

Nous avons constaté qu'il n'existe pas, à notre connaissance, de comparaison formelle entre ces deux approches majeures. C'est l'une des raisons principales pour lesquelles nous avons décidé de faire cette expérience. Cependant, pour faire cette comparaison, nous avons besoin d'un standard de référence (« Gold Standard ») pour la tâche que nous voulons comparer. Puisqu'il n'en existe pas pour l'analyse sémantique dans son ensemble, nous nous sommes rabattus sur une de ses sous-tâches, soit l'étiquetage des rôles sémantiques ou SRL (dans ce mémoire, les mentions futures du terme analyse sémantique font référence à cette tâche en particulier). L'une des seules compétitions à laquelle nous avions accès au « Gold Standard » est la « *Shared Task* » de CoNLL 2008(Johansson & Nugues, 2008b). Dans cette compétition, la grande majorité des systèmes, incluant celui qui a obtenu le meilleur score soit LTH Parser (Johansson & Nugues, 2008a), utilise des méthodes d'apprentissage machine. Une autre raison majeure qui nous a conduit à utiliser la compétition CoNLL pour effectuer nos comparaisons, outre le fait que ce soit une compétition reconnue, est le fait que nous avions accès aux scripts d'exécution ainsi qu'aux modèles de LTH Parser (un modèle, en apprentissage machine, est la représentation de l'apprentissage une fois que le système a été entraîné), le meilleur système de la compétition. Cela nous a permis de faire nos propres expériences avec ce système, choisi dans le cadre de notre comparaison. Nous nous sommes particulièrement focalisés sur la section sémantique a

[2] http://conll.cemantix.org/2011/task-description.html

« *Shared Task* » qui consiste à identifier, dans les phrases, les prédicats ainsi que les relations qui les lient à leurs arguments.

En résumé, dans ce mémoire, nous comparons deux systèmes sur une sous-tâche de l'analyse sémantique, soit le SRL. Ces deux systèmes sont : LTH Parser (appelé ultérieurement LTH), un analyseur basé sur l'apprentissage machine, et Anasem (Zouaq, et al., 2010), un analyseur sémantique à base de règles et de patrons. Ce dernier est présenté en détail au Chapitre 2.

1.2 Représentations sémantiques

La représentation sémantique consiste à définir une méthode par laquelle seront exprimées les informations sémantiques extraites des phrases. La plupart des chercheurs dans le domaine ne s'entendent pas pour dire quelle est la meilleure méthode. En fait, la représentation choisie est souvent liée à ce qu'on veut faire une fois l'analyse sémantique complétée. Toutefois, on peut cerner deux approches majeures : une représentation utilisant les cadres sémantiques (« *frame semantics* ») et une représentation en logique du premier ordre (FOL de « *First Order Logic* »).

1.2.1 Cadre sémantique

Le concept de cadres sémantiques vient de Charles J. Fillmore(Fillmore, 1976), le principe de cette approche est de décrire les événements, les relations, ou les entités en les reliant avec les éléments qui s'y rattachent. L'idée derrière ce concept est que le sens d'un mot peut être identifié parles mots dans son voisinage. Plusieurs projets ont vu le jour dans le but d'explorer ce principe et de créer une banque de données de cadres sémantiques, par exemple : FrameNet (Baker, et al., 1998), VerbNet(Schuler, 2005), WordNet (Fellbaum, 2010), PropBank (Palmer, et al., 2005) et NomBank (Meyers, et al., 2004). Tous ces projets sont maintenant des ressources très importantes pour le traitement automatique de la langue.

On peut mieux comprendre les cadres sémantiques lorsqu'on prend un exemple. Dans celui-ci, tiré de FrameNet, on regarde le concept de « cooking ». Dans ce cas particulier, « cooking » peut posséder jusqu'à quatre éléments (aussi appelé « *frame elements* ») : la personne qui cuisine (« *cook* »), ce qui est cuisiné (« *food* »), l'outil ou l'objet dans lequel on cuisine

(« *container* ») et ce qui procure la chaleur nécessaire à la cuisson (« *heating_instrument* »). Concrètement, dans la phrase « *The boys grill their catches on an open fire* », il y a le mot « *grill* » qui est un concept qui se rapporte à « *cooking*», « *the boys* » sont les personnes qui cuisinent (« *cook* »), « *their catches* » est la nourriture (« *food* ») et « *an open fire* » est la source de chaleur (« *heating_instrument* »). Cet exemple d'analyse sémantique se concentre sur le verbe « *grill* » et est accompagné de trois de ses quatre éléments possibles (la phrase ne fait pas mention de « *container* »). Donc, lorsqu'on veut faire une analyse sémantique en utilisant les cadres sémantiques comme représentation, on tente d'identifier les éléments et de les lier aux concepts appropriés de la ressource choisie (FrameNet, PropBank, etc.).

La campagne d'évaluation de CoNLL est basée exclusivement sur des cadres sémantiques (avec NomBank et PropBank), ce qui implique que l'analyse sémantique de LTH repose également sur cette notion. Cette particularité a compliqué la comparaison avec Anasem, qui utilise une représentation similaire à la logique du premier ordre.

1.2.2 Logique du premier ordre

Outre les cadres sémantiques, une autre approche consiste à utiliser la logique du premier ordre pour représenter les relations sémantiques. Généralement, cette approche est plus appropriée lorsqu'on tente de représenter l'analyse sémantique dans sa totalité. Son principe est que, dans une phrase, on peut identifier des prédicats et des relations liées à ces prédicats. On peut par la suite écrire, sous forme logique, les différentes relations dans la phrase. Par exemple, « *Anna loves Pete* » pourrait être écrit sous la forme logique suivante :

$$Loves(Anna, Pete)$$

Il est aussi possible de représenter des phrases plus complexes en utilisant des connecteurs logiques et des quantificateurs. Par exemple, la phrase : « *Every woman wears a dress* » peut être décrite de la façon suivante :

$$\forall x \left(Woman(x) \rightarrow \exists y \big(Dress(y) \wedge Wear(x, y) \big) \right)$$

En 1981, Hans Kamp a développé ce qui s'appelle la « *Discourse Representation Theory* » (DRT) (Kamp, 1981). Cette méthode est directement inspirée de la logique du premier ordre et a permis d'aller plus loin que de simples expressions logiques. En effet, Kamp a aussi introduit

l'idée d'une structure de représentation du discours « *Discourse representation Structure* » (DRS), qui est une structure inspirée de la représentation mentale qu'on peut se faire des phrases. Ce concept de représentation mentale, qui fonctionne aussi très bien en sémantique informatique, est qu'au fur et à mesure que la phrase progresse, les nouveaux éléments viennent s'ajouter à la représentation de la phrase.

Une DRS est constituée de deux parties, les référents, qui sont les éléments de la phrase, et les conditions, qui représentent les informations sur ces référents. Prenons par exemple la phrase « *Anna loves Pete* ». On peut le représenter par la DRS suivante :

$$\{x, y\}\{anna(x), pete(y), loves(x, y)\}$$

Il existe aussi une notation en forme de boite:

x,y
anna(x) pete(y) loves(x,y)

Un des avantages de cette méthode est qu'il est relativement simple d'ajouter des informations. Par exemple si la phrase était plus longue, par exemple : « *Anna loves Pete and he married her* », il suffit d'ajouter la relation « *marry(y,x)*» :

x,y
anna(x) pete(y) loves(x,y) marry(y,x)

Un exemple relativement récent de l'utilisation de cette théorie est Boxer (Bos, 2008a) de Johan Bos. Cet outil utilise en entrée une grammaire catégorielle combinatoire(Steedman, 2001) et fournit en sortie une DRS (il est possible de le tester en ligne[3]). C'est cette méthode qui est utilisé par Anasem.

[3] http://svn.ask.it.usyd.edu.au/trac/candc/wiki/Demo

1.3 Méthodes d'apprentissage machine vs méthode à base de règles

Regardons d'abord l'approche d'apprentissage machine supervisé un peu plus en détail. Cette méthode consiste à observer un corpus (un corpus est un regroupement d'un grand nombre de textes) complètement annoté, c'est-à-dire qui contient déjà les informations recherchées, et à construire un modèle en se basant sur ces observations. Grâce à ce modèle, il sera ensuite possible d'extraire les informations recherchées à partir de textes non-annotés. Il existe un grand nombre de procédés, par exemple on peut utiliser les SVM (« *Support Vector Machine* »)(Cortes & Vapnik, 1995) ou encore des arbres de décisions(Quinlan, 1986). Ces derniers possèdent un grand nombre de paramètres que l'on peut faire varier afin d'obtenir des résultats optimaux pour une situation en particulier. L'utilisation d'approches basées sur l'apprentissage machine est de plus en plus populaire, ce qui est dû à l'augmentation de la puissance de calcul des ordinateurs qui permettent maintenant de travailler avec un grand nombre de données en des temps relativement courts.

Pour pouvoir faire de l'apprentissage supervisé, il y a certaines étapes et ressources préliminaires qui sont nécessaires. Pour commencer, il faut trouver, ou créer, un jeu de données avec lesquels l'apprentissage se fera. Il faut une quantité de données relativement grande pour obtenir des résultats intéressants. Une fois le corpus obtenu, il faut définir les caractéristiques (« *features* ») sur lesquelles nous voulons que le système s'entraîne. Ce genre de système s'appelle un classificateur puisqu'il a pour but de classer des informations dans différentes catégories. Il est important de ne pas choisir trop de caractéristiques, puisque cela pourrait causer du sur-apprentissage et donc affecter les résultats de manière négative. Le sur-apprentissage est un phénomène qui se produit lorsque le système connait trop bien les données d'entrainement et qu'il n'arrive plus à classer les informations qui ne sont pas dans le corpus. Voici des exemples simples de caractéristiques pour l'analyse sémantique : la catégorie grammaticale des mots, la relation syntaxique des trois mots précédents et des trois mots suivants, la distance entre un mot et un verbe. Tout comme avec le corpus, il faut trouver un équilibre entre trop peu de caractéristiques et trop de caractéristiques pour la situation actuelle. Finalement, il faut faire des ajustements sur les paramètres du procédé choisi pour maximiser la solution. Pour ce faire, on réserve une section du corpus qui n'est pas utilisé lors de l'apprentissage.

Un bon exemple de système qui utilise de l'apprentissage supervisé est LTH(Johansson & Nugues, 2008b). L'analyse sémantique, faite par LTH, utilise une approche en trois parties. Premièrement, LTH crée une liste de candidats pour les prédicats et les arguments en utilisant une série de cinq classificateurs qui ont chacun une tâche spécifique : identification des prédicats, désambigüisation des prédicats, identification des supports (SU voir tableau 1.2 section 1.4) (pour les prédicats qui sont des noms), identification des arguments et classification des arguments. Tous ces classificateurs sont de type régression logistique linéaire régularisée L2 (« *L2-regularized linear logistic regression* ») implémenté en utilisant « LIBLINEAR » (Lin, et al., 2008)(Johansson & Nugues, 2008b). Ils sont entraînés sur les prédicats qui sont des verbes et sur les prédicats qui sont des noms, séparément. Après avoir obtenu la liste préliminaire des candidats, LTH utilise des contraintes logiques pour les filtrer (par exemple, retirer les duplicata, s'assurer que les arguments qui sont référencés existent bien). La dernière étape de son étiquetage des rôles sémantiques consiste à utiliser un classificateur global. Tous les classificateurs ont été entraînés, non pas directement sur le « Gold Standard », mais sur une version que LTH a annotée automatiquement en utilisant une validation croisée. Cette méthode permet de diminuer les impacts des erreurs d'analyse sur l'analyse sémantique puisque les classificateurs sont entraînés sur un corpus qui peut contenir des erreurs (la version annotée par LTH n'est pas parfaite).

Il existe aussi une autre approche appelée apprentissage non supervisé. Cette méthode est similaire à l'apprentissage supervisé à la différence que les corpus utilisés ne sont pas annotés. Toutefois, il n'existe pas d'analyseur sémantique entièrement non supervisé. Par contre, il y en a des semi-supervisés, c'est-à-dire avec un minimum de supervision. Un exemple de cette approche est le travail de Robert S. Swier et Suzanne Stevenson(Swier & Stevenson, 2004). Ils proposent une approche avec « bootstrapping » (d'où le semi-supervisé) qui assigne quelques rôles sémantiques aux mots. Puis, ils créent un modèle probabiliste, de manière itérative, pour assigner le reste des rôles sémantiques. À chaque itération, le modèle est de plus en plus grand et la probabilité limite pour assigner un rôle est diminuée jusqu'à ce que tous les rôles soient attribués. Naturellement, cette méthode n'obtient pas les meilleurs résultats, mais elle possède l'avantage de ne pas nécessiter de corpus annoté, qui n'est pas toujours disponible. Il existe également des méthodes d'analyse sémantique latente qui ne sont pas supervisées (Hofmann, 2001), mais là encore, on ne peut les considérer comme des analyseurs sémantiques à part entière puisque l'analyse sémantique latente n'est qu'une des facettes de l'analyse sémantique.

Ultimement, le but des méthodes basées sur l'apprentissage machine est de trouver les meilleures règles de classification possible en se basant sur des données pré-évaluées et d'appliquer ces règles sur de nouvelles données.

L'autre méthode, l'approche à base de règles, consiste essentiellement à créer les règles manuellement(Zouaq, et al., 2010)(Bos, 2008a). Au lieu de trouver ces règles à l'aide de calculs complexes et de corpus, on utilise les connaissances linguistiques et la logique pour les établir. Cette méthode a donc l'avantage de se faire sans l'utilisation d'un corpus. Dans de nouveaux domaines, ou dans des domaines où les données sont rares, cette méthode peut se révéler plus pratique que l'apprentissage machine, d'autant plus qu'elle peut couvrir la majorité de l'analyse sémantique. De plus, lorsque le domaine est bien circonscrit, il est souvent plus facile d'établir les règles manuellement. Cela dit, nous pensons qu'un système à base de règles, utilisé dans des situations complexes, peut obtenir des résultats comparables à ceux obtenus avec une approche basée sur l'apprentissage machine, d'où l'hypothèse de ce mémoire. Une autre caractéristique de cette méthode est qu'il est plus facile de faire des règles génériques qui couvrent un grand nombre de situations. Ceci permet une meilleure couverture en général, mais peut aussi donner des résultats moins performants dans des situations plus spécifiques.

À première vue, on pourrait penser que le traitement de la langue est une discipline qui se prête bien à l'utilisation de méthodes à base de règles. Toutefois, une analyse plus approfondie nous permet de constater que les règles qui régissent une langue (ici l'anglais) sont très complexes et qu'elles possèdent un grand nombre d'exceptions. C'est pourquoi il est difficile de définir manuellement l'ensemble des règles qui couvrent une langue, d'où la popularité des méthodes d'apprentissage automatique. Toutefois, avec Anasem, nous avons pu remarquer qu'une telle méthode peut être envisagée. En effet, nous verrons que, malgré notre nombre limité de patrons, les résultats obtenus par Anasem sont encourageants (chapitre 4).

1.4 CoNLL 2008

Puisque nous utilisons la campagne d'évaluation de CoNLL 2008 pour faire nos comparaisons, il est important de bien comprendre en quoi elle consiste et quelles ressources ont été mises à la disposition des participants.

D'abord, CoNLL (« *Conference on Natural Language Learning* ») est la conférence annuelle de SIGNLL[4] (« *Special Interest Group on Natural Language Learning* »). Elle a pour but de regrouper les membres internationaux de SIGNLL pour qu'ils puissent présenter et discuter de leur recherche dans un forum commun. Dans le cadre de cette conférence, il y a une « *Shared Task* » qui est en fait une compétition dont le but est de permettre aux chercheurs de comparer leurs méthodes sur un même jeu de données.

Tel que mentionné précédemment, l'objectif de cette compétition en 2008 était l'analyse syntaxique couplée avec l'analyse sémantique, plus précisément l'attribution de rôles sémantiques (SRL). Cette tâche proposait deux approches aux participants, soit l'approche ouverte, c'est-à-dire avec accès à n'importe quelles ressources externes, ou l'approche fermée, c'est-à-dire avec accès à des ressources prédéfinies dans le cadre de la compétition. Bien que l'approche ouverte semble très intéressante puisque les possibilités y sont très grandes, le concept derrière Anasem tend plus vers l'autre approche, soit pousser l'analyse sémantique le plus loin possible en n'utilisant que le minimum de ressources. Nous nous sommes donc concentrés sur l'approche dite fermée. Dans cette approche, les participants avaient accès à trois ressources, PropBank (Palmer, et al., 2005), NomBank (Meyers, et al., 2004) et un corpus, dit corpus d'entrainement, qui est en fait un sous-ensemble du Penn Treebank III (Marcus, et al., 1993) (Surdeanu, et al., 2008) auquel ont été ajoutées des annotations sémantiques provenant des deux autres ressources (NomBank et PropBank).

Un corpus d'entraînement est généralement séparé en trois sections appelées (en anglais) « *train* », « *development* » et « *test* ». La section « *train* » correspond à la plus grande partie du corpus. C'est sur cette section qu'on entraine le système, d'où son nom. La section « *development* », quant à elle, est utilisée après avoir entrainé son système et sert à tester le système afin d'y apporter des correctifs si nécessaire. La section « *development* » et la section « *train* » contiennent toutes les informations extraites du corpus. C'est pourquoi on peut utiliser la section de « *development* » pour vérifier les résultats. La dernière section, « *test* », n'est pas annotée, c'est-à-dire qu'elle ne contient pas les informations recherchées. C'est cette section qui est utilisée à la fin pour évaluer chaque système.

---

[4] http://ifarm.nl/signll/

Dans le cas de CoNLL 2008, le corpus d'entrainement est formé de 4 sections, soit le « *train* », le « *development* » et deux « *test* », le « *Brown* » et le « *Wall Street Journal* » (WSJ). Les sections comprennent un certain nombre de phrases sous la forme de tableaux qui respectent le format de CoNLL (voir ci-dessous). La section « train » contient 32 279 phrases. La section « *development* » contient 1334 phrases. Finalement, la section « *test* » provenant du WSJ, contient 2399 phrases et celle qui provient du « *Brown* » contient 425 phrases (Dans ce mémoire, lorsque nous ferons référence à la section « *test* », nous parlons du « *Brown* »). Techniquement, il n'y a pas de différences entre les sections, le corpus est subdivisé de cette façon pour faciliter l'apprentissage et les tests. Les 4 sections contiennent des phrases similaires, ce qui permet d'éviter les biais.

Dans le corpus d'entrainement de CoNLL 2008, chaque phrase du corpus est représentée par un tableau dans lequel se retrouvent les informations pertinentes pour son analyse. Toutes les sections du corpus, à l'exception du « *test* », sont complètement remplies, c'est-à-dire que les tableaux contiennent déjà les résultats de l'analyse. On appelle ces informations le « Gold Standard », ce qu'on pourrait qualifier de solution « idéale ». Pour toutes nos expérimentations, nous utilisons uniquement un sous-ensemble de ces tableaux comme valeur d'entrée. Chaque tableau est constitué d'un minimum de 12 colonnes dont voici les noms et descriptions :

Tableau 1.1 Explication des colonnes du format CoNLL

Numéro	Nom de la colonne	Description
1	ID	Un compteur pour identifier les mots dans la phrase, il commence toujours à 1 pour chaque phrase.
2	FORM	Le mot tel qu'écrit dans la phrase.
3	LEMMA	Le lemme associé au mot de la colonne précédente.
4	GPOS	La catégorie grammaticale qui provient de TreeBank. Lors de l'évaluation finale, elle n'est pas accessible, et il faut se baser sur la prochaine colonne. Elle sert donc uniquement lors de l'entrainement des systèmes.
5	PPOS	La catégorie grammaticale prédite par un analyseur de l'état de l'art(Giménez & Marquez, 2004).
6	SPLIT_FORM	Un espace réservé pour les mots qui sont liés par un trait d'union. Dans certains cas, il est nécessaire de les traiter séparément. C'est aussi pour cette raison que pour ce type de mots, des colonnes additionnelles sont ajoutées (colonnes 7 et 8).
7	SPLIT_LEMMA	Le lemme prédit pour la colonne SPLIT_FORM en utilisant WordNet.
8	PPOSS	La catégorie grammaticale de la colonne SPLIT_FORM en utilisant un analyseur de l'état de l'art et en validant avec le PPOS.
9	HEAD	La tête syntaxique à laquelle se relie le mot courant.
10	DEPREL	La relation de dépendance que possède le mot avec sa tête.
11	PRED	Les prédicats de la phrase, en incluant à la fois les noms et les verbes avec NomBank et PropBank comme références.
12+	ARG	Les colonnes suivantes représentent les arguments correspondant aux prédicats identifiés dans la colonne PRED. Les arguments se relient au prédicat dans l'ordre dans lequel ils apparaissent, c'est-à-dire que la première colonne d'argument se rattache au premier prédicat de la phrase, que la deuxième colonne se rattache au deuxième prédicat et ainsi de suite.

