Fusionner pour mieux analyser□ quelques idées et une première expérience

Francis Brunet-Manquat

GETA-CLIPS-IMAG (UJF & CNRS)
BP 53 – 38041 Grenoble Cedex 9, France
Francis.Brunet- Manquat@imag.fr

Mots-clefs – Keywords

Analyse de dépendance, analyse syntaxique, intégration, fusion.

Dependency parsing, syntactic parsing, integration, fusion.

Résumé – Abstract

L'objectif de cet article est de présenter nos travaux sur l'analyse d'un énoncé vers une structure de dépendance. Cette structure décrit les relations entre mots, des relations syntaxiques mais également des relations sémantiques de surface de l'énoncé de départ dans un certain contexte. L'idée est de créer une plateforme d'analyse capable d'intégrer des analyseurs linguistiques existants (syntaxiques ou de dépendance) et de fusionner leurs résultats dans le but d'obtenir une analyse de dépendance pour des énoncés quelconques.

The article's goal is to present our work about the parsing of a sentence to a dependency structure. This structure describes the relations between words, syntactic relations but also surface semantic relations of a sentence in a certain context. The idea is to create an analysis' platform capable of integrating existing linguistic parsers (syntactic parsers or dependency parsers) and to fusion their results to obtain a dependency parsing from a sentence.

1 Introduction

Notre laboratoire est impliqué dans deux projets internationaux importants□CSTAR, avec le projet européen associé NESPOLE! (http://nespole.itc.it) pour la traduction de parole et UNL, Universal Networking Language (http://www.unl.ias.unu.edu), pour la traduction de l'écrit. Ces deux projets se caractérisent notamment par la présence d'une représentation *pivot* des énoncés et par le fait que l'énoncé à traduire est susceptible d'être "□ruité□, c'est-à-dire pas forcément conforme à la grammaire académique de la langue française.

L'objectif de cet article est de présenter nos travaux sur l'étude et la réalisation d'un outil *robuste* pour l'analyse de dépendance. Cet outil doit être capable d'intégrer des analyseurs linguistiques existants (syntaxique ou de dépendance) et de fusionner leurs résultats dans le but d'obtenir une analyse de dépendance pour des énoncés quelconques. L'outil doit être

robuste□capable de fournir une analyse, même partielle, d'une entrée mal formée (erreur de syntaxe, mots erronés). La structure de dépendance ainsi obtenue décrit les relations entre mots, des relations syntaxiques mais également des relations sémantiques de surface de l'énoncé de départ dans un certain contexte. L'outil créé doit pouvoir être utilisé et paramétré par un utilisateur non spécialiste de l'informatique, mais ayant une bonne connaissance linguistique de la langue à traiter.

L'idée principale est de mettre en place une plateforme d'analyse capable d'extraire des informations linguistiques provenant d'un ensemble d'analyseurs et de traiter ces différentes informations pour obtenir un ensemble de relations probables entre les mots d'un énoncé. Pour ce faire, nous devons dans un premier temps étudier les résultats obtenus à partir de différents analyseurs pour déterminer si les informations de chaque résultat peuvent être extraites et fusionnées. À partir d'une telle idée, nous pourrons construire un analyseur de dépendance plus robuste et plus adaptable en combinant différentes sources d'analyse.

À long terme, nous comptons valider l'étude et la réalisation de notre outil d'analyse de dépendance, en concevant une extension permettant de générer des hypergraphes UNL (Universal Networking Language). Un hypergraphe décrit le *sens* de l'énoncé dans un contexte donné. Il est composé d'arcs représentant des relations sémantiques (tels qu'agent, objet, but,etc.) et de nœuds représentant les UW («□Universal Word□», ou acceptions interlingues) auxquelles sont associés des attributs sémantiques (Sérasset et Boitet 2000). Pour générer ce type de graphe, la structure de dépendance fournie par notre outil d'analyse nous semble particulièrement adéquate, vu qu'elle représente également un ensemble de relations entre mots. Une telle structure associée à un dictionnaire spécifique nous permettra dans un premier temps de générer des hypergraphes simples, composés d'informations sémantiques de surface.

Dans cet article, nous allons présenter l'étude réalisée sur différents analyseurs déjà existants. Puis nous décrirons l'organisation de notre outil d'analyse de dépendance. Finalement, nous présenterons une première expérimentation.