En ce qui concerne l'analyse sémantique, ce sont les colonnes PRED et ARG qui nous intéressent. La colonne PRED, qui représente les prédicats, prend une forme simple. Pour chaque ligne qui correspond à un prédicat, sa forme lemmatisée est inscrite dans la colonne PRED suivie d'un point et du numéro qui correspond au sens du mot selon NomBank ou PropBank. Pour ce qui est des colonnes ARG (colonnes 12 et suivantes), elles servent aux arguments se rapportant aux prédicats de la phrase. Il existe plusieurs types d'arguments possibles. Voici un tableau qui

présente les plus importants de ces arguments. (Dans les exemples, les mots en gras à gauche des parenthèses correspondent à l'argument (ou la tête de l'argument) du type inscrit à l'intérieur de la parenthèse, les mots en gras à droite des parenthèse sont les dépendants de l'argument, lorsque celui-ci en possède)

Tableau 1.2: Description des types d'arguments sémantiques[5]

Types d'arguments	Descriptions	Exemples
A0,A1, A2, A3...	Argument associé au verbe ou au nom auquel il se rapporte. Chaque prédicat (dans PropBank pour les verbes ou NomBank pour les noms) définit le sens de ses arguments.	**John** (A0) sang the **song** (A1). Dans l'exemple, « to sing » définit A0 comme étant la personne qui chante et A1 comme étant ce qui est chanté.
AM-CAU	Un argument de type « Cause Clauses ». Il indique la raison d'une action.	The doctor, **since**(AM-CAU) Scotty was sick, came often to the house.
AM-LOC	Un argument de type « Locatives ». Il indique où une action prend place. Une place peut être physique ou abstraite.	They had brandy **in**(AM-LOC) **the library**.
AM-TMP	Un argument de type « Temporal ». Il indique quand une action a pris place. Il inclut aussi les adverbes de fréquence, les adverbes de durée, les ordres (« first », etc.) et les répétitions.	**Soon**(AM-TMP) they were picking their way along the edge of the stream.
AM-PNC	Un argument de type « Purpose Clauses » (PNC: Purpose, Not Cause). Il indique la motivation d'une action.	Scotty was no longer allowed to make his regular trips into town **to**(AM-PNC) **see the doctor**.
AM-EXT	Un argument de type « Extent ». Il indique une quantité de changements qui affecte une action. Il est utilisé essentiellement pour les quantificateurs comme « a lot », les compléments numériques et les comparatifs.	About 10,000 diamond miners struck for **higher**(AM-EXT) wages.
AM-MNR	Un argument de type « Manner». Il indique comment une action a été faite. Les adverbes qui sont la réponse à la question « how » sont de ce type.	Promises of a **cleaner**(AM-MNR) bill are suspect.

[5] http://clear.colorado.edu/compsem/documents/propbank_guidelines.pdf et http://nlp.cs.nyu.edu/meyers/nombank/nombank-specs-2007.pdf

Tableau 1.2: Description des types d'arguments sémantiques (suite)

AM-ADV	Un argument de type « Adverbial ». Il indique une modification de la structure des événements d'un verbe. Ce type d'arguments est utilisé lorsque l'argument n'appartient à aucune autre catégorie.	His recording later turned up **as**(AM-ADV) **a court exhibit**.
AM-DIS	Un argument de type « Discourse ». Il indique un lien avec une autre phrase, par exemple « also, however, etc. ». Il inclut aussi les interjections et les vocatifs. Dans la plupart des cas, ce type d'arguments n'affecte pas le sens de la phrase.	**However**(AM-DIS), they decided not to sue him.
AM-MOD	Un argument de type « Modals ». Les arguments « Modals » sont: « will, may, can, must, shall, might, should, could, would ».	Problems of the junk market **could**(AM-MOD) prompt the Federal Reserve to ease credit
AM-DIR	Un argument de type « Directional ». Il indique le déplacement dans une direction quelconque. Lorsqu'il n'y a pas de direction, ce type d'arguments est considéré comme un « Locatives».	He went **down**(AM-DIR) in his bathrobe and slippers to have breakfast.
AM-NEG	Un argument de type « Negation ». Il indique un marqueur de négation. Dans les cas où un argument peut avoir plus d'une catégorie, la négation est toujours choisie.	He had come because he could **not**(AM-NEG) live out his life feeling that he had been a coward.
SU	Un argument de type « Support Chain ». C'est un type d'arguments qui provient de NomBank, il représente une chaine de dépendance dans les données.	His **share of** (SU) accomplishments. Dans cet exemple, la « Support Chain » est « share + of » et cette chaine supporte « accomplishments ».
R-*	Une référence à un autre type d'arguments.	I heard of some **people**(A0) **that**(R-A0) tried it back in the States.
C-*	Un argument qui est la suite d'un autre argument.	The doctor, since Scotty was no longer allowed to make his regular **trips**(A1) **into**(C-A1) **town** to see him, came often and informally to the house.

Voici maintenant un exemple concret (Tableau 1.3) de la représentation d'une phrase avec la méthode utilisée lors de la campagne de CoNLL. La phrase utilisée pour cet exemple est : « *You needn't be a high-powered securities lawyer to realize the prospectus is guilty of less than full disclosure.* » (Phrase tirée de la section de développement du corpus)

Tableau 1.3: Exemple du format CoNLL

ID	FORM	LEMMA	GPOS	PPOS	SPLIT_FORM	SPLIT_LEMMA	PPOSS	HEAD	DEPREL	PRED	ARG	ARG	ARG
1	You	you	PRP	PRP	You	you	PRP	2	SBJ	_	_	_	_
2	need	need	MD	VBP	Need	need	VBP	0	ROOT	_	_	_	_
3	n't	not	RB	RB	n't	not	RB	2	ADV	_	_	_	_
4	be	be	VB	VB	be	be	VB	2	VC	_	_	_	_
5	a	a	DT	DT	a	a	DT	10	NMOD	_	_	_	_
6	high-powered	high-powered	JJ	JJ	high	high	RB	8	HMOD	_	_	_	_
7	_	_	_	_	-	-	HYPH	6	HYPH	_	_	_	_
8	_	_	_	_	powered	powered	JJ	10	NMOD	_	_	_	_
9	securities	security	NNS	NNS	securities	security	NNS	10	NMOD	_	A1	_	_
10	lawyer	lawyer	NN	NN	lawyer	lawyer	NN	4	PRD	lawyer.01	A0	A0	_
11	to	to	TO	TO	to	to	TO	4	PRP	_	_	_	_
12	realize	realize	VB	VB	realize	realize	VB	11	IM	realize.01	_	_	_
13	the	The	DT	DT	the	the	DT	14	NMOD	_	_	_	_
14	prospectus	prospectus	NN	NN	prospectus	prospectus	NN	15	SBJ	_	_	_	_
15	is	Be	VBZ	VBZ	is	be	VBZ	12	OBJ	_	_	A1	_
16	guilty	guilty	JJ	JJ	guilty	guilty	JJ	15	PRD	_	_	_	_
17	of	Of	IN	IN	of	of	IN	16	AMOD	_	_	_	_
18	less	Less	RBR	JJR	less	less	JJR	17	PMOD	_	_	_	_
19	than	than	IN	IN	than	than	IN	18	NMOD	_	_	_	_
20	full	Full	JJ	JJ	full	full	JJ	21	NMOD	_	_	_	AM-MNR
21	disclosure	disclosure	NN	NN	disclosure	disclosure	NN	19	PMOD	disclosure.01	_	_	A1
22	.	.	.	.	.	.	.	2	P	_	_	_	_

Comme on peut le constater dans l'exemple, cette phrase contient trois prédicats qui sont indiqués dans la colonne « PRED ». Par conséquent, il y a trois colonnes supplémentaires pour les arguments (12, 13 et 14). Dans le cas présent, il y a deux arguments qui sont associés à chacun des prédicats. Le premier prédicat est « *lawyer*.01 », il possède les arguments « A0 », qui est défini dans PropBank comme étant la personne qui fait ce travail, et « A1 », qui est défini comme étant le thème. Donc, « A0 » est « *lawyer* », lui-même, et « A1 » est « *security* ». Pour le deuxième prédicat, « *realize.01* », on constate que son argument « A0 », qui, dans PropBank, représente la personne qui réalise, est « *lawyer* » et que son argument « A1 », qui représente ce que la personne réalise, est « *is* ». Ici, « *is* » est la tête du sous-arbre qui représente le « A1 » de « *realize*», soit « *the prospectus is guilty of less than full disclosure*». Finalement, le troisième prédicat est « *disclosure*.01 » et il possède un « A1 » et un « AM-MNR ». Le « A1 », selon NomBank, représente ce qui est dit, soit « *disclosure* » lui-même. Ce prédicat possède aussi un argument modificateur qui indique la manière dont il se produit, dans ce cas-ci cet argument est « *full* ».

Dans le cadre de notre comparaison, nous n'utilisons que les colonnes ID, FORM, PPOS, HEAD, DEPREL qui sont les colonnes importantes pour l'analyse sémantique. Nous ne considérons pas les « *split form*» dans la version actuelle d'Anasem. De plus, puisque nous n'avons pas accès aux GPOS lors de l'évaluation, la colonne 4 (Tableau 1.1) nous est inutile.

Une fois que l'analyse sémantique est effectuée, il faut pouvoir comparer les résultats au « Gold Standard » au moyen d'un ensemble de métriques. C'est ce dont nous traitons dans la section suivante.

1.5 Métriques d'évaluation

Le traitement automatique des langues (TAL) est un domaine qui englobe plusieurs sous-tâches comme la traduction automatique, la fouille de textes, la reconnaissance vocale, etc. La plupart de ces tâches ont pour objectif d'extraire des informations à partir d'un texte (ou d'une représentation d'un texte). Cependant, ces informations possèdent deux aspects : leur présence et leur type. Prenons par exemple un système qui extrait la relation entre un verbe et son complément (direct ou indirect). Supposons que ce système extrait de la phrase « Pierre mange une pomme » la relation entre « mange » et « pomme », et qu'il l'identifie comme étant un

complément indirect. Dans un premier temps, l'information extraite est pertinente, il y a bien une relation verbe-complément entre « mange » et « pomme ». Par contre, le type « complément indirect » est incorrect. C'est pour séparer ces résultats qu'il existe deux catégories : les résultats étiquetés, où la présence et le type doivent être corrects, et les résultats non-étiquetés, où seule la présence de l'information est vérifiée. Dans notre exemple, l'information est considérée comme correcte pour des résultats non-étiquetés, mais incorrecte pour des résultats étiquetés. Cette distinction est faite notamment en analyse sémantique où les informations extraites possèdent toujours un type.

Après avoir séparé les catégories, étiquetés et non-étiquetés, il faut évaluer ces résultats. Pour ce faire, on utilise souvent la mesure de la précision, la mesure du rappel, et la F-mesure, qui est une combinaison des deux mesures précédentes. Pour calculer ces mesures, il faut avoir accès à un corpus de référence (un « Gold Standard ») pour la tâche en question. Ces mesures se basent sur trois informations : les informations qu'on peut extraire, les informations extraites par le système et les informations extraites qui sont correctes. La précision est le ratio entre l'ensemble des informations extraites correctement et les informations extraites dans cet ensemble :

$$Précision = \frac{informations\ correctes\ extraites}{informations\ extraites}$$

Dans le cas d'un système qui extrait beaucoup d'informations incorrectes, la précision sera basse. Par contre, avec un système qui extrait un très petit nombre d'informations où la plupart sont correctes, la précision sera élevée. Supposons qu'une tâche nous demande d'identifier 124 items dans un texte et que nous en avons identifié 50 dont 41 sont correctes (9 items identifiés sont faux). En calculant la précision pour cet exemple (41/50= 82 %), on constate que bien que nous n'ayons identifié que 50 items, nous possédons quand même une précision élevée.

C'est entre autres pourquoi il faut aussi calculer une seconde mesure appelée le rappel. Celle-ci est le ratio entre les informations correctes extraites et l'ensemble des informations correctes qui aurait pu être extraites:

$$Rappel = \frac{informations\ correctes\ extraites}{toutes\ les\ informations}$$

Cette mesure nous permet d'évaluer la quantité d'information qui n'est pas extraite en ignorant les informations incorrectes. En calculant le rappel pour l'exemple précédent (41/124 = 33.1 %), on se rend compte que nous avons identifié seulement 33.1 % des items et donc il nous en manque 66.9 %. En comparant la précision et le rappel de notre exemple, on se rend compte que si on n'utilise qu'une seule mesure le résultat peut être trompeur.

Finalement, il existe une mesure qui combine la précision et le rappel, la F-mesure (ou F_1 score). Celle-ci nous permet de savoir si une mesure compense l'autre ou pas.

$$F - mesure = 2 * \frac{précision * rappel}{précision + rappel}$$

Naturellement, ces mesures doivent être appliquées sur les mêmes données pour qu'elles soient représentatives. On peut calculer cette mesure pour notre exemple (2* 33.1* 82 / (82+33.1) = 47.2 %) et on constate qu'elle est plus représentative des résultats obtenus (50 items identifiés sur les 124 dont 9 sont incorrects).

Dans le cadre de la campagne d'évaluation de CoNLL 2008, les organisateurs ont créé un outil[6] d'évaluation des résultats. Cet outil permet d'évaluer un système en fonction du « Gold Standard » de CoNLL. Il calcule la précision, le rappel et le F_1 score du système pour l'analyse sémantique et l'analyse syntaxique dans son ensemble, et ce, pour les résultats « *labeled*» et « *unlabeled* ». De plus, il fournit des informations détaillées sur les erreurs en fonction des types d'arguments, ainsi que la ventilation de la précision, du rappel et du F_1 score en fonction des types d'arguments et de prédicats. Nous avons utilisé cet outil pour obtenir les résultats de LTH. Toutefois, cet outil ne pouvait fonctionner avec notre système à base de règles Anasem dû à son format différent. Nous décrivons ce format ainsi que les autres particularités d'Anasem dans le chapitre suivant.

[6] http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=conll2008:software

CHAPITRE 2 ANASEM, UN SYSTÈME D'ANALYSE SÉMANTIQUE À BASE DE PATRONS

L'analyseur à base de règles choisi pour faire la comparaison avec un système basée sur l'apprentissage machine s'appelle Anasem. C'est un système d'analyse sémantique basé sur la notion de patrons dans des grammaires de dépendances. L'utilisation de tels patrons se base sur le travail de (Zouaq, 2008) pour l'extraction d'ontologies du domaine (OWL) à partir des textes. Basé sur ces patrons initiaux, Anasem a par la suite été étendu par de nouveaux patrons et par la notion de composition, permettant ainsi la génération de représentation du discours (DRS) (Zouaq, et al., 2010). Une des raisons majeures qui nous a portés à choisir cet analyseur est que nous avions accès au code source ainsi qu'à ses créateurs. Cela nous a grandement facilité la tâche. Dans ce chapitre nous expliquons la représentation choisie pour exprimer les résultats obtenus. Ensuite, nous présentons en détail l'architecture particulière de la version d'Anasem avant les transformations en explorant ses trois parties majeures : l'analyse syntaxique, le générateur d'arbres canoniques et l'identification des patrons sémantiques. Finalement, nous exposons les résultats antérieurs, c'est-à-dire le rappel et la précision, obtenus par Anasem lors de son évaluation sur un corpus de phrases extraites de contes pour enfants.

2.1 Le modèle de connaissances

Anasem (Zouaq, et al., 2010) est basé sur un modèle de connaissances qui possède 8 catégories différentes : *Entity* (ou *Named Entity*), *Event*, *Statement*, *Circumstance*, *Time*, *Number*, *Measure* et *Attribute*. Ces catégories permettent d'annoter les prédicats et arguments des phrases et sont associées à des catégories syntaxiques spécifiques.

Tableau 2.1: Les catégories du modèle et les catégories syntaxiques applicables[7]

Catégories du modèle	**Catégories Syntaxiques**	**Exemples**
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 heat in the garden**
Circumstance	Toute relation *advcl* (adverbial clause)	The accident happened **as the night was falling**
Time	Toute relation *tmod* (Temporal modifier)	He swam in the pool **last night**
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

Afin de représenter les résultats, Anasem utilise une version légèrement modifiée de la représentation en boites de la DRS(Blackburn & Bos, 2005). Dans nos DRS, il y a deux composants majeurs : les référents et les conditions. Les référents sont les identifiants des entités et des événements de la phrase. Les conditions sont les relations qui relient les identifiants afin de donner un sens à la phrase. Par exemple, dans la phrase « *They drank brandy in the lounge.* », il y a quatre référents, soit trois entités (*they*, *brandy*, *lounge*) et un événement (*drank*). Dans la section condition, il y a les associations des identifiants aux entités, l'association de l'identifiant d'événement à celui-ci et la relation entre « *lounge* » et l'événement. La DRS a donc la forme suivante (elle a été légèrement simplifiée afin de faciliter la compréhension):

[7] Directement extrait de (Zouaq, Gagnon, & Ozell, Grammaire de dependances et ontologies de haut niveau: vers un processus modulaire pour l'analyse semantique, 2010)

```
---------------------------------
[id1,id2,e1,id3]
---------------------------------
entity(id1,they)
entity(id2,brandy)
event(e1,drank,id1,id2)
entity(id3,lounge)
in(e1,id3)
---------------------------------
```

Toutes nos conditions respectent une certaine forme. Pour les entités, comme on peut le constater dans l'exemple, c'est la relation « `entity` », suivi de l'identifiant qui lui est associé et du mot qu'il représente. Un identifiant d'entité prend toujours la forme de « `id` » suivi d'un chiffre.

Pour les événements, c'est la relation « `event` » suivie de l'identifiant qui le représente, suivis de l'événement (le verbe) et des entités qui s'y rapportent. Un identifiant d'événement prend toujours la forme de « `e` » suivi d'un chiffre. Dans notre exemple, il y a deux entités qui sont associées avec l'événement « *drank* », soit ceux qui boivent « *they* » (sujet) et ce qu'ils boivent « *brandy* » (complément d'objet direct). Il est parfois possible d'avoir plus ou moins d'entités reliées à un événement. Par exemple, dans la phrase « *He came.* », l'événement « *came* » n'a qu'une seule entité associée, soit « *he* ». Par contre, dans la phrase « *Mary gave Bill a raise.* », l'événement « *gave* » est relié à trois entité: « *Mary* », « *raise* » et « *Bill* ».

Toutes les autres relations (*attributes*, *in*, etc.) peuvent être regroupées en deux catégories : les relations unaires et les relations binaires. Les relations unaires, comme les « *attributes*» (***big cat***), apparaissent sous la forme suivante : la relation, suivie de l'identifiant qui est affecté et le mot comme tel. Les relations binaires, comme « *in* », apparaissent sous la forme suivante : la relation, suivie des identifiants des deux entités/événements qui sont affectés.

2.2 L'architecture d'Anasem

Une des particularités principales d'Anasem est son architecture modulaire. Concrètement, il est composé de trois modules distincts, soit : l'analyseur syntaxique, le générateur d'arbres canoniques et l'analyseur sémantique en Prolog. De cette façon, il est possible de modifier ses différentes parties sans affecter l'ensemble de l'analyseur. Chacun des modules est responsable d'une partie de l'analyse sémantique, et c'est en les combinant qu'on obtient notre analyse sémantique sous forme de DRS.

2.2.1 L'analyseur syntaxique

L'analyse syntaxique est la première étape dans le processus modulaire. Elle prépare le terrain pour les étapes consécutives. Dans ce cas-ci, c'est l'analyseur syntaxique de Stanford (Klein & Manning, 2003) qui est utilisé, son module de dépendance (De Marneffe, et al., 2006) et son étiqueteur de catégories grammaticales (Toutanova, et al., 2003). L'analyse syntaxique est générée sous forme d'arbre de dépendances. Un arbre de dépendances est une représentation de la phrase sous la forme d'un arbre où les mots sont liés les uns aux autres en fonction de leurs dépendances. Voici un exemple d'arbre de dépendance tiré de Wikipedia.org[8]:

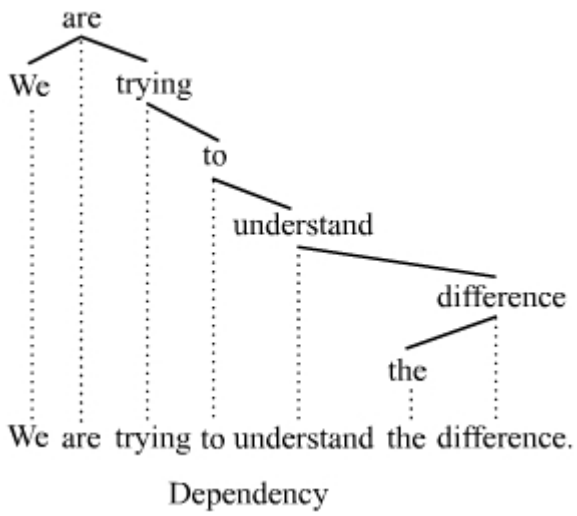

Figure 2.1: Exemple d'arbre de dépendance

L'analyseur de Stanford génère un arbre qui représente la phrase et qui contient les mots et leur tête de manière similaire à la Figure 2.1 (la tête d'un mot est le mot duquel il dépend). L'analyseur de Stanford se base sur une hiérarchie de relations grammaticales qui a pour racine la relation grammaticale générique dep « *dependant*». Tout d'abord, en utilisant le module de dépendance, l'analyseur associe le type dep à chaque relation syntaxique. Ensuite, à l'aide de patrons, le module de dépendance parcourt l'arbre généré pour raffiner le type de chaque relation le plus possible. Par exemple, « *dependant* » peut être raffiné en « *arg*», puis en « *subj* » et finalement en « *nsubj* » (l'ensemble des relations est disponible dans (Marneffe, et al., 2006)).

[8] http://en.wikipedia.org/wiki/File:Wearetryingtounderstandthedifference_(2).jpg

Nous utilisons le formalisme par dépendances car il possède des caractéristiques importantes pour l'analyse sémantique : il est facile à comprendre, avec des relations simples qui lient tous les mots d'une phrase deux à deux, il met en évidence une structure prédicat-arguments, grâce à ses dépendances, et enfin les systèmes de l'état de l'art utilisant cette méthode sont robustes (De Marneffe & Manning, 2008). Reprenons la phrase de notre exemple précédent: « *They drank brandy in the lounge.* ». Avec l'analyseur de Stanford, nous obtenons les catégories grammaticales ainsi que l'analyse par dépendances suivantes :

Catégories grammaticales :

```
They/PRP
drank/VBD
brandy/NN
in/IN
the/DT
lounge/NN
./.
```

Relations de dépendances :

```
nsubj(drank-2, They-1)
dobj(drank-2, brandy-3)
prep(drank-2, in-4)
det(lounge-6, the-5)
pobj(in-4, lounge-6)
```

Dans un premier temps, on obtient les catégories grammaticales (PoS) pour chaque mot, suivi des relations de dépendances pour les différentes paires de mots. Les relations sont présentées de la façon suivante : d'abord le nom de la relation, puis la paire de mots. Chaque mot est suffixé d'un trait d'union et du numéro correspondant à la position du mot dans la phrase. Le premier mot de la paire constitue la tête de la relation de dépendance.