2 Étude

Pour réaliser une plateforme d'analyse regroupant plusieurs analyseurs, il faut, dans un premier temps, connaître les particularités de ces analyseurs et surtout les caractéristiques des résultats fournis par ceux-ci. Une telle connaissance de chaque analyseur nous permet de déterminer la complémentarité de certains analyseurs, la performance de chaque analyseur selon le type d'énoncé (bruité ou non), la pertinence selon la typologie, le traitement de certains phénomènes linguistique, etc. Cette étude a été réalisée dans un premier temps sur des analyseurs du français, nous compléterons progressivement cette évaluation avec d'autres analyseurs du français puis nous étudierons les analyseurs d'autres langues, notre outil se devant d'être adaptable au plus grand nombre de langues possibles.

2.1 Analyseurs

Nous pouvons distinguer deux types d'analyseurs□les analyseurs linguistiques, fondés sur des formalismes grammaticaux, et les analyseurs probabilistes, fondés sur l'apprentissage à partir de corpus. Pour notre étude et la réalisation de notre outil, nous avons choisi d'utiliser dans un premier temps des analyseurs linguistiques. La plupart de ces analyseurs se

répartissent en trois catégories en fonction des résultats qu'ils fournissent (Monceaux et Robba 2002)□□

- *Les analyseurs fondés sur les constituants* qui retournent une segmentation en groupes.
- *Les analyseurs fondés sur les dépendances* qui retournent les dépendances entre mots d'une phrase.
- *Les analyseurs fondés sur les constituants et les dépendances* qui retournent une segmentation en groupes et des relations de dépendance entre ces groupes et entre les mots.

Dans la suite, nous présentons les quatre analyseurs à notre disposition□ l'analyseur syntaxique de la plateforme Xelda développé par Rank Xerox, l'analyseur du GREYC développé par Jacques Vergne, l'analyseur syntaxique du projet Lidia développé au GETA. Ces trois analyseurs sont fondés sur les constituants et les dépendances.

2.1.1 Analyseur syntaxique de la plateforme Xelda

L'outil d'analyse IFSP (Incremental Finite-State Parser) (Ait-Mokhtar et Chanod 1997) de la plateforme Xelda (http://www.xrce.xerox.com/ats/xelda/) est un analyseur syntaxique partiel qui construit les groupes syntagmatiques noyaux des phrases en entrée, puis utilise la structure ainsi construite pour extraire des relations syntaxiques entre les mots (sujet, sujet passif, objet direct, etc.). Les phrases sont préalablement segmentées et étiquetées avec un étiqueteur morpho-syntaxique, afin de réduire les éventuelles ambiguïtés d'analyse. Les phrases ainsi étiquetées sont alors segmentées en groupes noyaux (chunks) dont les modèles sont décrits par des suites d'étiquettes morpho-syntaxiques. Lorsque les groupes noyaux sont délimités, l'analyseur assigne à la phrase des étiquettes de fonction syntaxique principale (sujet, objet, etc.) et en fonction des règles d'extraction spécifiées sur la structure des groupes noyaux, extrait des relations de dépendances syntaxiques explicites entre les mots.

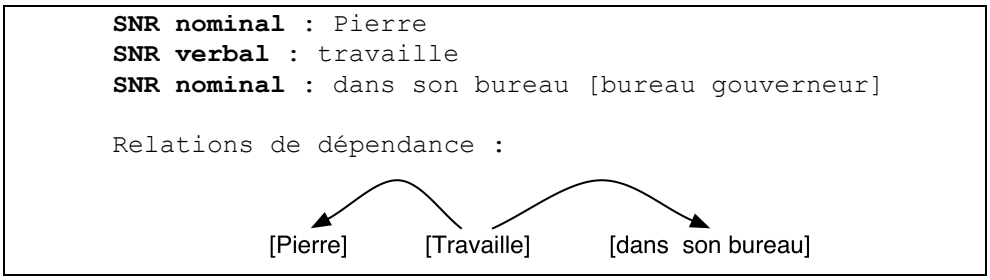
```
SUBJ(Pierre,travaille)
VMODOBJ(travaille,dans,bureau)

0:  (ROOT)
   1: SC(TAG <0 1>)
      2: NP(TAG <0 0>)
         3: Pierre PROPRN
         3: SUBJ(FUNC)
      2:  :v(HEAD)
      2: travaille VERB
   1: PP(TAG <2 4>)
      2: dans PREP
      2: son DET
      2: bureau NOUN
   1: .^.+SENT+SENT+SENT[5](WORD)
```

Exemple 1 : Résultat Xelda filtré de la phrase "Pierre travaille dans son bureau."

2.1.2 Analyseur du GREYC (Jacques Vergne)

L'analyseur syntaxique de GREYC combine des techniques d'étiquetage grammatical (Tagging) pour construire des segments non-récursifs (SNR) et un algorithme de calcul de dépendances pour calculer la structure fonctionnelle (Vergne 1998). Ce système est

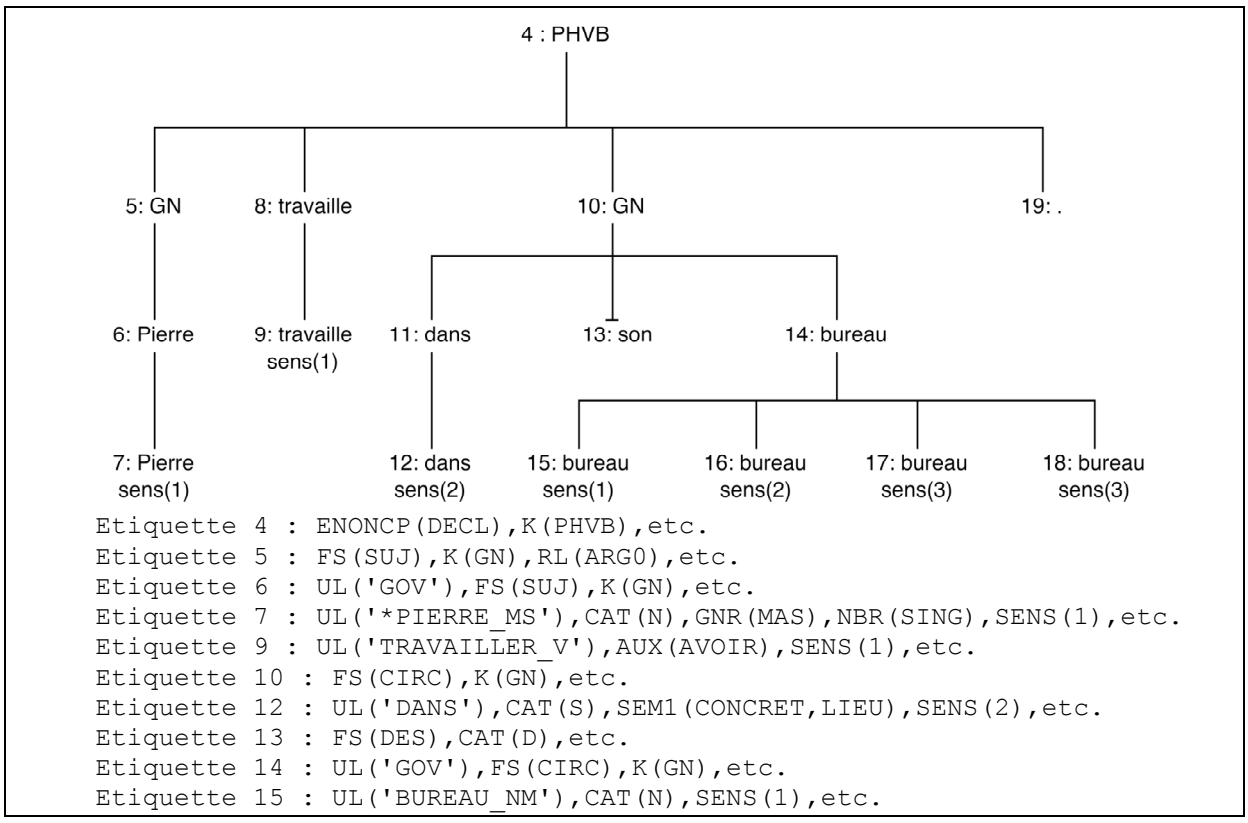
déterministe et a une complexité linéaire (http://users.infocaen.fr/~jvergne/). L'analyseur a été testé sur différents types de corpus (extraits d'articles du journal Le Monde) et a été évalué dans le cadre de l'action Grace (Adda et al 1999).



Exemple 2 : Résultat filtré de la phrase "Pierre travaille dans son bureau."