Une fois l'analyse syntaxique complétée, on utilise sa sortie, soit les catégories grammaticales et les relations de dépendances, pour exécuter les étapes suivantes.

2.2.2 Le générateur d'arbres canoniques

La deuxième étape dans le procédé modulaire consiste à construire un arbre canonique, qui facilite l'identification des patrons de manière compositionnelle. En effet, en considérant une phrase comme une composition de sous-parties (principe de compositionalité (Montague, 1970)), cela nous permet d'appliquer des patrons à ces sous-parties. La somme des patrons est censée représenter la phrase au complet.

Puisque nous utilisons un analyseur sémantique en Prolog, nous devons convertir les informations en un terme Prolog. Il est important que ce terme, qui doit avoir une structure en arbre, contienne à la fois les relations de dépendance, mais aussi les catégories grammaticales. Ces dernières nous permettent parfois de différencier certains patrons dans les étapes subséquentes.

Pour créer cet arbre, nous utilisons une méthode de composition récursive. Chaque partie de l'arbre prend la forme suivante : « RELATION/tree(token(WORD, POSITION)/POS, [CHILDREN]) », où RELATION est la relation grammaticale, WORD est le mot, POSITION est la position du mot dans la phrase, POS est la catégorie grammaticale et CHILDREN représente les enfants (les sous-partie de l'arbre) de cette relation. Les enfants respectent le même formalisme. Toutefois, la première relation, la tête, est toujours « `root` ». En prenant l'exemple précédant, le terme Prolog obtenu est le suivant :

```
root/tree(token(drank,2)/v,[nsubj/tree(token(they,1)/prp,[]),dobj/tree(token(b
randy,3)/n,[]),prep/tree(token(in,4)/prep,[pobj/tree(token(lounge,6)/n,[det/tr
ee(token(the,5)/d,[])])])])
```

On peut réécrire ce terme sous la forme d'un arbre et retirer les éléments inutiles pour la compréhension humaine:

```
root drank/v
  nsubj they/prp
  dobj brandy/n
  prep in/prep
    pobj lounge/n
      det the/d
```

Il est parfois nécessaire de modifier l'arbre obtenu pour mettre en évidences certains patrons, ou pour faciliter l'analyse. Voici la liste des situations, accompagnées d'exemples, pour lesquelles il a été nécessaire de créer des règles de transformation :

1. Énumérations avec conjonctions (conj)

   En prenant la phrase « *I love cats, dogs and rats* », on obtient la phrase suivante : « *I love cats and I love dogs and I love rats* ».

Avant	Après

root love/v
  nsubj i/prp
  dobj cats/n
  conj dogs/n
  cc and/cc
  conj rats/n
	root and/cc
	  coord love/v
	  dobj cats/n
	  nsubj i/prp
	  coord and/cc
	  coord love/v
	  dobj rats/n
	  nsubj i/prp
	  coord love/v
	  dobj dogs/n
	  nsubj i/prp

2. L'utilisation de coordinations (cc)

   Il existe cinq règles de transformations pour les coordinations, voici un exemple de la plus commune. En prenant la phrase « *The peasant eats an orange and an apple* », on obtient la phrase suivante : « *The peasant eats an orange and the peasant eats an apple* ».

Avant	Après

root/eats
  nsubj/peasant
  det/the
  dobj/orange
  det/an
  cc/and
  conj/apple
  det/an
	root/and
	  coord/eats
	  dobj/orange
	  det/an
	  nsubj/peasant
	  det/the
	  coord/eats
	  dobj/apple
	  det/an
	  nsubj/peasant
	  det/the

3. L'utilisation de négations (neg)

   En prenant la phrase « *the peasant does not eat an apple* », on obtient la même phrase, mais l'arbre de dépendances possède une racine différente et de ce fait même, une hiérarchie différente.

Avant	Après
root eat/v nsubj peasant/n det the/d aux does/v neg not/rb dobj apple/n det an/d	root not/rb nsubj peasant/n det the/d neg eat/v aux does/v dobj apple/n det an/d

4. L'utilisation de compléments de clauses sans sujet (xcomp)

   En prenant la phrase « *Paul likes to eat fish* », on obtient la phrase suivante : « *Paul likes Paul to eat fish* ».

Avant	Après
root likes/v nsubj paul/nnp xcomp eat/v aux to/prep dobj fish/n	root likes/v nsubj paul/nnp xcomp eat/v nsubj paul/nnp aux to/prep dobj fish/n

5. L'utilisation de particules (prt)

   En prenant la phrase « *Alice fends off the sharks* », on obtient la même phrase, mais la particule est fusionnée avec sa tête pour donner un seul mot.

Avant	Après
root fend/v nsubj alice/nnp prt off/prt dobj sharks/n det the/d	root fend-off/v nsubj alice/nnp dobj sharks/n det the/d

6. L'utilisation de compléments prépositionnels (pcomp)

   En prenant la phrase « *Alice tired herself with trying out* », on obtient la phrase suivante : « *Alice tired herself with Alice trying out* ».

Avant	Après
root tired/v nsubj alice/nnp dobj herself/prp prep with/prep pcomp trying/v prt out/prt	root tired/v prep with/prep pcomp trying-out/v nsubj alice/nnp nsubj alice/nnp dobj herself/prp

7. L'utilisation de modificateurs nominaux (nn)

   En prenant la phrase « *Intelligent tutoring systems are useful* », on obtient la même phrase, mais le modificateur nominal est fusionné avec sa tête pour ne donner qu'un mot.

Avant	Après
root useful/jj nsubj systems/n amod intelligent/jj nn tutoring/n cop are/v	root useful/jj nsubj turoring-systems/n amod intelligent/jj cop are/v

8. L'utilisation de modificateurs adverbiaux (advmod)

   En prenant la phrase « *Genetically modified food is dangerous* », on obtient la même phrase, mais le modificateur adverbial est fusionné avec sa tête pour ne donner qu'un mot.

Avant	Après
root dangerous/jj nsubj food/n amod modified/jj advmod genetically/rb cop is/v	root dangerous/jj nsubj food/n amod genetically-modified/jj cop is/v

Prenons par exemple la phrase : « *Eric visited Montreal and Quebec* ». Comme cette phrase possède une coordination, l'arbre initial est transformé de manière à obtenir une signification distributive du « and ». Le nouvel arbre représente la phrase suivante : « *Eric visited Montreal and Eric visited Quebec* ». Cette méthode a pour avantage de simplifier la tâche de l'identification des patrons, en séparant la phrase afin que chaque partie soit clairement définie et indépendante. On constate, dans l'exemple, que les deux phrases ont le même sens, bien que la formulation soit différente. Toutefois, cette sémantique distributive peut causer des problèmes lorsqu'il y a des imbrications. Par exemple, dans la phrase " *The young man and the old woman are talking and dancing.*", on se retrouve avec des informations dupliquées. Voici la DRS obtenue :

```
--------------------------------
[id1,e1,id2,e2,e3,e4]
--------------------------------
entity(id1,man)
attribute(id1,young)
event(e1,talking,id1)
entity(id2,woman)
attribute(id2,old)
event(e2,talking,id2)
attribute(id1,young)
event(e3,dancing,id1)
attribute(id2,old)
event(e4,dancing,id2)
--------------------------------
```

On constate que l'attribut « *young*» et « *old* » sont présent deux fois chacun. C'est parce que la phrase est presque entièrement dupliquée à cause de la double distribution. Une fois les distributions effectuées, la phrase prend la forme suivante : « *The young man and the old woman are talking and the young man and the old woman are dancing.* », d'où les attributs doubles. Un autre problème provient des phrases avec « *and* » dont la sémantique n'est pas distributionnelle. Par exemple, dans la phrase « *Amal and Michel lift the table* », la stratégie de transformation donne une représentation erronée du sens de la phrase.

2.2.3 Patrons en Prolog

La dernière étape de l'analyse sémantique consiste à identifier des patrons syntaxiques afin d'extraire les relations sémantiques définies précédemment. Tout d'abord, lorsqu'on parle de patron, on parle d'une représentation syntaxique particulière. Par exemple, on a un patron qui identifie, dans un arbre de dépendance, la relation entre un nom et un déterminant. Pour ce faire, on cherche la relation « *det* » dans l'arbre syntaxique. Voici un extrait d'un arbre de dépendance qui montre cette relation :

```
nsubj man/n
    det the/d
```

Cette analyse est faite de manière récursive, c'est-à-dire que pour tout patron identifié, chaque composant de celui-ci est aussi analysé par d'autres patrons, et ainsi de suite, jusqu'à ce qu'il n'y ait plus de composant non analysé. De cette façon, nous nous assurons de couvrir l'ensemble de la phrase. De plus, il existe une hiérarchie des patrons (basée sur l'ordre des patrons en Prolog) qui traite les patrons les plus importants d'abord.

En totalité, Anasem comprend actuellement environ 44 patrons qui représentent des concepts linguistiques. Ces patrons sont regroupés en trois catégories : les patrons principaux, les

patrons modificateurs et les autres. Les « autres » incluent un patron pour le traitement des sujets en général et un patron pour filtrer les pronoms. Dans les patrons principaux, il y a 26 patrons qui traitent les différentes combinaisons sujets-verbes-compléments, les pronoms, les noms propres, les prépositions, etc. Dans la catégorie des modificateurs, il y a 16 patrons qui traitent entre autres : les compléments propositionnels, les prépositions modifiant un verbe, les modificateurs adjectivaux, etc.

Voici un exemple de patron syntaxique (Patron_Nom_Commun) de la catégorie des patrons principaux pour l'extraction des noms communs :

```
semparseMainPattern(tree(Node/n,Children),tree(Node/n,Rest),Id,Id,Sem,SemOut):-
     var(Id),
     select(det/tree(Det/_,_),Children,Rest),!,

     generate(Id),
     semparseDet(Det,Id,Sem,Sem1),

     mergeDRS(Sem1,drs([Id],[entity(Id,Node)]),SemOut.
```

Il y a essentiellement 3 sections dans tous les patrons. D'abord, on vérifie si le patron s'applique à l'analyse syntaxique actuelle (les « *select* » dans l'exemple). Par la suite, on vérifie si les composants du patron possèdent eux-mêmes des patrons syntaxiques (les « *semparse* » dans l'exemple). Finalement, on ajoute à la DRS les nouvelles relations en fonction du patron identifié (le « *mergeDRS* » dans l'exemple).

Voici un exemple de l'analyse complète d'une phrase (« *The young man is dancing with a woman.* ») simple qui utilise le patron présenté. On obtient l'arbre canonique suivant: (dans cette exemple, l'arbre ne nécessite pas de transformations)

```
root dancing/v
  nsubj man/n
    det the/d
    amod young/jj
  aux is/v
  prep with/prep
    pobj woman/n
      det a/d
```

Ensuite, on tente de faire correspondre des parties de l'arbre aux patrons. Dans ce cas-ci, il y a quatre patrons qui correspondent à cette représentation. D'abord, le patron présenté (Patron_Nom_Commun), qui nous permet d'ajouter une nouvelle entité à notre DRS, qui est vide pour le moment :

```
---------------------------------
[id1]
---------------------------------
entity(id1,man)
---------------------------------
```

En continuant dans l'arbre, on arrive à extraire un deuxième patron, soit l'utilisation de « *amod*» que l'on représente comme un attribut. On peut donc ajouter des informations dans notre DRS:

```
---------------------------------
[id1]
---------------------------------
entity(id1,man)
attribute(id1,young)
---------------------------------
```

Comme nous avons atteint une feuille de l'arbre et qu'il n'y a plus d'autres feuilles à ce niveau, il faut remonter dans l'arbre pour trouver le prochain patron. Celui-ci concerne la présence d'un « *nsubj* », ce qui implique la présence d'un événement. On ajoute donc ces nouvelles informations dans la DRS :

```
---------------------------------
[id1,e1]
---------------------------------
entity(id1,man)
attribute(id1,young)
event(e1,dancing,id1)
---------------------------------
```

Ensuite, le prochain patron est le même qu'au début (*Patron_Nom_Commun*), la présence d'un nom avec un déterminant. On ajoute ces informations à la DRS :

```
---------------------------------
[id1,e1,id2]
---------------------------------
entity(id1,man)
attribute(id1,young)
event(e1,dancing,id1)
entity(id2,woman)
---------------------------------
```

Finalement, on arrive au dernier patron, qui concerne la préposition « *with* ». Le patron ajoute la relation entre « *woman* » et l'événement pour obtenir la DRS finale.

```
---------------------------------
[id1,e1,id2]
---------------------------------
entity(id1,man)
attribute(id1,young)
event(e1,dancing,id1)
entity(id2,woman)
with(e1,id2)
---------------------------------
```

C'est de cette façon que l'annotateur Prolog extrait les relations sémantiques par l'intermédiaire de patrons. Toutes les phrases suivent le même procédé d'analyse. On peut bien voir l'importance de pouvoir traiter les phrases de manière compositionnelle.

2.3 Résultats d'Anasem

Dans le passé, Anasem a été évalué sur un corpus de 185 phrases extraites de contes pour enfants(Zouaq, et al., 2010). Ces phrases ont été annotées manuellement pour mettre en évidence les événements et les entités de manière à créer un corpus de référence (« Gold Standard »). Une fois l'annotation complétée, ces mêmes phrases ont été analysées par Anasem. Les résultats ont été obtenus à l'aide de métriques standards, soit la précision et le rappel.

Dans le cadre de cette évaluation, voici ce que signifient la précision et le 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*[9]

Le tableau suivant présente les résultats obtenus:

Tableau 2.2: Résultats antérieurs d'Anasem.

	Précision (%)	Rappel (%)
Entités	95.09	80.16
Événements	94.87	85.27

On constate que les résultats des événements et des entités sont très similaires. Particulièrement au niveau de la précision, une différence de 0.22 % est très mince. Toutefois, au niveau du rappel, il y a une différence d'un peu plus de 5 %. On peut conclure qu'il est un peu plus difficile d'identifier les entités que les événements avec Anasem. Malgré la petite taille du

[9] Extrait de l'article (Zouaq, et al., 2010)

corpus d'évaluation, les résultats obtenus par Anasem sont encourageants. Toutefois, on peut noter qu'un corpus de contes est nécessairement composé de phrases très simples. Les tests effectués dans le présent mémoire sur le Penn Treebank nous permettent d'avoir une idée plus réaliste des résultats d'Anasem sur des phrases « du monde réel ».

CHAPITRE 3 MÉTHODOLOGIE DE COMPARAISON ET D'ÉVALUATION

Comme nous avons pu le constater dans la section précédente, Anasem possède sa propre structure de représentation pour l'analyse sémantique, soit les DRS. Nous avons également déjà indiqué que LTH, le système choisi pour la comparaison, produit des cadres sémantiques basés sur NomBank et propBank. Ces différences de formats rendent la comparaison des deux approches très difficile. C'est d'ailleurs une difficulté inhérente à tous les analyseurs sémantiques, car il n'existe pas de standard pour l'évaluation de leur analyse, ni pour le format des analyses.

Comme notre objectif principal dans ce mémoire est précisément de comparer ces deux types d'analyseurs, il y a donc plusieurs étapes nécessaires pour y parvenir, notamment l'adaptation d'Anasem afin de permettre la comparaison. Une fois cela complété, il reste à établir un procédé de comparaison des analyses (DRS versus cadres, modèle de connaissances d'Anasem versus cadres sémantique). Nous voulons comparer à la fois les analyses entre elles et chacune par rapport au « Gold Standard ».

Étant donné que nous avions un accès restreint à LTH, soit seulement l'exécution de l'analyse, et non pas au code source, il était beaucoup plus simple de transformer le format d'Anasem pour le rendre comparable à celui de CoNLL. C'est pourquoi toutes les adaptations décrites dans la section suivante se concentrent sur Anasem. Elles ont pour but de permettre d'utiliser les mêmes relations grammaticales que CoNLL, tout en conservant la logique des patrons en Prolog. De cette façon, la comparaison effectuée par la suite sera d'autant plus fiable qu'elle sera faite sur des systèmes qui utilisent les mêmes informations en entrée.

3.1 Processus d'adaptation

Initialement, nous voulions prendre la sortie d'Anasem et créer un outil qui transformerait celle-ci pour qu'elle puisse être comparée avec la sortie de l'analyseur de Johansson (LTH). Malheureusement, après plusieurs tentatives, nous nous sommes rendu compte que la transformation des verbes et de leurs arguments était possible, malgré certains cas problématiques, mais que les transformations des autres parties des phrases étaient extrêmement difficiles. Par exemple, Anasem identifie tous les prédicats dans une phrase alors que LTH

identifie seulement les prédicats avec des arguments, ou encore pour la représentation des compléments qui sont des groupes de mots, Anasem crée une nouvelle DRS à l'intérieur de la DRS principale alors que LTH ne fait que relier le mot à la tête du complément. Un travail considérable aurait été nécessaire afin de compléter ces transformations. De plus, la différence au niveau de la nomenclature, à la fois syntaxique et sémantique, entraine d'autant plus de complications. Nous avons donc abandonné cette idée au profit d'une comparaison manuelle. Toutefois, afin de rendre les sorties des analyseurs les plus semblables possible, nous avons décidé d'adapter Anasem en conséquence. L'objectif premier de la transformation était de permettre d'utiliser le format d'entrée de CoNLL par notre analyseur. De cette façon, nous pouvions nous assurer que les deux systèmes avaient accès exactement aux mêmes données et donc qu'il n'y avait pas de biais au niveau du format d'entrée. Cela nous permet d'obtenir une comparaison beaucoup plus fiable puisque nous utilisons les mêmes données en entrée.

Pour ce faire, trois sections d'Anasem ont dû être modifiées ou adaptées. Tout d'abord, il a fallu changer l'analyse syntaxique pour pouvoir prendre en compte le format syntaxique CoNLL, plutôt que la sortie de l'analyseur syntaxique de Stanford (Klein & Manning, 2003). En raison de différences dans ces deux analyses, il a fallu modifier (règles présentées à la section 2.2.2) et ajouter de nouvelles règles de transformation de l'entrée CoNLL avant de lancer l'identification des patrons. Finalement, pour la même raison, nous avons dû adapter les patrons afin qu'ils puissent utiliser la nouvelle nomenclature. Toutes ces modifications sont décrites ci-dessous.

3.1.1 Le générateur d'arbres canoniques

Comme mentionné précédemment, Anasem utilise un analyseur syntaxique sur lequel il base son analyse sémantique, en l'occurrence celui fait par Dan Klein et Christopher D. Manning de Stanford (Klein & Manning, 2003). La version originale d'Anasem, c'est-à-dire la version d'Anasem avant d'être adaptée, utilise une représentation en arbre canonique sur laquelle sont identifiés les patrons, qui a été expliquée en section 2.2.2. Puisque nous voulons conserver cette représentation avec l'analyse syntaxique de CoNLL, nous devons reconstruire cet arbre. Prenons par exemple la phrase suivante : « *The accident happened as the night was falling* ». En utilisant l'analyseur de Stanford comme décrit dans le chapitre 2, on obtient un arbre canonique comme celui-ci :

Arbre généré à partir de l'analyse de Stanford et de la transformation canonique
root happened/v nsubj accident/n det the/d advcl falling/v mark as/prep nsubj night/n det the/d aux was/v

La version adaptée d'Anasem n'a accès qu'aux colonnes suivantes du corpus de CoNLL : l'identifiant, la forme, la catégorie grammaticale, la tête de dépendance et la relation syntaxique (Tableau 3.1).

Tableau 3.1: Représentation simplifiée et traduite du format CoNLL de la phrase utilisée pour l'exemple

Identifiant	Forme	Catégorie grammaticale	Tête de dépendance	Relation syntaxique
1	The	DT	2	NMOD
2	accident	NN	3	SBJ
3	happened	VBD	0	ROOT
4	as	IN	3	TMP
5	the	DT	6	NMOD
6	night	NN	7	SBJ
7	was	VBD	4	SUB
8	falling	VBG	7	VC
9	.	.	3	P

En utilisant ces informations comme base d'entrée pour notre analyseur, il est possible de recréer un arbre canonique respectant le format CoNLL, notamment en utilisant les têtes de dépendance. Par contre, le nouvel arbre diffère au niveau de la structure et de la nomenclature, ce qui cause des problèmes lors de l'identification des patrons. Voici le nouvel arbre canonique qu'on obtient en utilisant uniquement ce tableau :

Tableau 3.2: Comparaison entre les arbres canoniques générés à partir du format de CoNLL et de Stanford.

Nouvel arbre généré à partir du format CoNLL	Arbre généré à partir de l'analyse de Stanford

root happened/v
  sbj accident/n
  nmod the/d
  tmp as/prep
  sub was/v
  sbj night/n
  nmod the/d
  vc falling/v
	root happened/v
	  nsubj accident/n
	  det the/d
	  advcl falling/v
	  aux was/v
	  nsubj night/n
	  det the/d
	  mark as/prep

Comme on peut le constater, il y a des différences majeures. Par exemple, la relation syntaxique « `nsubj` » de Stanford est appelée « `sbj` » dans CoNLL. Toutefois, ces différences de nomenclature étaient prévisibles puisque nous utilisons un analyseur syntaxique différent. De plus, la granularité des types de relation entre la nomenclature de CoNLL et Stanford est légèrement différente. Par exemple, un adjectif et un déterminant possèdent le même type de relation syntaxique dans le format CoNLL, c'est-à-dire un « *nmod* ». Alors qu'avec Stanford, les déterminants et les adjectifs possèdent leur propre type de relation syntaxique (« *det*» et « *amod*» respectivement). Ces différences compliquent grandement la comparaison de ces deux analyseurs. Toutefois, nous nous attarderons sur ce problème au moment du traitement des patrons dans la section 3.1.3.

Une autre différence importante se trouve au niveau de la structure de l'arbre. On peut constater que la racine (« `root` ») est la même, mais lorsqu'on descend dans les arbres, des différences commencent à apparaître. Par exemple, dans l'analyse de Stanford, « `falling` » est la tête de trois mots, soit : « `as` », « `was` » et « `night` ». Dans la version CoNLL, c'est la préposition « `as` » qui est la tête et seul le verbe « `was` » est son dépendant. Cette différence de structure rend l'identification de patrons difficile.

Ce genre de différence, au niveau de la structure, se produit parce que les représentations choisies par les analyseurs diffèrent dans certains cas spécifiques. Il est possible d'identifier ces situations et de faire un traitement en conséquence. Après avoir examiné un grand nombre de phrases, nous avons pu faire ressortir cinq situations dans lesquelles ce genre de problèmes survient.

- Utilisation des auxiliaires
- Utilisation de verbe à l'infinitif (to dance, etc.)
- Utilisation d'un verbe et un subordonné de conjonction (sub)
- Utilisation des compléments prédicatifs (prd)
- Utilisation du possessif.

Pour chacune de ces situations, nous avons développé des règles de transformation pour corriger la structure de l'arbre en conséquence[10]. Par exemple, voici les représentations syntaxiques de la phrase « *He loved to go to school* » qui contient un verbe à l'infinitif *(« to go* »):

Tableau 3.3: Exemple de différence de traitement entre CoNLL et Stanford pour certaines situations

Stanford	CoNLL avant la transformation	CoNLL après la transformation
root loved/v **xcomp** go/v nsubj he/prp **aux** to/prep prep to/prep pobj school/n nsubj he/prp	root loved/v sbj he/prp **oprd** to/prep **im** go/v dir to/prep pmod school/n	root loved/v **oprd** go/v sbj he/prp **aux** to/prep dir to/prep pmod school/n sbj he/prp/v

De manière générale, ces transformations sont dues à l'inversion de l'attribution de la relation entre deux mots. Comme nous pouvons le constater dans l'exemple précédent, il y a certaines situations (auxiliaires, verbe à l'infinitif, etc.) où la relation syntaxique n'est pas associée aux mêmes mots dans les deux formats. En prenant le verbe à l'infinitif « *to go* », avec Stanford « to » est un « aux» (auxiliaire) et « *go* » possède la relation qui lie le verbe au reste de la phrase (dans ce cas-ci un « *xcomp* »). Cependant, avec CoNLL, c'est l'inverse, « *go* » est un « *im* » (« *infinitive marker* ») et c'est à « *to* » que la relation est attachée (« *oprd* »). C'est ce genre d'inversion qui est corrigé par les règles de transformation.

En revenant à l'exemple de la phrase « *The accident happened as the night was falling* », si nous appliquons ces règles de transformation, on constate que nous obtenons la structure visée,

[10] Voir dans l'annexe 1 des exemples concrets, incluant le code en Prolog.

c'est-à-dire que la structure du nouvel arbre est très similaire à la structure obtenue avec Stanford. Voici l'arbre, après la modification, juxtaposé à l'arbre généré à partir de l'analyse de Stanford :

Tableau 3.4: Comparaison entre les arbres canoniques générés à partir du format de CoNLL et de Stanford après les ajustements.

Nouvel arbre généré à partir du format CoNLL	Arbre généré à partir de l'analyse de Stanford

root happened/v
  sbj accident/n
  nmod the/d
  tmp falling/v
  sbj night/n
  nmod the/d
  aux was/v
  complm as/prep
	root happened/v
	  nsubj accident/n
	  det the/d
	  advcl falling/v
	  nsubj night/n
	  det the/d
	  aux was/v
	  mark as/prep

3.1.2 La modification des règles de transformation

Une fois que les arbres canoniques ont été générés, il reste le problème de la nomenclature. Dans l'optique de limiter les modifications d'Anasem, nous voulions appliquer les règles de transformation qui étaient utilisées initialement (Sections 2.2.2). Ces transformations ont pour but de faciliter l'identification des patrons sémantiques.

Nous avons dû modifier neuf règles sur les douze règles de la section 2.2.2. Dans la plupart des cas, les modifications se sont limitées aux changements des termes pour que la nouvelle nomenclature puisse être comprise. C'est le cas pour les règles concernant les conjonctions de coordination, le compléments propositionnel avec un sujet externe (« xcomp ») ou les compléments prépositionnels (« pcomp»). Par contre, un ajustement intéressant a été nécessaire pour le traitement de la négation. En effet, dans le format CoNLL, la négation n'a pas d'identifiant particulier lors de l'analyse syntaxique, contrairement à ce que fait l'analyseur de Stanford.

Il a donc été nécessaire d'ajouter des mécanismes de reconnaissance de la négation afin de d'identifier ce concept dans l'analyse CoNLL. Dans ces cas, l'utilisation de la catégorie grammaticale et du mot « `not` » a permis de le faire. Voici un exemple de cette situation avec la phrase « *The cat does not sleep at home* » :

Tableau 3.5: Comparaison du traitement de la négation à différentes étapes de la transformation

Représentation visée (Stanford)	Représentation extraite du tableau en format CoNLL	Représentation après les transformations
root not/rb nsubj cat/n det the/d neg sleep/v aux does/v prep at/prep pobj home/n	root does/v adv not/rb sbj cat/n nmod the/dt vc sleep/v loc at/in pmod home/n	root not/rb sbj cat/n nmod the/d neg sleep/v aux does/v loc at/prep pmod home/n

Comme on peut le constater, une fois que les règles de transformations ont été appliquées, la représentation visée, c'est-à-dire celle que nous obtenons avec l'analyseur de Stanford, est très semblable au résultat. On peut également remarquer que le concept de négation est présent, ce qui permet aux autres règles et aux patrons de l'identifier dans les étapes suivantes.

3.1.3 La modification des patrons

Une fois que l'arbre a été généré et qu'il a subi les modifications nécessaires afin de le ramener à une structure connue, nous passons à une des étapes les plus importantes, soit l'identification des patrons sémantiques. L'objectif de toutes les étapes précédentes consistait à obtenir des arbres canoniques similaires aux arbres originaux (version non-transformée de l'analyseur) pour pouvoir utiliser nos patrons originaux en ne les modifiant que pour s'adapter à la nouvelle nomenclature.

Tel qu'attendu, dans la grande majorité des cas, les patrons n'ont dû subir que des modifications mineures. Ces modifications sont essentiellement reliées au fait que nous utilisons un ensemble différent de relations grammaticales. Il faut donc établir un appariement entre la terminologie utilisée par Stanford et celle de CoNLL. Pour ce faire, nous avons pris une partie du corpus de CoNLL et l'avons fait analyser syntaxiquement par l'analyseur de Stanford. Par la suite, nous avons fait des appariements entre les deux terminologies en comparant directement les relations et en compilant des statistiques pour chaque relation. Nous avons obtenu une table de conversion pour tous les types de relations. La table suivante en présente un extrait (la version intégrale est disponible dans l'annexe 3):

Tableau 3.6: Extrait de la table de conversion

Stanford	CoNLL
acomp	PRD
advcl	SUB
advmod	ADV
agent	PMOD
amod	NMOD
appos	APPO
attr	TMP
aux	OPRD
auxpass	VC
cc	DEP
ccomp	VC
complm	OBJ
conj	CONJ
det	NMOD
dobj	OBJ
expl	SBJ
infmod	IM
iobj	OBJ
mark	ADV
measure	AMOD
neg	ADV
nn	NMOD
nsubj	SBJ

Stanford et CoNLL utilisent des analyses différentes qui ne possèdent pas une nomenclature bijective, c'est-à-dire qu'il n'est pas possible de relier chaque terme de CoNLL à un et un seul terme de Stanford et vice versa. Donc, dans certains cas, notre table de conversion utilise l'appariement le plus fort. Prenons par exemple la relation « *det* ». Admettons que dans 90 % des cas, lorsque cette relation apparaît dans une phrase analysée par Stanford, son équivalent dans la phrase analysée par CoNLL est « *nmod* ». Dans ce cas, notre table de conversion suggère de remplacer les relations « *det* » par des relations « *nmod* ». Cela, même si dans 10 % des cas, c'est une autre relation. (Les chiffres de cet exemple sont fictifs, ils servent seulement à la compréhension)

Cette table de conversion nous sert de guide pour adapter les patrons à la nouvelle nomenclature. Toutefois, puisque la table de comparaison a été générée sans utiliser les règles de transformation, il y a certaines conversions auxquelles on ne peut se fier. Par exemple, la table de

conversion[11] nous propose de convertir les « *oprd* » en « aux» et les « *im* » en « *xcomp* », alors qu'en fait, ce que nous voulons faire c'est : « *oprd* » en « *xcomp* » et « *im* » en « *aux*» (voir

Tableau 3.3 pour l'exemple).

Chaque patron passe à travers une série d'étapes de transformation plus ou moins complexes. Prenons d'abord un patron simple, soit l'identification d'un sujet et de son complément d'objet direct. Selon la table de conversion, l'analyseur de Stanford identifie un sujet comme étant un « *nsubj* » et un complément direct comme « *dobj* ». Leur équivalent en format CoNLL est « *sbj* » et « *obj* ». En remplaçant les noms des relations, la règle originale fonctionne avec les nouveaux termes. Par contre, le cas des adjectifs modificateurs (« *amod* ») est un peu plus complexe. D'abord, un « *amod*» dans CoNLL est un « *advmod* » avec Stanford et les « *amod* » dans Stanford font partie des « *nmod* » dans CoNLL (il y a aussi d'autres catégories qui font partie de « *nmod* », par exemple les « *det*»). En fait, pour ce type de relation, le nom de la catégorie dans CoNLL vient de ce qui est modifié : « *amod* », modificateur d'adverbe ou d'adjectif et « *nmod*», modificateur de nom. Pour les départager, on peut donc se fier à la catégorie grammaticale.

Après avoir fait la traduction des relations, il est parfois nécessaire de faire des corrections causées par la différence de granularité entre les deux systèmes. Comme mentionné précédemment, « the » qui est un déterminant est classifié comme « *det* » par Stanford alors qu'il est un « *nmod* » pour CoNLL. La table de conversion nous propose de transformer « *det* » en « *nmod* ». Par contre, cette simple transformation causerait des problèmes puisque les adjectifs qui modifient les noms sont aussi des « *nmod* ». L'ajout d'une étape supplémentaire est nécessaire afin de départager ce genre de relations. Pour ce faire, nous utilisons la catégorie grammaticale (PoS) pour les différencier. La PoS d'un déterminant est représenté par « *d* » alors que pour un adjectif c'est « *jj* ». Grâce à ce procédé de différentiation, que nous ajoutons au patron responsable du traitement des « *nmod* », il est possible de différencier les déterminants des autres « *nmod* », malgré l'ambigüité de la relation syntaxique.

[11] Voire la table complète à l'annexe 3

Dans certains cas, des règles deviennent désuètes puisqu'elles couvrent des structures qui n'existent pas dans CoNLL. Un bon exemple de ce phénomène est le « *clause complement* » (« *ccomp*») de Stanford. Ce type de compléments désigne le dépendant d'un verbe qui agit comme un objet ou un adjectif. La règle responsable de ce type de complément n'est pas utilisée dans la version adaptée des patrons. Par exemple, prenons la phrase : « *Tom says that Mia likes to swim* » et regardons l'arbre syntaxique obtenu après l'application des règles de transformation.

Tableau 3.7: Exemple de perte de spécificité au niveau de CoNLL

Stanford après transformations	CoNLL après transformations
root says/v nsubj tom/nnp **ccomp** likes/v xcomp swim/v nsubj mia/nnp aux to/prep complm that/prep nsubj mia/nnp	root says/v sbj tom/nnp **obj** likes/v oprd swim/v sbj mia/nnp aux to/prep complm that/prep sbj mia/nnp

On constate que « *likes* » est un « *ccomp* » pour Stanford, mais simplement un « *obj* » pour CoNLL. Par conséquent, « *likes* » est traité comme un simple objet. Cependant, cela signifie qu'il y a une perte de spécificité au niveau des résultats.

Afin d'avoir une couverture maximale, nous avons adapté les 26 patrons principaux et les 16 patrons modificateurs en utilisant les méthodes mentionnées précédemment. Il est maintenant possible de prendre les phrases de la campagne d'évaluation de CoNLL 2008 et de les faire analyser par Anasem. Ce dernier utilise la nomenclature grammaticale de CoNLL pour baser son analyse, ce qui permet de faire une comparaison avec les résultats du « gold standard ». En effet, puisque les systèmes comparés ont accès aux mêmes données en entrée, une bonne partie des problèmes de nomenclature sont réglés. Toutefois, tout ce procédé n'affecte pas la forme de la sortie, c'est-à-dire que nous obtenons toujours une DRS comme représentation de l'analyse.

3.2 Méthodologie d'évaluation

Puisque nous avons décidé de faire une comparaison manuelle, il était très important d'établir une méthode claire et simple pour éviter qu'il se glisse des erreurs ou un biais lors de la comparaison. Une évaluation manuelle signifie également qu'il est pratiquement impossible de

comparer toutes les phrases disponibles dans le corpus. Le temps nécessaire pour le faire serait trop long. C'est pourquoi nous avons aussi défini une méthode pour la sélection des phrases qui serait utilisée pour l'évaluation (Section 3.2.1). De plus, puisqu'Anasem fait une analyse qui va plus loin que l'assignation des rôles sémantiques, il nous est impossible de calculer une précision, représentative des performances d'Anasem, directement sur les résultats obtenus en utilisant le « Gold Standard » de CoNLL. Toutefois, la précision étant une mesure importante, nous avons dû trouver un autre moyen de la calculer (Section 3.2.3).

3.2.1 Méthode de sélection des phrases d'évaluation

Avant de choisir les phrases à analyser, nous avons établi une série de contraintes auxquelles les phrases devaient se conformer pour pouvoir être choisies. Rappelons ici que les patrons définis dans Anasem sont essentiellement destinés à traiter des phrases déclaratives. Il n'est donc pas pertinent d'analyser des phrases non déclaratives puisqu'il n'y a pas de patron approprié pour ces cas. De plus, certains types de caractères causent des exceptions dans Anasem, qui est toujours en développement. Pour y remédier, nous avons établi une liste de contraintes :

- Une phrase ne doit pas contenir un ou plusieurs caractères parmi les suivants : " - `&$%()_:\/.
- Une phrase ne doit pas contenir de citation ou de dialogue ("«»")
- Une phrase ne peut contenir de mots avec un trait d'union.
- Une phrase doit avoir un minimum de 5 mots.
- Une phrase doit avoir un maximum de 30 mots.
- Une phrase doit contenir au moins un verbe.

Le rôle de la première contrainte est de retirer les phrases qui causeraient une interruption prématurée de l'analyse. Pour ce qui est de la deuxième règle, elle retire les citations et les dialogues (souvent sous la forme « ".." said Mr. X») parce que ce genre de phrases n'est pas traité par Anasem. Elles consistent généralement en une phrase dans une autre ou encore en une interjection.

La troisième règle a pour but d'éviter une particularité de CoNLL. En effet, lorsqu'il y a mot composé avec un trait d'union, le tableau représentant la phrase (section 1.4) est modifié afin

d'y ajouter de nouvelles lignes. Cet ajout permet de traiter les mots du mot composé de manière séparée. Puisque ce n'est qu'une particularité qui complexifie le processus d'analyse, sans pour autant lui apporter quelque avantage, nous avons préféré écarter ce type de mots de notre comparaison. Rappelons qu'Anasem traite les mots composés avec un trait d'union comme un seul mot.

Les dernières règles servent de filtre pour s'assurer le plus possible d'avoir des phrases déclaratives et non pas des interjections ou des interrogations. De plus, nous voulons garder le nombre de mots dans un écart raisonnable (entre 5 et 30 mots) afin d'éviter les phrases trop longues (extrêmement complexes à analyser) et les phrases trop courtes. Par exemple, la phrase « *At law school, the same* » a été écartée de la comparaison.

Une fois ces règles définies, nous avons pris une section au hasard dans le corpus, dans la partie « *development* », et nous avons extrait les phrases qui correspondaient à ces critères. Ce procédé nous a permis d'obtenir environ 400 phrases distinctes. Ce nombre était trop élevé pour une comparaison manuelle, donc nous avons dû faire une sélection parmi celles-ci. Pour ce faire, nous avons sélectionné 51 phrases au hasard[12]. Par la suite, nous avons reproduit le procédé, mais cette fois-ci avec la section « test » (avec le « gold standard») du corpus pour en extraire 50 autres phrases, pour un total de 101 phrases[13]. Ces cinquante dernières phrases ont été choisies dans le but de pouvoir les utiliser pour faire une comparaison avec le système de Johansson, LTH (Johansson & Nugues, 2008b).

3.2.2 Méthode de comparaison des résultats

Afin de comparer manuellement les sorties des analyseurs, il est nécessaire de limiter les choix des personnes responsables de la comparaison. De plus, il est important de réduire au maximum les biais et les erreurs possibles qui pourraient se produire lors de cette comparaison.

Tout d'abord, la comparaison s'est faite en deux étapes distinctes, soit en premier lieu avec les 51 phrases de la section de développement (le « *development* »), ce qui nous a permis de développer la méthode de comparaison, et en second lieu avec les 50 phrases de la section de test

[12] À l'aide de www.random.org

[13] La liste complète des phrases est disponible dans l'annexe 2

(le « *test* »). Toutes les comparaisons ont été faites par rapport au « Gold Standard ». Comme mentionné précédemment, malgré nos efforts d'adaptation, il y a toujours beaucoup de différences entre le format de CoNLL et la DRS. C'est d'ailleurs la raison pour laquelle nous avons procédé à une comparaison manuelle plutôt qu'automatique. De plus, ces différences rendent la mesure de la précision impossible à calculer. Nous avons donc concentré la comparaison sur la mesure du rappel dans un premier temps et nous avons fait une seconde expérimentation afin d'obtenir la précision. Celle-ci sera expliquée dans la section 3.2.3.

La méthode de comparaison fonctionne de la façon suivante. Avant de commencer, il faut fournir à Anasem une version incomplète, c'est-à-dire en retirant les colonnes 11 et 12+ (PRED et ARG), de la table représentant la phrase en format CoNLL. Prenons par exemple la phrase « *Economists say an August rebound in permits for multifamily units signaled an increase in September starts, though activity remains fairly modest by historical standards.* » (Cette phrase est directement extraite du corpus dans la section du « *development* ». C'est une des phrases choisies pour la comparaison.). Voici la représentation complète de cette phrase en format CoNLL :

Tableau 3.8: Représentation CoNLL complète de la phrase utilisée pour l'exemple

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	Economist	economist	NNS	NNS	Economist	economist	NNS	2	SBJ	_	A0	_	_	_	_	_	_	_
2	say	say	VBP	VBP	say	say	VBP	0	ROOT	say.01	_	_	_	_	_	_	_	_
3	an	an	DT	DT	an	an	DT	5	NMOD	_	_	_	_	_	_	_	_	_
4	August	august	NNP	NNP	August	august	NNP	5	NMOD	_	_	AM-TMP	_	_	_	_	_	_
5	rebound	rebound	NN	NN	rebound	rebound	NN	11	SBJ	rebound.(	_	_	_	A0	_	_	_	_
6	in	in	IN	IN	in	in	IN	5	NMOD	_	_	A1	_	_	_	_	_	_
7	permits	permit	NNS	NNS	permits	permit	NNS	6	PMOD	permit.01	_	_	A0	_	_	_	_	_
8	for	for	IN	IN	for	for	IN	7	NMOD	_	_	_	A1	_	_	_	_	_
9	multifami	multifami	RB	JJ	multifami	multifami	JJ	10	NMOD	_	_	_	_	_	_	_	_	_
10	units	unit	NNS	NNS	units	unit	NNS	8	PMOD	_	_	_	_	_	_	_	_	_
11	signaled	signal	VBD	VBD	signaled	signal	VBD	2	OBJ	signal.01	A1	_	_	_	_	_	_	_
12	an	an	DT	DT	an	an	DT	13	NMOD	_	_	_	_	_	_	_	_	_
13	increase	increase	NN	NN	increase	increase	NN	11	OBJ	increase.(	_	_	_	A1	A2	_	_	_
14	in	in	IN	IN	in	in	IN	13	NMOD	_	_	_	_	_	A1	_	_	_
15	Septembe	septembe	NNP	NNP	Septembe	septembe	NNP	16	NMOD	_	_	_	_	_	_	AM-TMP	_	_
16	starts	start	NNS	NNS	starts	start	NNS	14	PMOD	start.02	_	_	_	_	_	A2	_	_
17	,	,	,	,	,	,	,	11	P	_	_	_	_	_	_	_	_	_
18	though	though	IN	IN	though	though	IN	11	ADV	_	_	_	_	AM-ADV	_	_	_	_
19	activity	activity	NN	NN	activity	activity	NN	20	SBJ	_	_	_	_	_	_	_	A1	A1
20	remains	remain	VBZ	VBZ	remains	remain	VBZ	18	SUB	remain.01	_	_	_	_	_	_	_	_
21	fairly	fairly	RB	RB	fairly	fairly	RB	22	AMOD	_	_	_	_	_	_	_	_	_
22	modest	modest	JJ	JJ	modest	modest	JJ	20	PRD	_	_	_	_	_	_	_	A3	_
23	by	by	IN	IN	by	by	IN	20	ADV	_	_	_	_	_	_	_	AM-ADV	_
24	historical	historical	JJ	JJ	historical	historical	JJ	25	NMOD	_	_	_	_	_	_	_	_	A0
25	standards	standard	NNS	NNS	standards	standard	NNS	23	PMOD	standard.(	_	_	_	_	_	_	_	A2
26	.	.	.	.	.	.	.	2	P	_	_	_	_	_	_	_	_	_

Ce que notre analyseur reçoit, c'est un sous-ensemble des colonnes de cette table, soit les colonnes 0, 1, 4, 8, 9. Elles correspondent respectivement à la position, le mot, la catégorie grammaticale, la tête syntaxique et la relation syntaxique (Tableau 3.9):

Tableau 3.9 : Informations reçues par Anasem ainsi que l'arbre syntaxique construit

0	1	4	8	9
1	Economist	NNS	2	SBJ
2	say	VBP	0	ROOT
3	an	DT	5	NMOD
4	August	NNP	5	NMOD
5	rebound	NN	11	SBJ
6	in	IN	5	NMOD
7	permits	NNS	6	PMOD
8	for	IN	7	NMOD
9	multifami	JJ	10	NMOD
10	units	NNS	8	PMOD
11	signaled	VBD	2	OBJ
12	an	DT	13	NMOD
13	increase	NN	11	OBJ
14	in	IN	13	NMOD
15	Septembe	NNP	16	NMOD
16	starts	NNS	14	PMOD
17	,	,	11	P
18	though	IN	11	ADV
19	activity	NN	20	SBJ
20	remains	VBZ	18	SUB
21	fairly	RB	22	AMOD
22	modest	JJ	20	PRD
23	by	IN	20	ADV
24	historical	JJ	25	NMOD
25	standards	NNS	23	PMOD
26	.	.	2	P

Arbre syntaxique

```
root say/v
  sbj economists/n
  obj signaled/v
    sbj rebound/n
      nmod an/d
      nmod august/nnp
      nmod in/prep
        pmod permits/n
          nmod for/prep
            pmod units/n
              nmod multifamily/jj
    obj increase/n
      nmod an/d
      nmod in/prep
        pmod starts/n
          nmod september/nnp
    adv modest/jj
      amod fairly/rb
      sbj activity/n
      adv by/prep
        pmod standards/n
          nmod historical/jj
      complm though/prep
      cop remains/v
```

Une fois l'analyse complétée à partir du texte original, Anasem crée une DRS qui représente l'analyse qu'il a faite de la phrase. Pour notre exemple, la DRS correspondant à la phrase est la suivante :

Tableau 3.10: La DRS représentant la phrase de l'exemple avec les termes de l'analyseur de Stanford

```
---------------------------------
[id1,id2,id3,id4,id5,id6,e1,e2]
---------------------------------
entity(id1,economists)
entity(id2,rebound)
entity(id3,permits)
entity(id4,units)
attribute(id4,multifamily)
nmod(id3,id4,for)
nmod(id2,id3,in)
attribute(id2,august)
entity(id5,increase)
entity(id6,starts)
attribute(id6,september)
nmod(id5,id6,in)
event(e1,signaled,id2,id5)
amAdv(e1,modest)
event(e2,say,id1,e1)
---------------------------------
```

Une fois que nous possédons la DRS (sortie d'Anasem) et la version complète du tableau (sortie provenant du Gold Standard CoNLL), il faut d'abord repérer tous les prédicats identifiés par le « Gold Standard » et tous les types d'arguments présents dans la section sémantique du tableau. Ensuite, il faut identifier les prédicats détectés par Anasem. Pour ce faire, on regarde dans la DRS afin de voir si ces prédicats sont présents sous la forme d'entités (entity(id1,*economists*)) ou d'évènements (event(e1,*signaled*,...,...)). Dans l'exemple, deux des huit prédicats sont absents de la DRS, soit « *Remain* » et « *Standard* ».