2.1.3 Lidia

Cet outil d'analyse a été développé pour le projet LIDIA (Large Internationalisation des Documents par Interaction avec l'Auteur) (Boitet et Blanchon 1995) à partir du générateur de système de TAO ARIANE-G5 (Boitet 1990). Cet analyseur fournit un arbre de constituant comme résultat d'analyse. Le modèle linguistique utilisé repose sur une structure arborescente «□multi-niveau□. La structure produite est dite multi-niveau car les nœuds portent, entre autres, des décorations complexes qui représentent trois niveaux d'interprétation□le niveau des classes syntaxiques et des classes syntagmatiques, le niveau des fonctions syntaxiques, et enfin le niveau des relations logiques et sémantiques.



Exemple 3 : Résultat LIDIA filtré de la phrase "Pierre travaille dans son bureau."

2.2 Étude qualitative

Cette étude nous a permis de répertorier les différentes informations produites par les analyseurs (relations entre mots, groupements de mots, etc.) et de comparer les différentes particularités de chaque analyseur afin de vérifier la présence d'informations utiles pour réaliser une analyse de dépendance. Cette étude nous permet surtout d'évaluer la complémentarité des analyseurs et de définir certains des critères à utiliser lors de la fusion des résultats des différents analyseurs. Nous avons classé toutes ces informations linguistiques en 6 catégories⊡

- *Syntagmes*⊡ groupe nominal, groupe verbal, groupe adjectival, groupe adverbial, groupe prépositionnel, proposition relative, proposition subordonnée, etc.
- *Relations syntaxiques*⊡ sujet, objet, gouverneur, complément d'agent, complément circonstanciel, coordination, apposition, inclusion, etc.
- *Relations logiques (relations prédicatives)*⊡numérotation des arguments, etc.
- *Relations sémantiques*⊡ contexte, but, cause, conséquence, instrument, bénéficiaire, conséquence, manière, matière, lieu, qualification, etc.
- *Catégories morpho-syntaxiques*⊡ nom, verbe, adjectif, adverbe, subordonnant, coordonnant, ponctuation, mot inconnu, pronom, préposition, déterminant, participe, etc.
- *Variables grammaticales*⊡genre, nombre, temps, mode, personne, etc.

	Analyseur Vergne	Analyseur Xelda	Analyseur Lidia
Syntagmes			
Groupe nominal	x	x	x
Groupe verbal	x	x	x
Groupe prépositionnel	x	x	x
Groupe adjectival		x	x
Groupe adverbial			x
Proposition relative	x		x
Proposition infinitive			x
Proposition participale	x	x	x
Proposition subordonnée	x		x
Relations syntaxiques			
Sujet	x	x	x
Objet	x	x	x
Complément d'agent			x
Complément circonstanciel			x
Coordination	x		x
⋮			

Figure 1 : Extrait d'un tableau de classement

Nous avons cherché à détailler chacune de ces six catégories puis de faire correspondre, si possible, à chaque information détaillée les informations fournies par les analyseurs (voir Figure 1 : Extrait d'un tableau de classement). Ceci nous permet de voir plus clairement quelles informations sont disponibles, quelles informations pourront être contradictoires et surtout quelles informations seront complémentaires.

3 Outil d'analyse de dépendance

L'outil d'analyse de dépendance ne doit pas intégrer les analyseurs, mais il doit être capable d'extraire les informations linguistiques de leurs résultats, de les interpréter et de les fusionner. Notre outil se compose de quatre modules⊡

- *Module de chargement*⊡Ce module extrait les informations linguistiques des résultats obtenus par un analyseur linguistique.
- *Module de normalisation*⊡ Ce module traite l'information extraite pour la faire correspondre à notre norme/standard. À chaque information est associée un indice de confiance calculé en fonction de l'information et des particularités de l'analyseur.
- *Module de fusion*⊡Ce module fusionne les informations normalisées.
- *Module de génération*⊡Ce module génère une structure de dépendance en fonction des informations fusionnées, des indices de confiance associés et de nos règles de génération.