Finalement, on relie les arguments de la section sémantique du tableau de CoNLL avec les arguments de la DRS. Ces arguments peuvent prendre deux formes générales dans la DRS : reliés aux événements (ils apparaissent sur la ligne d'un événement) ou comme complément seul. La première forme est souvent associée aux sujets et aux compléments d'objets directs qui sont liés à l'évènement. Par exemple, « event(e1,signaled,id2,id5) » possède deux arguments, id2 et id5. On constate que ces identifiants font référence à « *Rebound* » et à « *Increase* ». La seconde forme est associée aux autres types de compléments (temporels, location, etc.). Par exemple, « nmod(id2,id3,in) » indique qu'il y a un argument qui relie l'entité « id2 » à l'entité « id3 » et que la relation est « *in* ».

Il ne reste qu'à prendre les prédicats un par un et vérifier si l'on peut faire corréler les arguments du « Gold Standard » avec les arguments de la DRS. Puisque nous considérons les arguments sans étiquettes, le type d'arguments dans la DRS n'a pas besoin d'être le même, le fait de l'avoir identifié suffit. Par exemple, « *august* » est un argument de type temporel, mais il est identifié dans la DRS comme un attribut de « *rebound* », nous considérons donc que l'argument est correctement détecté. Nous utilisons la colonne « commentaire » pour identifier des cas particuliers. Par exemple, nous prenons en note les « *self arguments* », c'est-à-dire lorsque l'un des arguments d'un prédicat est son propre prédicat. On peut voir un exemple de ce concept dans la phrase d'exemple avec les prédicats suivants: « *Permit* », « *Increase* », « *Start* » et « *Remain* ». Ce phénomène n'étant pas traité dans Anasem, ils ne sont donc jamais identifiés. Voici le tableau obtenu après ce procédé de comparaison :

Tableau 3.11: Tableau de comparaison de la phrase d'exemple analysée par l'analyseur et le « gold standard »

Prédicats	Détecté	A0	A1	A2	A3	AM-TMP	AM-ADV	commentaire
Say	1	1	1					
Rebound	1		1			1		
Permit	1	0	1					Le A0 est un « self argument ».
Signal	1	1	1				1	
Increase	1		1	0				Le A2 est un « self argument ».
Start	1			0		1		Le A2 est un « self argument ».
Remain	0		0		0		0	
Standard	0	0		0				Le A2 est un « self argument ».

Donc, pour cette phrase, nous avons identifié cinq prédicats parmi les sept possibles et dix arguments parmi les dix-huit possibles. Par contre, parmi ces 18 arguments, quatre arguments sont des « *self arguments* ». Le procédé est répété pour toutes les autres phrases. Une fois l'ensemble des résultats compilés, on calcule le rappel. Pour ce faire, on applique la formule (section 1.5) qui consiste à faire la somme de tous les arguments (et/ou les prédicats) qui ont été détectés et la diviser par le nombre total d'arguments (et/ou de prédicats).

Il est important de noter qu'il existe un autre type d'arguments, en plus des « *self arguments* », qui n'est pas traité par Anasem. Cet argument, le SU, représente les « *Support Chain* » qui sont propres à NomBank (voir Tableau 1.2). C'est pourquoi nous calculons les résultats avec et sans cet argument.

3.2.3 Méthode d'évaluation de la précision

Comme l'évaluation est faite exclusivement sur le rappel, nous n'avons aucune mesure de précision. Il serait donc fort pertinent de trouver un moyen d'avoir une estimation de la précision d'Anasem sur le corpus de CoNLL. Un des problèmes majeurs est qu'Anasem couvre des relations sémantiques qui ne figurent pas dans CoNLL. Par exemple, dans la section de phrases « ... his sweaty armpits... », le « Gold Standard » indique qu'« *armpits* » ne possède qu'un seul attribut, « *his* ». Anasem identifie aussi « *sweaty* » comme attribut, ce qui est tout à fait correct. Voici une autre situation dans laquelle le « Gold Standard » diffère en raison de sa structure « prédicat-argument » : seuls les mots possédant des arguments sont considérés comme des prédicats. Par exemple, dans la phrase « *The man came often to the house* », il n'y a qu'un prédicat (« *come*»), comme on peut le constater dans le tableau ci-dessous.

Tableau 3.12: Exemple de l'absence d'un prédicat

1	The	_	_	DT	The	_	DT	2	NMOD	_	_
2	man	man	_	NN	man	man	NN	3	SBJ	_	A1
3	came	come	_	VBD	came	come	VBD	0	ROOT	come.01	_
4	often	_	_	RB	often	_	RB	3	TMP	_	AM-TMP
5	to	_	_	TO	to	_	TO	3	DIR	_	A4
6	the	_	_	DT	the	_	DT	7	NMOD	_	_
7	house	house	_	NN	house	house	NN	5	PMOD	_	_
8	.	_	_	.	.	_	.	3	P	_	_

Par contre, si l'on ajoute quelques mots (« of Commons») pour donner des arguments à « house », voici ce que nous obtenons.

Tableau 3.13: Exemple de l'apparition d'un prédicat avec son argument

1	The	_	_	DT	The	_	DT	2	NMOD	_	_	_
2	man	man	_	NN	man	man	NN	3	SBJ	_	A1	_
3	came	come	_	VBD	came	come	VBD	0	ROOT	come.01	_	_
4	often	_	_	RB	often	_	RB	3	TMP	_	AM-TMP	_
5	to	_	_	TO	to	_	TO	3	DIR	_	A4	_
6	the	_	_	DT	the	_	DT	7	NMOD	_	_	_
7	house	house	_	NN	house	house	NN	5	PMOD	house.03	_	_
8	of	_	_	IN	of	_	IN	7	NMOD	_	_	A1
9	commons	commons	_	NN	commons	commons	NN	8	PMOD	_	_	_
10	.	_	_	.	.	_	.	3	P	_	_	_

On constate que dans la seconde phrase, il y a un mot qui est devenu un prédicat, alors qu'il ne l'était pas dans la première phrase (« *house* »). Contrairement à CoNLL, Anasem considère « *house* » comme un prédicat dans les deux phrases. Voici les DRS représentant les deux versions de la phrase :

The doctor came often to the house	The doctor came often to the house of commons
----------	----------
[id1,e1,id2,id3]	[id1,e1,id2,id3]
----------	----------
resolve(id1)	resolve(id1)
entity(id1,man)	entity(id1,man)
event(e1,came,id1)	event(e1,came,id1)
resolve(id2)	resolve(id2)
entity(id2,house)	entity(id2,house)
to(e1,id2)	entity(id3,commons)
manner(e1,often)	of(id2,id3)
amTmp(e1,id3)	to(e1,id2)
----------	manner(e1,often)

Si on devait calculer la précision d'Anasem directement par rapport au « Gold Standard », notre outil serait défavorisé puisqu'il identifie toujours tous les prédicats, alors que, dans CoNLL, seuls les prédicats qui possèdent des arguments sont identifiés.

Pour remédier à cette situation, nous avons décidé de faire appel à trois spécialistes afin d'obtenir une évaluation de la précision. Il faut noter que les spécialistes choisis sont des personnes qui ont déjà travaillé sur Anasem. Pour ce faire, nous leur avons fourni les 50 phrases de la section de « *test* », les 50 DRS correspondantes, ainsi que la méthodologie de comparaison présentée ci-dessus. Par la suite, nous leur avons demandé de vérifier la précision des résultats obtenus. Pour ce faire, ils devaient déterminer si chaque élément de la DRS représentait effectivement une vraie relation. Une fois leur évaluation individuelle effectuée, les experts ont comparé leurs résultats. En cas de conflit, une discussion était effectuée jusqu'à l'obtention d'un consensus.

Prenons par exemple la phrase suivante avec sa DRS : « *Mr McKinley examined everything with critical care, seeking something material to blame for his son's illness*. »

```
---------------------------------
[id1,id2,id3,e1,id4,id5,id6,e2,e3,id7]
---------------------------------
namedentity(id1,mckinley)
entity(id2,everything)
entity(id3,something-material)
event(e1,blame,id1)
entity(id4,illness)
entity(id5,son)
of(id5,id6)
entity(id6,him)
of(id4,id5)
for(e1,id4)
event(e2,seeking,id1,id3)
event(e3,examined,id1,id2)
entity(id7,care)
attribute(id7,critical)
with(e3,id7)
---------------------------------
```

Cette phrase comprend 10 prédicats (« *entity* » et « *event* ») auxquels sont rattachés 10 arguments (« *of* », « *for* », « *attribute* », « *with* » et les arguments des « *event* »). Dans ce cas particulier, les trois experts ont déterminé que tous les éléments étaient correctement représentés.

Ce même procédé a été effectué sur l'ensemble des phrases de tests. Les résultats obtenus pour ces phrases sont dans le tableau ci-dessous en compagnie des résultats obtenus par LTH pour les mêmes données, mais évalués par l'outil d'évaluation de CoNLL. Les prédicats et les arguments ont été d'abord mesurés séparément puis combinés.

Tableau 3.14: Précision des résultats d'Anasem et de LTH sur la section de « *test* »

	Anasem	LTH
Prédicats	92 %	79 %
Arguments	80 %	84 %
Prédicats et arguments	86 %	81 %

Il est important de se rappeler que cette précision a pour but de donner une idée générale de ce à quoi on peut s'attendre. Elle représente la précision d'un sous-ensemble des phrases. On peut voir qu'elle est très bonne lorsqu'il s'agit de prédicats et qu'elle est relativement bonne pour les arguments. Ces valeurs nous permettent de mieux interpréter les résultats obtenus lors de la comparaison.

CHAPITRE 4 PRÉSENTATION DES RÉSULTATS

Dans ce chapitre nous présentons l'ensemble des résultats obtenus lors de la comparaison d'Anasem, après l'adaptation, à la fois avec le « Gold Standard » et avec LTH. Dans un premier temps, nous expliquons la méthode de représentation utilisée pour comparer les résultats. Ensuite, nous présentons les résultats globaux des 101 phrases comparées et nous nous concentrons sur un sous-ensemble des résultats, appelés « les analyses des phrases entièrement couvertes », ainsi que ce qu'il représente. Nous poursuivons avec les résultats obtenus en utilisant une classification basée sur la reconnaissance des arguments. Finalement, nous comparons les résultats obtenus par Johansson (LTH) en utilisant le même sous-ensemble de phrases du corpus que celui utilisé par Anasem.

4.1 Représentation des résultats

Avant d'analyser les résultats obtenus, il faut comprendre la façon utilisée pour les représenter. Tout d'abord, il faut se rappeler que nous avons choisi les phrases à comparer dans deux sections différentes du corpus. La séparation des résultats est importante, car nous ne pouvons utiliser ces deux sections pour la comparaison avec LTH. En effet, l'analyseur de Johansson utilise la section de développement lors de son apprentissage, et donc les résultats obtenus avec cette section seraient biaisés. Par contre, puisqu'Anasem n'utilise pas de méthode d'apprentissage, les deux sections ont la même valeur, en autant que la comparaison se fait uniquement avec le « Gold Standard ». C'est pourquoi nous nous basons sur la combinaison des deux sections lors de la présentation des résultats, mais nous utilisons seulement la section de « *test* » lorsqu'on compare Anasem avec LTH.

Comme mentionné précédemment, nous avons regroupé les résultats de deux façons; une, centrée sur les phrases, et une autre, centrée sur la reconnaissance des arguments indépendamment des phrases.

4.1.1 Analyse basée sur les phrases

Tout d'abord, nous avons séparé les phrases en fonction de leur provenance dans le corpus (« *development* » ou « *test* »). Ensuite, pour chaque phrase analysée, nous avons pris le tableau de comparaison correspondant, présenté à la section 3.2.2 (

Tableau 3.11), et nous avons effectué la somme des prédicats correctement identifiés sur le total de prédicats possibles (le « rappel », section 1.5) basé sur le « Gold Standard ». Nous avons utilisé le même procédé pour tous les types d'arguments. Nous avons aussi calculé le nombre total d'arguments marqué comme « *self argument* » pour les deux sections du corpus. Nous avons donc obtenu, pour chaque phrase, le taux d'identification pour les prédicats et les arguments. Ces résultats, nommés «l'ensemble des phrases » se retrouvent à la section 0.

Nous avons constaté que, lorsqu'il manque un ou plusieurs prédicats, cela indique que la phrase a été partiellement analysée par Anasem. En fait, l'omission d'un prédicat entraîne l'omission de tous les arguments qui y sont liés. Ce problème peut être causé par plusieurs raisons, qui seront couvertes dans le chapitre 5. Dans le but de bien comprendre l'impact de ces phrases sur l'ensemble des résultats, nous avons recalculé les résultats sans tenir compte de ces phrases pour obtenir un deuxième ensemble de résultats. Ces résultats, nommés « analyse des phrases entièrement couvertes » se retrouvent à la section 4.2.1.

4.1.2 Analyse basée sur la reconnaissance d'arguments

Nous avons aussi calculé les résultats basés sur la reconnaissance des arguments. Nous voulions avoir les résultats sur tous les prédicats indépendamment des phrases. Pour ce faire, nous avons pris en compte tous les prédicats correctement identifiés par Anasem, et ce, indépendamment des phrases. Nous avons ensuite procédé de la même façon que lors de l'analyse basée sur les phrases pour calculer le taux d'identification des arguments de ces prédicats.

Puisque les résultats obtenus lors de l'analyse basée sur les phrases sont affectés négativement par les phrases partiellement analysées, nous avons voulu voir si le même effet se produisait lorsqu'on se concentre sur les prédicats. Nous avons donc recalculé les résultats, mais cette fois-ci, en ne comptant que les arguments des prédicats se trouvant dans des phrases entièrement couvertes.

4.2 Résultats centrés sur les phrases

Comme mentionné précédemment, les phrases testées proviennent de deux sections du corpus. Elles seront donc d'abord présentées séparément. Il est important de noter que tous les résultats obtenus sont des mesures de rappel, c'est-à-dire que les chiffres représentent le nombre

de prédicats (ou d'arguments) correctement identifiés sur le nombre maximal possible (nombre dans le "Gold Standard"). Nous présentons les résultats sur l'ensemble des phrases d'abord, puis nous présentons les résultats des phrases entièrement couvertes. Analyse de l'ensemble des phrases.

Résultats sur la section « *development* »

Parmi les 51 phrases de la section de développement du corpus, il y a 28 phrases entièrement couvertes, 18 phrases partiellement analysées et 5 phrases qui ne sont pas analysées. Ces dernières ont été retirées des résultats, car les erreurs se trouvent au niveau de l'analyse syntaxique et non au niveau de l'analyse sémantique. Elles sont explorées plus en détail à la section 4.6. En moyenne, un peu plus d'une phrase sur deux est entièrement couverte.

Dans l'ensemble des phrases analysées, il y a un total de 167 prédicats possiblement identifiables et, parmi ceux-ci, 135 sont identifiés par Anasem, ce qui correspond à un taux de 80.8 % de réussite. En ce qui concerne les arguments, sur un total de 378 possibles, 241 ont été identifiés, pour un taux de réussite de 63.8 %. Même si nous ne vérifions pas le type de l'argument, il est intéressant de voir la ventilation des résultats en fonction des types spécifiés dans le « Gold Standard ». Voici un tableau qui présente pour chaque type d'arguments, le nombre d'occurrences correctement identifiées sur le nombre total possible :

Tableau 4.1: Ventilation des résultats en fonction des arguments pour la section de développement

Types d'arguments	**A0**	**A1**	**A2**	**A3**	**A4**	**AM-CAU**	**AM-LOC**	**AM-TMP**	**AM-PNC**	**AM-EXT**	**AM-MNR**	**AM-ADV**
Identifiés	59	94	16	5	0	2	7	24	2	2	7	7
Total	87	135	36	9	1	2	11	33	2	2	10	11
Pourcentage	67.8	69.6	44.4	55.6	0	100	63.6	72.7	100	100	70	63.6

AM-DIS	**AM-MOD**	**AM-DIR**	**AM-NEG**	**SU**	**R-A0**	**R-A1**	**R-AM-MNR**	**C-A1**	**C-AM-MNR**
6	2	1	2	0	0	0	1	4	0
10	10	1	2	7	1	1	1	5	1
60	20	100	100	0	0	0	100	80	0

Comme on peut le constater, certains types d'arguments sont beaucoup moins présents que d'autres. Par conséquent, leurs résultats sont beaucoup moins significatifs. La majorité des arguments sont de type A0, A1, A2 et AM-TMP.

En retirant les SU, qui ne sont pas traités par Anasem, et les arguments qui ont été identifiés comme des « *self arguments* », le nombre total d'arguments diminue à 356, ce qui correspond à un taux d'identification de 67.7 % pour le rappel.

Résultats sur la section « *test* »

50 phrases ont été extraites de la section de tests du corpus. Parmi celles-ci, 25 phrases sont entièrement couvertes, 19 phrases sont partiellement analysées et 6 phrases ne sont pas analysées. Encore une fois, nous reviendrons sur ces dernières dans la section 4.6. Le ratio de phrases entièrement couvertes en fonction du total des phrases est relativement similaire à la section précédente.

Cette fois-ci, il y a 131 prédicats et 294 arguments dans le « Gold Standard ». Au total 94 prédicats (71.8 %) et 163 arguments (55.4 %) ont été correctement identifiés. On peut voir la ventilation de ces résultats en fonction des types d'arguments dans le tableau suivant :

Tableau 4.2: Ventilation des résultats en fonction des arguments pour la section de tests

Types d'arguments	**A0**	**A1**	**A2**	**A3**	**A4**	**AM-CAU**	**AM-LOC**	**AM-TMP**	**AM-PNC**	**AM-EXT**	**AM-MNR**	**AM-ADV**
Identifiés	48	55	9	0	2	3	8	5	1	0	11	11
Total	85	91	19	1	3	4	12	8	5	0	14	16
Pourcentage	56.5	60.4	47.4	0	66.7	75	66.7	62.5	20	-	78.6	68.8

AM-DIS	**AM-MOD**	**AM-DIR**	**AM-NEG**	**SU**	**R-A0**	**R-A1**	**R-AM-MNR**	**C-A1**	**C-AM-MNR**
1	0	3	4	0	1	0	0	1	0
2	7	7	8	6	2	0	0	4	0
50	0	42.9	50	0	50	-	-	25	-

Lorsqu'on compare la fréquence des types d'arguments, on constate que les A0, A1 et A2 sont encore très présents, mais que les AM-TMP sont moins communs. Pour ce qui est des autres types, leur fréquence est relativement similaire à celle de la section précédente.

En retirant les « *self arguments* » et les SU des arguments possibles, pour la même raison qu'antérieurement, nous obtenons une amélioration du rappel de 2.4 % soit 57.8 % de réussite en abaissant le nombre total des arguments à 282.

Résultats sur les sections combinées

Finalement, nous pouvons combiner les deux sections, puisque la comparaison est faite entre Anasem, qui ne se base pas sur des techniques d'apprentissage, et le « Gold Standard ». Cela nous donne un total de 101 phrases, parmi lesquelles 90 sont entièrement couvertes ou

partiellement analysées et 11 ne sont pas analysées. De plus, 53 phrases parmi les 90 sont entièrement couvertes.

Ces phrases contiennent 298 prédicats et 672 arguments parmi lesquels Anasem en identifie respectivement 229 et 404, résultant en un taux d'identification des prédicats de 76.8 % et d'identification des arguments de 60.1 %. Encore une fois, il est possible de voir la ventilation des arguments combinés en fonction de leur type dans le tableau suivant :

Tableau 4.3: Ventilation des résultats en fonction des arguments pour les sections combinées

Types d'arguments	**A0**	**A1**	**A2**	**A3**	**A4**	**AM-CAU**	**AM-LOC**	**AM-TMP**	**AM-PNC**	**AM-EXT**	**AM-MNR**	**AM-ADV**
Identifiés	107	149	25	5	2	5	15	29	3	2	18	18
Total	172	226	55	10	4	6	23	41	7	2	24	27
Pourcentage	62.2	65.9	45.5	50	50	83.3	65.2	70.7	42.9	100	75	66.7

AM-DIS	**AM-MOD**	**AM-DIR**	**AM-NEG**	**SU**	**R-A0**	**R-A1**	**R-AM-MNR**	**C-A1**	**C-AM-MNR**
7	2	4	6	0	1	0	1	5	0
12	17	8	10	13	3	1	1	9	1
58.3	11.8	50	60	0	33.3	0	100	55.6	0

On observe que les arguments de type A0, A1, A2 sont toujours très présents, mais on voit aussi que les AM-LOC, AM-TMP, AM-MNR et AM-ADV sont plus significatifs puisqu'ils sont présents dans les phrases analysées à plus de 20 reprises chacun.

Tout comme nous l'avons fait avec chacune des sections, il est possible d'avoir des résultats un peu plus élevés en retirant les « *self arguments* » et les SU qui ne sont pas traités par Anasem. Dans ce cas-ci, les arguments identifiables passent de 672 à 638 pour un taux d'identification de 63.3 % au lieu de 60.1 %.

4.2.1 Analyse des phrases entièrement couvertes

Puisque les phrases entièrement couvertes ne comprennent que les phrases dont tous les prédicats sont identifiés correctement, seul le taux d'identification des arguments est pertinent.

Résultats sur la section « *development* »

Pour la section de développement, il y a 28 phrases qui sont entièrement couvertes. Dans ces phrases, il y a 87 prédicats et 206 arguments dont 164 arguments sont identifiés. Le taux de réussite est 79.6 % de rappel, soit près de 19 % de plus qu'en utilisant toutes les phrases. On peut aussi regarder la ventilation des résultats.

Tableau 4.4: Ventilation des résultats en fonction des arguments pour les phrases complètes de la section de développement

Types d'arguments	A0	A1	A2	A3	A4	AM-CAU	AM-LOC	AM-TMP	AM-PNC	AM-EXT	AM-MNR
Identifiés	42	63	10	1	0	2	5	16	2	2	4
Total	54	72	13	2	1	2	6	17	2	2	4
Pourcentage	77.8	87.5	76.9	50	0	100	83.3	94.1	100	100	100

AM-ADV	AM-DIS	AM-MOD	AM-DIR	AM-NEG	SU	R-A0	R-A1	R-AM-MNR	C-A1	C-AM-MNR
6	4	2	1	1	0	0	0	1	2	0
6	7	8	1	1	4	1	0	1	2	0
100	57.1	25	100	100	0	0	-	100	1	-

De manière générale, la plupart des arguments qui sont présent a plus de 5 reprises ont un taux très élevé d'identification. Par contre, deux types d'arguments ont des résultats relativement bas, soit le AM-MOD et le AM-DIS, mais ils ne sont présents que 8 et 7 fois respectivement.

Si l'on traite les résultats tel qu'effectué précédemment, en retirant les SU et les « *self arguments* », nous obtenons un taux d'identification de 83.2 % avec 164 arguments identifiés sur un total de 197.

Résultats sur la section « test »

Pour ce qui est des phrases de la section de tests, il y a 25 phrases entièrement couvertes sur les 50. Dans ces 50 phrases, il y a 60 prédicats et 107 arguments correctement identifiés. Le taux d'identification est 77.5 %, ce qui est supérieur aux résultats précédant (55.4 %). Évidemment, lorsque les phrases sont entièrement couvertes, les résultats sont de meilleure qualité.

Tableau 4.5: Ventilation des résultats en fonction des arguments pour les phrases compléetement analysées de la section de tests

Types d'arguments	A0	A1	A2	A3	A4	AM-CAU	AM-LOC	AM-TMP	AM-PNC	AM-EXT	AM-MNR
Identifiés	31	37	7	0	1	2	6	3	0	0	6
Total	42	40	11	0	1	2	6	4	0	0	7
Pourcentage	73.8	92.5	63.6	-	100	100	100	75	-	-	85.7

AM-ADV	AM-DIS	AM-MOD	AM-DIR	AM-NEG	SU	R-A0	R-A1	R-AM-MNR	C-A1	C-AM-MNR
6	0	0	2	4	0	1	0	0	1	0
8	0	3	5	4	3	1	0	0	1	0
75	-	0	40	100	0	100	-	-	1	-

La distribution des arguments est très similaire à la section précédente; on retrouve moins d'AM-MOD que les autres types d'arguments. Par contre, il n'y a pas de AM-DIS dans cette section. Il y a aussi une légère diminution au niveau des A2, mais elle pourrait être due à la faible représentation de ce type d'arguments.

En retirant les « *self arguments* » et les SU, le total d'arguments passe de 138 à 130 soit une augmentation du taux de près de 5 % pour atteindre 82.3 %.

Résultats sur les sections combinées

Finalement, en combinant tous les résultats des phrases entièrement couvertes, cela nous permet d'obtenir des résultats plus représentatifs. Nous avons donc 101 phrases, dont 53 sont annotées comme « entièrement couvertes », qui contiennent 344 arguments dont 271 ont été identifiés par Anasem. Le taux de réussite d'identifications des arguments est donc 78.8 %. Comparé au 60.1 % que nous avions avec l'ensemble des phrases, ce résultat est beaucoup plus près de ce à quoi on pourrait s'attendre pour un analyseur de l'état de l'art. On peut regarder la répartition des résultats en fonction des arguments dans le tableau suivant :

Tableau 4.6: Répartition des résultats en fonction des arguments pour la combinaison des deux sections.

Types d'arguments	**A0**	**A1**	**A2**	**A3**	**A4**	**AM-CAU**	**AM-LOC**	**AM-TMP**	**AM-PNC**	**AM-EXT**	**AM-MNR**
Identifiés	73	100	17	1	1	4	11	19	2	2	10
Total	96	112	24	2	2	4	12	21	2	2	11
Pourcentage	76	89.3	70.8	50	50	100	91.7	90.5	100	100	90.9

AM-ADV	**AM-DIS**	**AM-MOD**	**AM-DIR**	**AM-NEG**	**SU**	**R-A0**	**R-A1**	**R-AM-MNR**	**C-A1**	**C-AM-MNR**
12	4	2	3	5	0	1	0	1	3	0
14	7	11	6	5	7	2	0	1	3	0
85.7	57.1	18.2	50	100	0	50	-	100	100	-

On remarque tout d'abord le haut taux de réussite d'identifications des arguments de types A1, AM-LOC, AM-TMP, AM-MNR et AM-ADV.Ttous se situent autour de 90 % avec au moins 11 cas différents. Le cas des A1 est particulièrement remarquable, puisque 100 des 112 arguments de ce type sont correctement identifiés. Par contre, on constate que les problèmes au niveau des AM-MOD persistent.

Lorsque les arguments non traités par Anasem, SU et « *self arguments* », sont retirés, le total des arguments descend à 327 pour obtenir un taux de réussite de 82.9 %.

4.3 Résultats basés sur la reconnaissance d'arguments

Après avoir obtenu les résultats de l'analyse en fonction des phrases, nous avons voulu connaitre l'impact des phrases partiellement analysées sur la reconnaissance des arguments. Pour ce faire, nous avons extrait tous les prédicats correctement identifiés dans la section de tests. L'objectif est de regarder la différence entre les taux de reconnaissance des arguments dans les phrases entièrement couvertes, et les taux de reconnaissance des arguments dans l'ensemble des phrases. De cette façon, il est possible de déterminer si une phrase partiellement analysée affecte l'identification des arguments.

4.3.1 Tous les prédicats détectés

D'abord, nous avons extrait les 94 prédicats et leurs arguments des 50 phrases de la section de tests. À ces prédicats sont associés 221 arguments de plusieurs types, dont 162 ont été correctement identifiés. On obtient un rappel de 73.3 %, ce qui situe les résultats entre ceux de l'analyse de base, qui comprend toutes les phrases, et ceux des phrases entièrement couvertes. En fait, ce résultat ignore les arguments associés aux prédicats qui ne sont pas identifiés. Ces résultats étaient anticipés, parce que les prédicats choisis sont tous correctement identifiés, mais certains se trouvent dans des phrases partiellement analysées. Grâce au tableau de distribution des arguments en fonction de leur type, il est possible d'avoir une vue plus détaillée sur ces résultats.

Tableau 4.7: Tableau de la distribution des résultats, basés sur les prédicats, en fonction des types d'arguments pour la section de test.

Types d'arguments	**A0**	**A1**	**A2**	**A3**	**A4**	**AM-CAU**	**AM-LOC**	**AM-TMP**	**AM-PNC**	**AM-EXT**	**AM-MNR**
Identifiés	48	55	9	0	2	3	7	5	1	0	11
Total	64	64	15	0	3	4	9	6	4	0	13
Pourcentage	75	85.9	60	-	66.7	75	77.8	83.3	25	-	84.6

AM-ADV	**AM-DIS**	**AM-MOD**	**AM-DIR**	**AM-NEG**	**SU**	**R-A0**	**R-A1**	**R-AM-MNR**	**C-A1**	**C-AM-MNR**
11	1	0	3	4	0	1	0	0	1	0
14	2	6	6	6	3	1	0	0	1	0
78.6	50	0	50	66.7	0	100	-	-	100	-

Tel qu'attendu, ces résultats sont relativement similaires aux résultats obtenus précédemment (Section 4.2).

4.3.2 Prédicats détectés dans les phrases entièrement couvertes

Les résultats obtenus dans cette section sont identiques aux résultats obtenus dans la section 4.2.1 pour la section « *test* » du corpus. En effet, dans toutes les phrases entièrement couvertes, tous les prédicats sont toujours détectés.

Le nombre de prédicats diminue à 60 et le nombre d'arguments passe de 221 à 138. Parmi ces arguments, Anasem a correctement identifié 107 arguments pour un rappel de 77.5 %. Malgré un écart de 4.2 % (77.5 % par rapport à 73.3 %), on peut en déduire que lorsqu'il y a présence de prédicats non identifiés dans une phrase, les résultats obtenus pour les arguments des prédicats identifiés de cette phrase sont inférieurs aux résultats obtenus pour les phrases entièrement couvertes. Voici le tableau représentant la ventilation des résultats en fonction des types d'arguments :

Tableau 4.8: Tableau de la distribution des résultats, basés sur les prédicats, en fonction des types d'arguments pour les phrases complètes de la section de test.

Types d'arguments	**A0**	**A1**	**A2**	**A3**	**A4**	**AM-CAU**	**AM-LOC**	**AM-TMP**	**AM-PNC**	**AM-EXT**	**AM-MNR**
Identifiés	31	37	7	0	1	2	6	3	0	0	6
Total	42	40	11	0	1	2	6	4	0	0	7
Pourcentage	73.8	92.5	63.6	-	100	100	100	75	-	-	85.7

AM-ADV	**AM-DIS**	**AM-MOD**	**AM-DIR**	**AM-NEG**	**SU**	**R-A0**	**R-A1**	**R-AM-MNR**	**C-A1**	**C-AM-MNR**
6	0	0	2	4	0	1	0	0	1	0
8	0	3	5	4	3	1	0	0	1	0
75	-	0	40	100	0	100	-	-	100	-

4.4 Retour sur les résultats

Somme toute, on constate qu'Anasem semble avoir plus de facilité pour identifier certains types d'arguments. Par exemple, on remarque que les plus hautes valeurs de rappel sont souvent parmi les types d'arguments suivants: A1, AM-LOC, AM-TMP, AM-MNR et AM-ADV. On constate aussi que les types d'arguments qui ont un faible taux d'identification, sont souvent parmi les types suivants: A2, A3, A4, AM-MOD et SU.

La séparation des résultats entre les phrases entièrement analysées et l'ensemble des phrases nous permet de voir une nette amélioration au niveau du rappel. On constate aussi que la section de "*development*" semble être sensiblement plus facile à analyser. Voici le tableau résumé des résultats pour le rappel des arguments :

Tableau 4.9: Résultats du rappel pour les arguments en fonction des types de phrase et des sections du corpus.

	Section de « *development*»	Section de « *test*»	Sections combinées
L'ensemble les phrases	63.8 %	55.4 %	60.1 %
Phrases entièrement couvertes	79.6 %	77.5 %	78.8 %

4.5 Comparaison avec LTH

Nous avons examiné les différentes façons d'interpréter les résultats obtenus en les comparant avec le « Gold Standard ». Toutefois, nous voulons aussi comparer Anasem avec le système qui a remporté la compétition de CoNLL 2008, LTH. Comme nous avons déjà accès aux modèles de LTH, il est relativement simple de l'exécuter sur les mêmes phrases qu'Anasem. Toutefois, comme la section de développement a servi pour son entrainement, nous n'utilisons que la section test pour faire nos comparaisons.

Afin d'obtenir les taux de précision et de rappel de LTH, nous utilisons l'outil d'évaluation fourni par la compétition CoNLL[14]. Puisque ce sont les résultats « *unlabeled* », c'est-à-dire les résultats qui ne tiennent pas compte des types d'arguments, qui nous intéressent, il a été nécessaire de modifier légèrement le code de l'outil afin de présenter les résultats des arguments sans labels (ceux-ci étaient calculés mais non affichés sur la console de l'outil). Ces modifications sont mineures et elles n'affectent pas le fonctionnement de l'outil, elles ne font qu'afficher des informations supplémentaires. Parmi les 314 arguments et 142 prédicats des phrases de la section test, LTH a identifié 250 arguments et 136 prédicats, ce qui correspond à un taux de réussite de 95.8 % pour les prédicats et 79.6 % pour les arguments. Voici la ventilation des résultats en fonction des arguments :

[14] http://elflaco.barcelona.corp.yahoo.com/conll2008/public/eval08.pl

Tableau 4.10: Ventilation des résultats de Johansson en fonction des types d'arguments

Types d'arguments	A0	A1	A2	A3	A4	AM-CAU	AM-LOC	AM-TMP	AM-PNC	AM-EXT	AM-MNR
Identifiés	69	97	16	1	2	2	5	9	2	0	11
Total	87	109	20	1	3	4	11	10	5	0	15
Pourcentage	79.3	89	80	100	66.7	50	45.5	90	40	-	73.3

AM-ADV	AM-DIS	AM-MOD	AM-DIR	AM-NEG	SU	R-A0	R-A1	R-AM-MNR	C-A1	C-AM-MNR
11	2	7	3	8	0	2	0	0	3	0
19	2	7	7	8	0	2	0	0	4	0
57.9	100	100	42.9	100	-	100	-	-	75	-

Dans un premier temps, on constate que LTH surpasse Anasem lorsqu'on le compare aux résultats sur l'ensemble des phrases de la section de tests. En effet, en ce qui concerne les prédicats, Anasem obtient un taux d'identification de 71.8 %, soit 24 % de moins que LTH. Concernant les arguments, LTH obtient 24.2 % de plus qu'Anasem (55.4 %). Il faut noter que ces résultats ne tiennent pas compte des arguments de type SU. En ignorant ce type d'arguments, Anasem gagne 1.2 % pour obtenir un taux d'identification de 56.6 %.

Ce qui est intéressant, c'est lorsqu'on compare ces résultats avec les résultats d'Anasem sur les phrases complètement analysées. En effet, puisqu'on peut attribuer la majorité des erreurs à l'absence de patrons sémantiques, comparer les phrases « complètes » nous donne une indication sur la nature des résultats qu'un système possédant un plus grand nombre de règles obtiendrait. La comparaison des prédicats est inutile puisque par définition nous ne choisissons que des phrases qui possèdent tous leurs prédicats. Nous pouvons donc comparer les résultats de LTH (79.6 %) avec ceux d'Anasem (79.3 %), en ne tenant pas compte des arguments de type SU. De plus, si l'on retire aussi les « self arguments », qui ne sont pas traités avec l'analyseur Anasem, les résultats dépassent ceux de LTH avec un taux d'identification des arguments de 82.3 %.

Lorsqu'on regarde les types d'arguments dans CoNLL, on constate qu'il y a deux grandes familles. La première, qu'on pourrait appeler les arguments principaux (A0, A1, A2, etc.), et la seconde, les arguments modificateurs (tous les autres). Il est intéressant de regarder le taux de succès de l'identification de ces familles d'arguments pour chaque analyseur. En effet, comme on peut voir dans le Tableau 4.11 avec Anasem, l'écart du taux de ces familles d'arguments est relativement bas (4 % et 5 %) alors qu'avec LTH il est plus élevé (15 %). De plus, on remarque que dans le cas des phrases entièrement analysées, Anasem identifie les arguments modificateurs avec un meilleur taux de réussite que LTH.

Tableau 4.11: Valeur de rappel pour la détection des arguments en fonction de leur famille

	Arguments principaux	Arguments modificateurs
Toutes les phrases du corpus de test – Anasem	57 %	53%
Les phrases entièrement couvertes du corpus de test – Anasem	81 %	76 %
Résultats sur le corpus de test - LTH	84 %	69 %

Ces résultats semblent montrer qu'Anasem est plus consistant dans ses résultats, et ce, indépendamment du type d'arguments. Par contre, il faut rappeler que les tests ont été faits sur un petit corpus et qu'il est possible que ceux-ci ne soient pas représentatifs d'un phénomène général.

4.6 Phrases non analysées

Nous avons mentionné 11 phrases qui n'avaient pas été analysées dans la section 4.2. Ces phrases ont produit des erreurs lors de l'exécution de l'analyse, ce qui a eu pour résultat une fin prématurée de l'exécution d'Anasem. Ce genre de problème peut être causé par deux situations; soit la phrase est problématique pour Anasem à la base, c'est-à-dire que la version d'Anasem, avant d'avoir été adaptée, produit une erreur lors de son analyse, soit il y a présence de caractères non traités par Anasem qui causent des problèmes. Dans le premier cas, puisque le but de ce mémoire n'est pas de corriger Anasem, nous avons ignoré ces phrases. Dans le second cas, c'est un problème de robustesse d'Anasem suite à l'utilisation de la nouvelle nomenclature, qui ne se produisait pas avec la nomenclature de Stanford. En effet, en utilisant la nomenclature CoNLL, nous avons introduit de nouvelles formes de relations syntaxiques qui possèdent un trait d'union. Par exemple, dans la phrase « *The aroma of patronage is in the air*. », selon CoNLL, le mot « in » possède comme relation syntaxique « *loc-prd* ». Ces dernières n'existaient pas auparavant et l'analyseur ne supporte pas ce genre de relation (qui utilise un trait d'union). Toutefois, étant donné que ce type de relation n'est pas commun, nous avons jugé préférable de concentrer nos efforts sur les phrases dites « fonctionnelles ».

CHAPITRE 5 DISCUSSION

Dans ce chapitre, nous allons revenir sur les résultats présentés dans la section précédente. L'objectif est de mettre en évidence les points importants se rapportant aux expérimentations que nous avons faites. Dans un premier temps, nous reviendrons sur les résultats bruts, c'est-à-dire sur l'ensemble des phrases, pour expliquer pourquoi ces résultats sont relativement bas. Nous poursuivrons avec les résultats de l'analyse des phrases complètes et ce qu'ils représentent. Par la suite, nous présenterons les limites qui se rapportent à l'adaptation et à la méthode d'évaluation que nous avons utilisées. Finalement, nous examinerons la question de l'influence des types d'arguments sur les résultats.

5.1 Erreurs d'analyse

Comme on a pu le constater dans la section 4.2, les résultats sur l'ensemble des phrases sont relativement bas. En effet, avec un taux d'identification des prédicats de 76.8 % et d'identification des arguments de 60.1 %, ces résultats sont quand même loin de ceux de Johansson (avec 95.8 % et 79.6 % respectivement). Il y a deux raisons possibles qui expliquent pourquoi une phrase n'est pas analysée correctement. D'abord, il survient parfois des erreurs au niveau des informations (relations syntaxiques, catégories grammaticales, etc.) qu'Anasem utilise, et ce, malgré l'utilisation d'un « Gold Standard ». Dans ces cas, il est impossible d'utiliser les patrons correctement. La seconde raison est reliée au nombre limité de patrons dans la version actuelle d'Anasem. Il y a donc plusieurs cas pour lesquels des patrons sont absents. Dans ces situations, les sous-arbres syntaxiques qui se rapportent aux patrons absents sont ignorés. Puisqu'on extrait tous les noms et verbes dans une phrase pour les insérer dans la DRS, on peut utiliser l'absence de prédicats comme indication de l'absence de patrons dans les phrases. C'est pour cette raison que nous avons décidé de distinguer les phrases dont tous les prédicats ont été identifiés.

Pour bien comprendre ce qui se produit, voici un exemple d'une phrase tirée du corpus de développement : « *This article is adapted from remarks at a Hoover Institution conference on national service, in which Mr Szanton also participated* ». Tout d'abord, voici l'arbre syntaxique obtenu pour cette phrase, après les transformations nécessaires (les mots de la phrase sont en **gras**).

```
root adapted/v
  adv from/prep
    pmod remarks/n
      loc at/prep
        pmod conference/n
          nmod a/d
          nmod institution/nnp
            name hoover/nnp
          nmod on/prep
            pmod service/n
              nmod national/jj
          nmod participated/v
            adv in/prep
              pmod which/wdt
                sbj szanton/nnp
                  title mr./nnp
            adv also/rb
  sbj article/n
    nmod this/d
  aux is/v
```

Comme expliqué précédemment, cet arbre est généré à partir des informations reçues en entrée provenant de CoNLL. Par la suite, on applique les règles de transformation afin de faciliter l'identification des patrons. Comme vu dans la section 2.2.3, au fur et à mesure que les patrons sont identifiés, Anasem ajoute les informations dans la DRS. Voici la DRS qui représente la phrase de l'exemple:

```
---------------------------------
[id1,e1,id2,id3]
---------------------------------
resolve(id1)
entity(id1,article)
event(e1,adapted,id1)
entity(id1,remarks)
entity(id2,conference)
entity(id3,service)
attribute(id3,national)
nmod(id2,id3,on)
attribute(id2,institution)
loc(id1,id2,at)
from(e1,id1)
---------------------------------
```

La première remarque qu'on peut avoir face à cette DRS, c'est qu'il y a une confusion au niveau des identifiants, « *article* » et « *remarks* »: ils sont tous les deux identifiés comme « id1 ». Ce genre de confusion survient lorsqu'une nouvelle entité est considérée comme déjà existante. C'est une erreur au niveau de l'attribution des identifiants, mais qui n'a pas d'impact sur l'identification des prédicats et arguments dans le reste de la phrase, c'est-à-dire que si les

identifiants avaient été correctement attribués, nous aurions obtenu la même analyse, mais avec les bons identifiants.

En second lieu, on observe qu'il manque des parties de la phrase. On constate que si l'on tente de reconstruire la phrase en n'utilisant que les éléments de la DRS, nous obtenons la phrase suivante : « *article adapted from remarks at institution conference on national service* » (bien entendu, certains mots, tels les déterminants ou les auxiliaires, ne se retrouvent pas dans la DRS). De cette phrase reconstituée, on remarque qu'il manque des informations, d'où le fait qu'elle soit classée comme incomplète. D'abord, il manque le nom de l'institution « *Hoover* ». On peut expliquer cette omission en regardant l'arbre syntaxique. Dans ce dernier, la branche qui représente « *Hoover* » a la forme suivante : « `name hoover/nnp` » et le problème se situe au niveau de la relation syntaxique. En effet, la relation « `name` » ne fait pas partie des patrons traités par Anasem dans sa version actuelle. Par conséquent, cette relation est ignorée et l'analyse se poursuit. L'autre omission est en fait une sous-section de la phrase. Comme on peut le voir dans l'arbre syntaxique, le sous-arbre qui a comme tête « `nmod` **`participated`**`/v` » n'a pas été traité. Encore une fois, ce manque dans l'analyse est causé par l'absence de patrons qui traitent les « `nmod`» qui sont des verbes. Donc, dans cette phrase, il y a deux sections qui ne sont pas couvertes par les patrons et c'est ce qui cause les erreurs d'analyse. Pour conclure l'exemple, voici le tableau de comparaison qui met en évidence les absences.

Tableau 5.1: Tableau de comparaisons pour la phrase de l'exemple

Prédicats	Détecté	A0	A1	A2	A3	R-A1	AM-DIS	commentaire
Adapt	1		1		1			
Remark	1		1	0				Self argument (A2)
Conference	1			1				
Service	1		0	1				Self argument (A1)
participate	0	0	0			0	0	

Cet exemple reflète bien la situation qui se produit dans la majorité des phrases incomplètes. Malgré nos efforts, nous ne couvrons qu'une petite partie des patrons existants, et cela a un effet direct sur nos résultats. Par contre, lorsqu'on regarde les phrases complètes qui ont un taux d'identification de 82.9 %, on voit que ce sont des phrases dont la majorité des patrons sont couverts, par conséquent on peut voir le potentiel de cette méthode. En effet en ajoutant des patrons, on ne peut qu'améliorer les résultats.

Nous avons constaté que dans certaines situations, malgré nos efforts, nous n'avons pas été capables de transformer l'arbre généré avec le format de CoNLL afin d'obtenir la structure de l'arbre avec le format de Stanford. Dans ces situations, il est donc très difficile de repérer les patrons puisqu'ils n'ont pas la forme attendue. La phrase suivante représente un exemple de cette situation: « *They cannot stop to grasp and embrace and sit in the back seat of cars along a dark country lane.* »

Tableau 5.2: Comparaison entre la structure d'une phrase pour différents formats

Arbre avec le format de Stanford	Arbre avec le format CoNLL

root and/cc
  coord not/rb
  nsubj they/prp
  neg stop/v
  xcomp grasp/v
  aux to/prep
  cc and/cc
  aux can/md
  coord and/cc
  coord not/rb
  nsubj they/prp
  neg stop/v
  xcomp sit/v
  prep in/prep
  pobj seat/n
  det the/d
  amod back/jj
  prep of/prep
  pobj cars/n
  prep along/prep
  pobj country-lane/n
  det a/d
  amod dark/jj
  aux can/md
  coord not/rb
  nsubj they/prp
  neg stop/v
  xcomp embrace/v
  aux can/md
	root not/rb
	  sbj they/prp
	  neg stop/v
	  prp and/cc
	  coord grasp/v
	  aux to/prep
	  coord and/cc
	  coord embrace/v
	  aux to/prep
	  coord sit/v
	  aux to/prep
	  loc in/prep
	  pmod seat/n
	  nmod the/d
	  nmod back/jj
	  nmod of/prep
	  pmod cars/n
	  loc along/prep
	  pmod lane/n
	  nmod a/d
	  nmod dark/jj
	  nmod country/n
	  aux can/md

Dans cet exemple, un des problèmes majeurs est la présence de « *xcomp* », qui ne possède pas d'équivalent direct dans CoNLL. Dans ce cas, les « *xcomp* » sont des « *prp* » dans CoNLL. De plus, l'énumération et la négation ne sont pas traitées de manière identique. Avec CoNLL, la racine est la négation alors qu'avec Stanford, c'est l'énumération (le « *and* ») qui est la racine. Dû aux différences entre les structures et la nomenclature des formats, l'analyse est très compliquée. Donc, dans ce cas-ci, le problème ne se situe pas au niveau des patrons, mais plutôt pendant la

transformation de l'arbre syntaxique. C'est un autre type de problème auquel on pouvait s'attendre à cause de la complexité de la langue ainsi que la grande quantité de formulations. Dans le cadre de ce mémoire, nous avons utilisé un nombre restreint de règles de prétraitement afin de couvrir les cas les plus communs.

5.2 Importance des analyses complètes

Les résultats des analyses des phrases complètes sont très bons et même meilleurs que ceux de Johansson en ne comptant pas les « *self arguments* » dans le sous-ensemble de phrases testées. Malgré le fait que ces phrases ne représentent qu'environ seulement 50 % de l'ensemble des phrases, ces résultats sont très importants. Comme nous l'avons expliqué précédemment, nous pouvons les utiliser comme un indicateur du potentiel de cette méthode. En effet, le cœur de ce système repose sur les patrons. Si un patron est absent, l'analyse ne sera pas correcte. C'est pourquoi, en ne regardant que les résultats des analyses complètes, on obtient les résultats d'un système dont la majorité des patrons sont couverts.

Il est important de noter que lorsque l'analyse d'une phrase est classée comme entièrement couverte, cela ne signifie pas que tous les éléments sémantiques ont été identifiés. Une phrase entièrement couverte indique que tous les prédicats ont été identifiés. Cette classification permet de savoir si certaines sections importantes de la phrase, c'est-à-dire celles contenant un prédicat, ont été ignorées. Par conséquent, les patrons manquants dans les phrases entièrement couvertes ne concernent que les arguments. On peut en déduire que ces patrons affectent principalement les feuilles de l'arbre syntaxique, c'est-à-dire les relations qui impliquent des mots qui ne possèdent pas de dépendant, puisqu'ils n'entrainent pas l'omission de sous-sections importante de l'arbre (comme dans le cas du premier exemple de la section 5.1).

De manière pratique, il est presque impossible de couvrir l'ensemble des formes et formulations dans une langue avec un nombre fini de patrons. Par contre, il est plus réalisable d'avoir un système qui nous donne une analyse complète dans la grande majorité des cas. Les résultats obtenus nous donnent une bonne indication de ce que nous pourrions obtenir avec un tel système.

5.3 Limites

Dans l'ensemble de ce projet, nous avons identifié quelques limitations. Tout d'abord, il y a les limitations qui sont reliées à l'adaptation de la version originale de l'analyseur pour qu'il soit compatible avec CoNLL, notamment, les différences de granularités et de formats de sortie, la sélection des phrases de tests et l'absence de ressources externes. Il y a aussi les limites qui concernent l'évaluation comme telle. En effet, il faut tenir compte du fait que c'est une analyse manuelle, que l'évaluation est effectuée sur une petite partie du corpus et qu'on utilise une analyse syntaxique « parfaite » (prise directement du « Gold Standard »).

Tout d'abord, le problème de la granularité est une des limites de cette solution. En effet, à partir du moment où les adaptations ne sont pas bijectives, il y a des risques de pertes d'informations. Lorsqu'il y a des relations syntaxiques dans une représentation qui englobe plus d'une relation de l'autre représentation, cela peut créer de la confusion (« NMOD » est un bon exemple de ce cas), et ce, malgré les étapes supplémentaires par lesquelles nous passons pour tenter de minimiser ce problème, comme l'utilisation de la catégorie grammaticale. Il est donc important d'en tenir compte lors de l'analyse des résultats.

Une autre limitation à notre solution se situe à l'étape finale qui consiste à comparer des systèmes qui utilisent deux formats de sortie différents pour la même tâche. Faire une comparaison manuelle implique une intervention humaine qui apporte deux problèmes majeurs. D'abord, cela limite grandement la quantité de tests qu'il est possible d'effectuer. En effet, nous ne pouvons pas comparer des milliers d'analyses puisque pour chaque phrase analysée il faut un temps considérable pour faire la comparaison. Par conséquent, la représentativité des résultats dépend grandement des phrases choisies. D'où l'importance de la sélection des phrases analysée pour éviter à la fois le biais et les sous-ensembles non représentatifs. Le second problème est relié à l'évaluation manuelle. Afin d'éviter un maximum d'erreurs, nous avons utilisé une méthode simple pour faire la comparaison. De plus, il est presque impossible d'être totalement impartial et donc, encore une fois, il faut garder en tête ces limites lorsqu'on regarde les résultats.

Une autre limite importante se situe au niveau de la méthode d'évaluation. En effet, puisqu'Anasem n'utilise pas exactement les mêmes ressources que les systèmes qui participaient à la campagne d'évaluation de CoNLL, il est impossible d'attribuer les sens de NomBank et PropBank aux prédicats identifiés. De plus, non seulement les formats de sortie sont très

différents (DRS vs prédicats-argument basés sur des cadres sémantiques), mais la méthode de sélection des prédicats l'est aussi. Nous avons pu constater que pour CoNLL, un des critères pour qu'un mot soit considéré comme un prédicat est qu'il possède au moins un argument. Ce qui n'est pas le cas pour Anasem. C'est pourquoi nous avons dû limiter notre comparaison à la mesure du rappel et que nous n'avons pas pu mesurer la précision sur l'ensemble du corpus. C'est cela qui nous a poussé à faire une autre évaluation de la mesure de précision. Malgré le fait que cette mesure ait été calculée par rapport aux DRS et non par rapport à CoNLL, elle donne une bonne indication de la précision en général.

Il faut aussi prendre en considération le corpus. D'abord, pour faire notre évaluation nous n'utilisons que 101 phrases sur les milliers de phrases disponibles. Il est donc difficile de tirer des conclusions définitives avec un corpus de cette taille. Par contre, les résultats obtenus nous donnent une bonne approximation des performances de notre système. Il faut aussi considérer la qualité des phrases dans le corpus. Certaines phrases sont parfois formulées de manière inhabituelle ou elles sont trop complexes, ce qui donne des résultats non représentatifs de la langue en général. Voici quelques phrases qui démontrent bien ces phénomènes (toutes les phrases sont extraites directement du corpus):

- "*It did not.*"
- "*And so he had, so he had.*"
- "*He, and Mrs. Dalloway, too, had never permitted themselves the luxury of joys that dug into the bone marrow of the spirit.*"
- "*At law school, the same."*
- "*His parents talked seriously and lengthily to their own doctor and to a specialist at the University Hospital -- Mr. McKinley was entitled to a discount for members of his family -- and it was decided it would be best for him to take the remainder of the term off, spend a lot of time in bed and, for the rest, do pretty much as he chose -- provided, of course, he chose to do nothing too exciting or too debilitating.*"

Finalement, il y a une importante différence au niveau des données utilisées en entrée. En effet, lors de la compétition, les participants devaient faire à la fois l'analyse sémantique et syntaxique. Puisque notre système ne fait que l'analyse sémantique, Anasem utilise l'analyse

syntaxique disponible dans le corpus, soit une analyse syntaxique dite « parfaite ». Ceci donne un avantage à Anasem, puisqu'il n'a pas besoin de tenir compte des erreurs possibles au niveau de l'analyse syntaxique. Toutefois, les systèmes de l'état de l'art en analyse syntaxique atteignent généralement de très bons résultats(McClosky, et al., 2012). .

5.4 Impact des types d'arguments

Dans cette section, nous considérerons les résultats combinés des phrases, sans tenir compte de la section du corpus d'où elles proviennent. Dans un premier temps, nous nous concentrerons sur l'ensemble des phrases pour ensuite comparer avec les résultats des phrases entièrement couvertes. Il est important de rappeler que nous n'avons pas évalué la validité du type des arguments, mais seulement si la relation en question avait été extraite par Anasem. Puisque nous travaillons avec le rappel, les chiffres obtenus représentent des relations identifiées, mais pas nécessairement avec le type approprié. Les arguments manquants représentent une absence de relation entre le prédicat et son argument, et non pas une mauvaise relation.

On peut d'abord considérer les arguments les plus communs en se rapportant au premier tableau de la section 4.2.3 où est présentée la ventilation des résultats en fonction des types d'arguments. Dans ce tableau, on remarque que la plupart des relations sont de type A0 et A1. Ce qui est tout à fait attendu, puisque pour les verbes, dans la majorité des cas, le A0 représente le sujet de l'action et le A1, le récipiendaire de l'action. Bien entendu, la définition de ces types d'arguments est propre au prédicat auxquels ils se rapportent, et donc parfois A0 ou A1 peuvent être différents de ce à quoi on s'attend. Par exemple, pour le verbe « *want*.01 », A0 est le « *wanter*» et A1 est « *thing wanted* ». Dans ce cas-ci, ce verbe possède aussi un A2 (*Beneficiary*), A3 (*in exchange for*) et un A4 (*from*). On peut comprendre qu'il est presque impossible d'associer le bon type d'arguments sans avoir accès à la définition des prédicats et des arguments. Par contre, pour les noms, il est commun d'avoir seulement un A1 et pas de A0. Par exemple, dans « *a cargo of crude* » le nom « cargo » n'a qu'un A1 qui est « *of crude*». De plus, pour « cargo » en particulier, il ne peut y avoir qu'un A1 (qui sert de quantifieur) et/ou un A3 (qui représente un thème secondaire). Il n'y a pas de définition pour les arguments A0 ou A2. On constate que pour les arguments de types A0 et A1, nos résultats sont relativement bas, soit respectivement 62.2 % et 65.9 % (Tableau 4.3). De plus, en regardant le tableau de la section 4.2.2, soit les phrases entièrement analysées, on obtient un taux respectivement de 76 % et de 89.3 % pour les mêmes

types d'arguments. Donc, en ce qui concerne les arguments de type A0 et A1, on peut dire qu'ils sont bien traités par rapport aux autres types, mais qu'ils sont beaucoup affectés lorsque la phrase n'est pas complètement analysée. Toutefois, on remarque que A0 est presque tout le temps inférieur à A1. Il y a deux raisons majeures qui expliquent ce phénomène. D'abord, plus de la moitié (11 sur 21) des « self arguments » sont du type A0, alors qu'il n'y en a que 2 de type A1. La seconde raison se rapporte au type de prédicat. Il est vrai que dans la majorité des cas, lorsque le prédicat est un verbe, les arguments A0 et A1 sont tous les deux présents, mais lorsque le prédicat est un nom, c'est moins évident. L'argument A1 semble plus facile à identifier que le A0.

Il y a plusieurs types d'arguments qui sont représentés dans les phrases à moins de 10 reprises. Ces types, soit A4, AM-CAU, AM-PNC, AM-EXT, AM-DIR, R-* et C-*, ne sont pas assez présents pour qu'on puisse vraiment en tirer des conclusions. De plus, lorsqu'on regarde les résultats des phrases complètes, ces types sont encore moins représentés. Nous allons donc les laisser de côté.

Tous les autres types reviennent entre 5 et 10 fois au moins. De ceux-ci, quelques-uns ressortent particulièrement, comme les AM-MOD où seulement 2 des 17 relations ont été trouvées. Par contre, on sait que ce type d'arguments est directement lié à une liste de mots prédéfinie (voir Tableau 1.2). Nous en concluons que les relations syntaxiques qui sont associées à ces mots ne font pas partie des patrons que nous avons déjà définis.

Comme nous l'avons expliqué précédemment, Anasem ne traite pas les arguments de type « *Support Chain* » (SU). C'est justement pourquoi nous avons présenté des résultats sans les compter. Ce type est un concept introduit par NomBank.

Il est intéressant de regarder le changement entre les résultats sur l'ensemble et ceux sur les analyses complètes particulièrement pour les AM-LOC, AM-TMP, AM-MNR, AM-ADV et AM-NEG. D'abord, avec les trois premiers, les résultats sont passés d’entre 65 % et 70 % à plus de 90 % (en conservant au moins 10 occurrences). Pour ce qui est des AM-ADV, ils sont passés de 66.7 % à 85.7 %. On remarque que ces types sont bien couverts par les patrons actuels lorsque tous les prédicats d'une phrase sont identifiés. Ensuite, avec AM-NEG, bien que nous n'ayons que très peu d'exemples, nous sommes passés de 60 % de réussite (6/10) à 100 % (5/5) pour les phrases complètes. Bien entendu, le fait qu'il y ait si peu d'exemples explique la grande variation des résultats.

Finalement, le dernier type d'arguments est le AM-DIS, qui se rapporte au discours. On remarque que les taux de réussite sont presque identiques entre l'ensemble des phrases et celles qui sont entièrement couvertes. Il semble donc que cette distinction n'ait que peu d'effet sur ce type. Toutefois, avec seulement 12 occurrences au maximum, ces résultats sont approximatifs.

CONCLUSION

Ce mémoire a pour but de présenter les résultats de la comparaison entre un analyseur sémantique à base de règles, Anasem, et un analyseur sémantique basé sur l'apprentissage machine, celui de Johansson de la campagne d'évaluation de CoNLL 2008 (LTH). Anasem possède une architecture modulaire et utilise une série de patrons qui sont appliqués sur l'analyse syntaxique basée sur une grammaire de dépendance. De son côté, LTH utilise des classificateurs qui sont entraînés sur un corpus. Dû à l'absence d'analyseur sémantique complet utilisant de l'apprentissage machine, nous avons comparé les systèmes sur une sous-tâche de l'analyse sémantique, soit l'étiquetage sémantique. Suite à cette comparaison, nous avons pu constater qu'Anasem peut obtenir des résultats comparables à LTH lorsque les phrases sont entièrement couvertes. De plus, nous avons abordé les difficultés rencontrées lors de l'adaptation et la comparaison de systèmes utilisant un formalisme et une nomenclature différents.

Initialement, nous avons posé une hypothèse. Nous pouvons maintenant revenir sur celle-ci afin de la discuter. L'hypothèse concerne les résultats obtenus lors de notre comparaison. Il est difficile de donner une réponse définitive sur ce sujet. Toutefois, nous avons pu constater qu'avec Anasem, dans son état actuel, nous pouvons obtenir des résultats comparables lorsque les phrases sont complètement couvertes. Par contre, si l'on regarde les résultats pour toutes les phrases, il y a encore place au perfectionnement. Tout au long de ce mémoire, nous avons constaté qu'il y a plusieurs aspects de ce projet à approfondir davantage. Tout d'abord, les résultats montrent qu'il y a une carence de couverture des patrons. Nous avons pu contourner quelques effets de ce problème en utilisant les phrases entièrement couvertes, mais il y aurait lieu de rajouter des patrons à notre analyseur pour qu'il puisse avoir une meilleure couverture. Toutefois, il faudrait aussi s'interroger sur la possibilité de créer assez de patrons de manière manuelle pour couvrir l'ensemble des phénomènes linguistiques. Peut-être que la solution serait d'utiliser un procédé mixte, c'est-à-dire d'utiliser une base de patrons créés manuellement et de la compléter en trouvant de nouveaux patrons à l'aide de l'apprentissage machine. Un autre problème, relié à la méthode de comparaison manuelle, est le nombre limité de comparaisons que nous avons pu réaliser. Il est évident que dans des travaux futurs, il serait nécessaire d'étendre le nombre de phrases comparées pour avoir des résultats plus représentatifs. Pour ce faire, il faudrait trouver un moyen d'automatiser la comparaison. De cette façon, nous pourrions comparer un très grand

nombre de phrases et éliminer le facteur humain. Il serait aussi intéressant de changer le corpus, et de faire cette comparaison sur des sujets différents afin de voir leurs impacts sur les résultats. Naturellement, cela implique d'entrainer les analyseurs basés sur l'apprentissage machine à nouveau ou bien d’explorer d'autres analyseurs, spécifiques au nouveau domaine.

Enfin, l'absence de standard en analyse sémantique a eu un grand impact sur notre travail. Nous croyons que l'établissement d'un tel standard favoriserait la recherche en analyse sémantique. Par contre, nous sommes conscients que c'est une tâche de taille et que nous en sommes encore loin.

BIBLIOGRAPHIE

Arrivé, M., 1969. Les Éléments de syntaxe structurale, de L. Tesnière. *Langue française*, 1(1), pp. 36-40.

Baker, C. F., Fillmore, C. J. & Lowe, J. B., 1998. *The Berkeley FrameNet Project.* Montréal, Association for Computational Linguistics, pp. 86-90.

Baluja, S., Mittal, V. O. & Sukthankar, R., 2000. Applying Machine Learning for High-Performance Named-Entity Extraction.. *Computational Intelligence*, November, 16(4), pp. 586-595.

Blackburn, P. & Bos, J., 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics.*:CSLI.

Bos, J., 2008a. Wide-Coverage Semantic Analysis with Boxer. Dans: J. Bos & R. Delmonte, éds. *Semantics in Text Processing. STEP 2008 Conference Proceedings.*:College Publications., pp. 277-286.

Bos, J., 2008b. Introduction to the shared task on Comparing Semantic Representations. Dans: *Proceedings of the 2008 Conference on Semantics in Text Processing.* Venice: Association for Computational Linguistics, pp. 257-261.

Cortes, C. & Vapnik, V., 1995. Support-vector networks. *{Machine Learning,* 20(3), pp. 273-297.

De Marneffe, M.-C. & Manning, C. D., 2008. "The Stanford typed dependencies. Dans: *COLING Workshop on Cross-framework and Cross-domain Parser* .

Fellbaum, C., 2010. WordNet. Dans: R. Poli, M. Healy & A. Kameas, éds. *Theory and Applications of Ontology: Computer Applications.*:Springer Netherlands, pp. 231-243.

Fillmore, C. J., 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, Volume 280, pp. 20-32.

Fillmore, C. J. R. J. &. B. C. F., 2004. *Framenet and representing the link between semantic and syntactic relations.*. Taipei, Institute of Linguistics, Academia Sinica, pp. 19-59.

Giménez, J. & Marquez, L., 2004. *SVMTool: A general POS tagger generator based on Support Vector Machin*.

Hofmann, T., 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2), pp. 177-196.

Johansson, R. & Nugues, P., 2008a. *The effect of syntactic representation on semantic role labeling.*, ACL, pp. 393-400.

Johansson, R. & Nugues, P., 2008b. *Dependency-based syntactic-semantic analysis with PropBank and NomBank.* Manchester, United Kingdom, Association for Computational Linguistics, pp. 183--187.

Kamp, H., 1981. A Theory of Truth and Semantic Representation. Dans: *Formal Semantics: The Essential Readings.* Blackwell, pp. 189-222.

Klein, D. & Manning, C. D., 2003. *Accurate Unlexicalized Parsing*, pp. 423-430.

Lee, H. et al., 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. Dans: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task.*:Association for Computational Linguistics, pp. 28-34.

Lin, C.-J., Weng, R. C. & Keerthi, S. S., 2008. Trust Region Newton Method for Logistic Regression. *The Journal of Machine Learning Research*, Volume 9, pp. 627-650.

Marcus, M., Santorini, B. & Marcinkiewicz, M. A., 1993. Building a large annotated corpus of English: the Penn Treebank.. *Computation Linguistics*, p. 19(2).

Marneffe, M.-C. d., MacCartney, B. & Manning, C. D., 2006. *Generating Typed Dependency Parses from Phrase Structure Parses*.

Marquez, L., 2009. *Semantic Role Labeling : Past, Present and Future*.

McClosky, D. et al., 2012. Stanford's System for Parsing the English Web. Dans: *Proceedings of First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL) at NAACL 2012.*

Meyers, A. et al., 2004. *The NomBank Project: An Interim Report.* Boston, Association for Computational Linguistics, pp. 24-31.

Montague, R., 1970. Formal Philosophy (Universal Grammar). Dans: R. Thomason, éd. New Haven: Yale University Press, pp. 222-246.

Palmer, M., Gildea, D. & Kingsbury, P., 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal,* p. 31:1.

Quinlan, J. R., 1986. Induction of decision trees. *Machine Learning,* 1(1), pp. 81-106.

Reid, S., Wasow, T. & Bender, E., 2003. *Syntactic Theory: A Formal Introduction.* 2nd éd.:CSLI.

Saussure, F. d., 1986. *Course in general linguistics.* LaSalle, Ill: Open Court.

Schuler, K. K., 2005. VerbNet: A broad-coverage, comprehensive verb lexicon. *Dissertations available from ProQuest,* p. Paper AAI3179808. http://repository.upenn.edu/dissertations/AAI3179808.

Soon, W. M., Ng, H. T. & Lim, D. C. Y., 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 01 12, XXVII(4), pp. 521-544.

Steedman, M., 1998. Categorial Grammar. Dans: F. Keil & R. Wilson, éds. *The MIT Encyclopedia of Cognitive Sciences.* Cambridge(Massachusetts): MIT Press.

Steedman, M., 2001. *The Syntactic Process.*:The MIT Press.

Stevenson, M. & Greenwood, M., 2009. Dependency Pattern Models for Information Extraction. *Research on Language & Computation,* pp. 13-39.

Surdeanu, M. et al., 2008. *The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies.*

Swier, R. S. & Stevenson, S., 2004. *Unsupervised Semantic Role Labelling.* Barcolona, Association for Computational Linguistics, pp. 95-102.

Toutanova, K., Klein, D., Manning, C. & Singer, Y., 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network.*, pp. 252-259.

Yarowsky, D., 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. Dans: *Proceedings of the 14th conference on Computational linguistics.* Nantes: Association for Computational Linguistics, pp. 454-460.

Zouaq, A., 2008. *Une approche d'ingénierie ontologique pour l'acquisition et l'exploitation des connaissances à partir de documents textuels.* Montreal.

Zouaq, A., Gagnon, M. & Ozell, B., 2010. *Grammaire de dependances et ontologies de haut niveau: vers un processus modulaire pour l'analyse semantique*.

Zouaq, A., Gagnon, M. & Ozell, B., 2010. *Semantic Analysis using Dependency-based Grammars.*, Bahri Publications, pp. 85-101.

ANNEXE 1 – RÈGLES DE TRANSFORMATIONS DU FORMAT STANFORD AU FORMAT CONLL

Utilisation des auxiliaires (règle A)

Phrase: "*He was getting chased*"

Avant	Après

root was/v
  sbj he/prp
  vc getting/v
  vc chased/v
	root chased/v
	  sbj he/prp
	  aux was/v
	  aux getting/v

Code en Prolog

```
patron(a,_/tree(_,[vc/_])).
transfo(a,R/T,R/TreeOut):-
        retirerFils([vc/VC],R/T,T2),
        findall(Fils,possedeFils(T2,Fils),FrereVC),
        findall(Fils2,possedeFils(VC,Fils2),FilsVC),
        info(VC,A),
        info(T,B),
        ajouterFils(FrereVC,tree(A,[aux/tree(B,[])]),Tree),
        ajouterFils(FilsVC,Tree,TreeOut),
        write('RULE A'),nl.
```

Utilisation de verbe à l'infinitif (to go, etc.) (règle B)

Phrase: "*He loved to go to school*"

Avant	Après

root loved/v
  sbj he/prp
  oprd to/prep
  im go/v
  dir to/prep
  pmod school/n
	root loved/v
	  oprd go/v
	  sbj he/prp
	  dir to/prep
	  pmod school/n
	  aux to/prep
	  sbj he/prp/v

Code en Prolog

```
patron(b,_/tree(_,[im/_])).
transfo(b,R/T,R/TreeOut):-
        retirerFils([im/IM],R/T,T2),
        findall(Fils,possedeFils(T2,Fils),FrereIM),
        findall(Fils2,possedeFils(IM,Fils2),FilsIM),
        info(IM,A),
        info(T,B),
```

```
        ajouterFils(FrereIM,tree(A,[aux/tree(B,[])]),Tree),
        ajouterFils(FilsIM,Tree,TreeOut),
        write('RULE B'),nl.
```

Utilisation d'un verbe et un subordonné de conjonction (règle C)

Phrase: "Tom says that Mia likes to swim"

Avant	Après

root says/v
  sbj tom/nnp
  obj that/prep
  sub likes/v
  sbj mia/nnp
  oprd to/prep
  im swim/v
	root says/v
	  sbj tom/nnp
	  obj likes/v
	  oprd swim/v
	  sbj mia/nnp
	  aux to/prep
	  sbj mia/nnp
	  complm that/prep

Code en Prolog

```
patron(c,_/tree(_,[sub/_])).
transfo(c,R/T,R/TreeOut):-
        retirerFils([sub/SUB],R/T,T2),
        findall(Fils,possedeFils(T2,Fils),FrereSUB),
        findall(Fils2,possedeFils(SUB,Fils2),FilsSUB),
        info(SUB,A),
        info(T,B),
        ajouterFils(FrereSUB,tree(A,[complm/tree(B,[])]),Tree),
        ajouterFils(FilsSUB,Tree,TreeOut),
        write('RULE C'),nl.
```

Utilisation des compléments prédicatifs (prd)(règle D)

Phrase: "*It would be easy to dance with Anna*"

Avant	Après

root would/md
  sbj it/prp
  vc be/v
  prd easy/jj
  amod to/prep
  im dance/v
  adv with/prep
  pmod anna/nnp
	root easy/jj
	  amod dance/v
	  adv with/prep
	  pmod anna/nnp
	  sbj it/prp
	  sbj it/prp
	  aux to/prep
	  sbj it/prp
	  aux would/md
	  cop be/v

Code en Prolog

```
patron(d,_/tree(_,[prd/_])).
transfo(d,R/T,R/TreeOut):-
```

```
retirerFils([prd/PRD],R/T,T2),
findall(Fils,possedeFils(T2,Fils),FrerePRD),
findall(Fils2,possedeFils(PRD,Fils2),FilsPRD),
info(PRD,A),
info(T,B),
ajouterFils(FrerePRD,tree(A,[cop/tree(B,[])]),Tree),
ajouterFils(FilsPRD,Tree,TreeOut),
write('RULE D'),nl.
```

Utilisation du possessif. (règle E)

Phrase: "*It is Marta's favorite car*"

Avant	Après

root is/v
  sbj it/prp
  prd car/n
  nmod marta/nnp
  suffix s/pos
  nmod favorite/jj
	root car/n
	  poss marta/nnp
	  possessive s/pos
	  nmod favorite/jj
	  sbj it/prp
	  cop is/v

Code en Prolog

```
patron(e,_/tree(_,[nmod/tree(_,[suffix/_])])).
transfo(e,R/T,R/TreeOut):-
      retirerFils([nmod/tree(Poss,Children)],R/T,R/T1),
      retirerFils([suffix/Suffix],poss/tree(Poss,Children),T2),
      ajouterFils([possessive/Suffix],T2,Tree),
      ajouterFils([Tree],R/T1,R/TreeOut),
      write('RULE E'),nl.
```

ANNEXE 2 – 101 PHRASES D'ÉVALUATION

Phrase de la section du "developement"

1. Economists are divided as to how much manufacturing strength they expect to see in September reports on industrial production and capacity utilization, also due tomorrow.
2. South Africa's National Union of Mineworkers said that about 10,000 diamond miners struck for higher wages at De Beers Consolidated Mines Ltd.
3. A few hours after the party launched its own affinity credit card earlier this month, the Tories raised the nation's base interest rate.
4. Under the laws of the land, the ANC remains an illegal organization, and its headquarters are still in Lusaka, Zambia.
5. Mr. Simmons said Keystone's new mill is expected to produce about 585,000 tons of steel rods this year, up from 413,000 tons in 1988.
6. The newly formed Resolution Trust Corp, successor to the Bank Board, filed suit against Mr Keating and several others on Sept 15.
7. Since Senate leaders have so far fogged it up with procedural smokescreens, promises of a cleaner bill are suspect.
8. Solicitor General Kenneth Starr argued that the 1973 Supreme Court decision, Roe vs Wade, recognizing a constitutional right to abortion, was incorrect.
9. Late last month, the appeals court agreed that most of the case should be dismissed.
10. Nevertheless, the company said shipments were up slightly to 679,000 metric tons from 671,000, buffing the impact of the unexpected earning decline.
11. Economists say an August rebound in permits for multifamily units signaled an increase in September starts, though activity remains fairly modest by historical standards.
12. But most advisers think the immediate course for individual investors should be to stand pat.
13. Trinity Industries Inc said it reached a preliminary agreement to manufacture 1,000 coal rail cars for Norfolk Southern Corp.

14. His recording later turned up as a court exhibit.
15. Moody's says the frequency of corporate credit downgrades is the highest this year since 1982.
16. Over the next days and weeks, they say, investors should look for stocks to buy.
17. Panic selling also was unwise during other big declines in the past.
18. Nevertheless, the problems of the junk market could prompt the Federal Reserve to ease credit in the months ahead.
19. Many takeover stocks plunged Friday, as speculators retained their confidence in corporate buyers but fled from the so called whisper stocks, the targets of rumored deals.
20. Administration officials say President Bush was briefed throughout Friday afternoon and evening, even after leaving for Camp David.
21. But the battle is more than Justin bargained for.
22. Are such expenditures worthwhile, then?
23. This article is adapted from remarks at a Hoover Institution conference on national service, in which Mr Szanton also participated.
24. This attitude is clearly illustrated in the treatment of Max, the trading room's most flamboyant character.
25. Walter Sisulu and the African National Congress came home yesterday.
26. It might well win Senate passage.
27. Then how should we think about national service?
28. But the officials feared that any public announcements would only increase market jitters.
29. The Boeing strike is starting to affect airlines.
30. When the Brady Task Force's powerful analysis of the crash was released in January 1988, it immediately reshaped the reformers ' agenda.
31. The aroma of patronage is in the air.

32. But such convolutions would still block the networks from grabbing a big chunk of the riches of syndication.

33. Mr Kaye had sold Capetronic Inc, a Taiwan electronics maker, and retired, only to find he was bored.

34. Other Hong Kong manufacturers expect their results to improve only slightly this year from 1988.

35. Housing starts are expected to quicken a bit from August's annual pace of 1,350,000 units.

36. But with foreign companies snapping up U.S movie studios, the networks are pressing their fight harder than ever.

37. Economists are divided as to how much manufacturing strength they expect to see in September reports on industrial production and capacity utilization, also due tomorrow.

38. It will have its headquarters in Munich.

39. Worse, Congress has started to jump on the Skinner bandwagon.

40. In fact, the network hopes to set up offices in Warsaw and anywhere else in the East Bloc that will have it.

41. Boys on busy street corners peddle newspapers of every political stripe.

42. The Greek courts have decided in favor of extradition in the Rashid case, but the matter awaits final approval from Greece's next justice minister.

43. The British Broadcasting Corp and the U.S State Department's Voice of America broadcast over Hungarian airwaves, though only a few hours a day each in Hungarian.

44. Radio Free Europe and its sister station for the Soviet Union, Radio Liberty, say they won't cut back their more than 19 hours of daily broadcasts.

45. But many of them can be quite profoundly reoriented by productive and disciplined service.

46. The results were announced after the stock market closed.

47. Firm prices were generally in line with the tentative prices announced earlier this fall.

48. In addition, there are six times as many troubled banks as there were in the recession of 1981, according to the Federal Deposit Insurance Corp.
49. Despite this loss, First Chicago said it doesn't need to sell stock to raise capital.
50. The annual interest rate for each of the next 11 years will be set each fall, when details of a new series are released.
51. He thinks government officials are terrified to let a recession start when government, corporate and personal debt levels are so high.

Phrase de la section du "test"

1. He rose late and went down in his bathrobe and slippers to have breakfast either alone or with Rachel.
2. Then began the journey through their own mine fields.
3. Clumps of brush that they passed were so many enchained demons straining in anger to tear and gnaw on his bones.
4. He did not like Boxell.
5. It will be good for you.
6. Rachel had to bend toward Scotty and ask him to repeat.
7. He had come here in order to test himself.
8. They had brandy in the library.
9. The walk and his fears had served to overheat him and his sweaty armpits cooled at the touch of the night air.
10. They visited the shipyards at Brest and Pierre had to sign the register, vouching for the integrity of the visiting foreigner.
11. Jefferson Lawrence was alone at the small, perfectly appointed table by the window looking out over the river.
12. You could hear them from our outpost.
13. That's to say, he was trigger happy.

14. He made the decision with his eyes open, or so he thought.
15. Two of our men were killed, a third was wounded.
16. Smiling at Warren's protestations, the old monk took his grip from him and led him down a corridor to a small parlor.
17. Soon they were picking their way along the edge of the stream which glowed in the night.
18. The dark forms moved like mourners on some nocturnal pilgrimage, their dirge unsung for want of vocal chords.
19. His father was a constant visitor.
20. The other patrons were taxi drivers and art students and small shopkeepers.
21. A foot misplaced, a leg missing.
22. He had come because he could not live out his life feeling that he had been a coward.
23. They lay on his lap, palms up, stiffly motionless, the tapered fingers a little thick at the joints.
24. We knew the enemy was subdued, because a flare was fired as the signal.
25. And then the questions came, eager, interested questions, and many compliments on his having overcome his infirmity.
26. They were far off and looked tiny.
27. On the forward slope in front of his own post stretched two rows of barbed wire.
28. Poverty imposes a kind of chastity on the ambitious.
29. Virginia and Rachel talked to each other quietly now, as allies who are political rather than natural might in a war atmosphere.
30. Since Mr. McKinley had to give a lecture, Rachel and Scotty drove home alone in the Plymouth.
31. She was wise enough to realize a man could be good company even if he did weigh too much and didn't own the mint.
32. She put the slipper neatly by its mate at the foot of the bed.

33. He had not mentioned Kate.
34. They can not stop to grasp and embrace and sit in the back seat of cars along a dark country lane.
35. The reporters had not yet discovered that this was his hideaway.
36. Then we pull out under our mortar and artillery cover, but nobody pulls out until I say so.
37. He did not spill over with hatred for the enemy.
38. He did not mind the useless, kindly questions.
39. She seemed to speak to herself.
40. He felt tired and full and calm.
41. They did not speak much.
42. He was able, now, to sit for hours in a chair in the living room and stare out at the bleak yard without moving.
43. Congress is full of politicians, and if you want to get along with them, you have to be politic.
44. He was calm, drugged, and lazy.
45. Scotty would reply softly and his father, apologetically, would ask him to repeat.
46. I heard of some that tried it back in the States, and he'd knock them clear across the room.
47. Therefore, he decided he was unfair to the young man and should make an effort to understand and sympathize with his point of view.
48. With leather cups fitted in his handlebars, he steered his bicycle.
49. The doctor, since Scotty was no longer allowed to make his regular trips into town to see him, came often and informally to the house.
50. Mr. McKinley examined everything with critical care, seeking something material to blame for his son's illness.

ANNEXE 3 – TABLE DE CONVERSION

Voici la liste des types de relation syntaxique pour Stanford (colonne de gauche) et l'équivalent le plus probable pour CoNLL (colonne de droite):

acomp	PRD
advcl	SUB
advmod	ADV
agent	PMOD
amod	NMOD
appos	APPO
attr	TMP
aux	OPRD
auxpass	VC
cc	DEP
ccomp	VC
complm	OBJ
conj	CONJ
conj_and	CONJ
conj_but	CONJ
conj_negcc	PMOD
conj_nor	CONJ
conj_or	CONJ
conj_than	SUB
conj_v.	PMOD
conj_vs.	PMOD
conj_yet	CONJ
cop	OBJ
csubj	SBJ
csubjpass	SBJ
dep	NMOD
det	NMOD
dobj	OBJ
expl	SBJ
infmod	IM
iobj	OBJ
mark	ADV
measure	AMOD
neg	ADV
nn	NMOD
nsubj	SBJ
nsubjpass	SBJ
num	NMOD
number	DEP

parataxis	ROOT
partmod	APPO
pcomp	VC
pobj	PMOD
poss	NMOD
preconj	NMOD
pred	VC
predet	NMOD
prep	TMP
prep_about	PMOD
prep_above	PMOD
prep_according_to	PMOD
prep_across	PMOD
prep_after	PMOD
prep_against	PMOD
prep_along	PMOD
prep_along_with	PMOD
prep_alongside	PMOD
prep_amid	PMOD
prep_among	PMOD
prep_around	PMOD
prep_as	PMOD
prep_as_of	PMOD
prep_at	PMOD
prep_away_from	PMOD
prep_barring	CONJ
prep_based_on	PMOD
prep_because	CONJ
prep_because_of	PMOD
prep_before	PMOD
prep_behind	PMOD
prep_below	PMOD
prep_between	PMOD
prep_beyond	PMOD
prep_by	PMOD
prep_close_to	PMOD
prep_compared_to	PMOD
prep_compared_with	PMOD
prep_concerning	NMOD

prep_contrary_to	PMOD
prep_depending_on	PMOD
prep_despite	PMOD
prep_due_to	PMOD
prep_during	PMOD
prep_excluding	PMOD
prep_followed_by	NMOD
prep_following	PMOD
prep_for	PMOD
prep_from	PMOD
prep_in	PMOD
prep_in_accordance_with	PMOD
prep_in_addition_to	PMOD
prep_in_front_of	PMOD
prep_including	PMOD
prep_instead_of	PMOD
prep_into	PMOD
prep_like	PMOD
prep_near	PMOD
prep_of	PMOD
prep_off	PMOD
prep_off_of	PMOD
prep_on	PMOD
prep_on_behalf_of	PMOD
prep_on_top_of	PMOD
prep_out_of	PMOD
prep_outside	PMOD
prep_over	PMOD
prep_pending	PMOD
prep_per	PMOD
prep_regarding	OBJ
prep_since	PMOD
prep_such_as	PMOD
prep_than	PMOD
prep_through	PMOD
prep_throughout	PMOD
prep_to	PMOD
prep_toward	PMOD
prep_under	PMOD
prep_unlike	PMOD
prep_until	PMOD

prep_upon	PMOD
prep_versus	PMOD
prep_via	PMOD
prep_vs.	PMOD
prep_with	PMOD
prep_within	PMOD
prep_without	PMOD
prepc_about	PMOD
prepc_after	PMOD
prepc_around	IM
prepc_as	PMOD
prepc_at	PMOD
prepc_before	PMOD
prepc_by	PMOD
prepc_for	PMOD
prepc_from	PMOD
prepc_in	PMOD
prepc_including	PMOD
prepc_into	SUB
prepc_like	P
prepc_of	PMOD
prepc_on	PMOD
prepc_over	PMOD
prepc_such_as	PMOD
prepc_than	PRD
prepc_to	PMOD
prepc_toward	PMOD
prepc_unlike	PRD
prepc_upon	NMOD
prepc_while	SUB
prepc_with	PMOD
prepc_without	PMOD
prt	PRT
purpcl	IM
quantmod	DEP
rcmod	NMOD
rel	PMOD
tmod	TMP
xcomp	IM
xsubj	SBJ