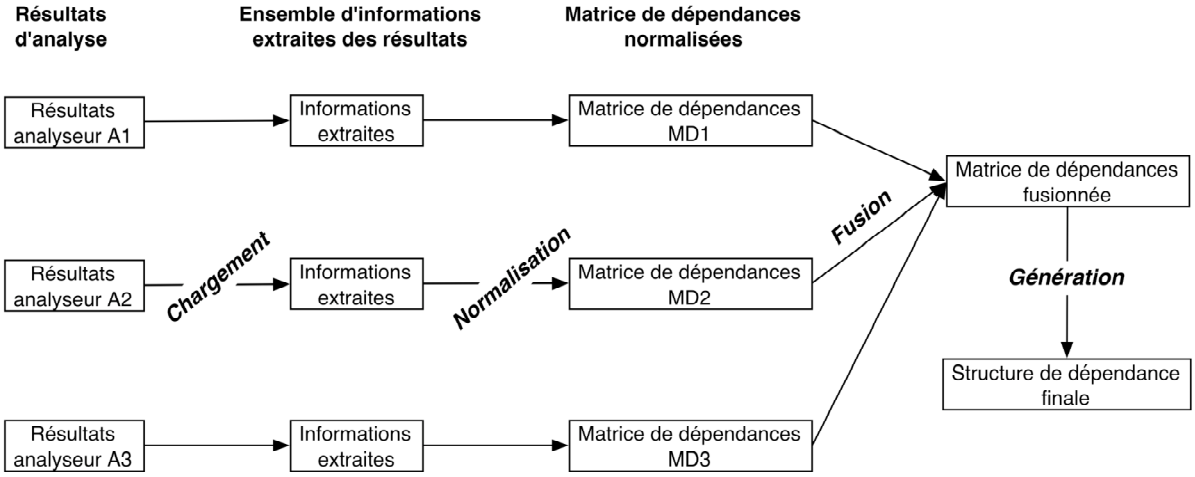


Figure 2 : Architecture fonctionnelle de l'outil d'analyse

3.1 Chargement⊡récupération de l'information

Le module de chargement est un module capable de charger n'importe quels types de résultats. Nous prendrons l'hypothèse que les analyseurs linguistiques se répartissent la plupart du temps en trois catégories en fonction des résultats qu'ils produisent (voir 2.1 Analyseurs)⊡

- Les analyseurs fondés sur les constituants.
- Les analyseurs fondés sur les dépendances.
- Les analyseurs fondés sur les constituants et les dépendances.

Dans un premier temps, il faut être capable de décrire le format de représentation des résultats de l'analyseur, dans notre cas nous décrivons ce format sous la forme d'une grammaire JAVACC (un générateur d'analyseurs syntaxiques pour JAVA). Puis à l'intérieur de cette grammaire, nous introduisons des méthodes de chargement en fonction de la catégorie du résultat.

- Soit une méthode permettant de charger des relations de dépendance.
  - **AjoutRelation**(TypeRelation, Mot1, Mot2).
- Soit une méthodes permettant de charger des constituants.
  - **AjoutSyntagme**(TypeSyntagme, Ensemble de mots ou de syntagmes).

Nous devrons également avoir une méthode de chargement de l'information associée à un mot.

- **AjoutCatégorie**(Mot, Information).
- **AjoutCatégorie** (Mot, Information, Valeur).

3.2 Normalisation de l'information

Après avoir extrait toutes les informations des résultats d'un analyseur, il nous faut encore traiter toutes ces données pour les faire correspondre à notre représentation. Nous utilisons une représentation matricielle qui nous semble plus adéquate pour représenter un ensemble de relations. Ce type de représentation permet de manipuler et fusionner facilement des données.

Pour normaliser toutes ces informations extraites de chaque analyseur, nous nous aidons du tableau de classement (voir 2.2 Étude). Ce tableau fait correspondre à chaque donnée extraite son information normalisée. Par exemple, pour le résultat fourni par Xelda, la relation SUBJ sera normalisée pour correspondre à la relation SUJET et ainsi de suite pour toutes les relations et les catégories simples à normaliser. Mais ce tableau devra également contenir, pour certaines informations, des renseignements supplémentaires concernant la normalisation de celle-ci. Par exemple, la relation SUBINV (sujet inversé) extraite de l'analyse de Xelda pourra être transformée en une relation sujet grâce à la règle normalisation.

```
SUBINV($var1[1], $var2) ::= SUJET::xelda($var1, $var2)
```

D'autres relations posent des problèmes plus difficiles par exemple la relation VMODOBJ extraite de l'analyse de Xelda pourrait avoir plusieurs significations syntaxiques. complément circonstanciel (voir Exemple 1 : Résultat Xelda filtré de la phrase "Pierre travaille dans son bureau."), complément d'objet direct, etc. L'idée est d'associer à chaque information linguistique pouvant être extraite un indice de confiance[2].

```
VMODOBJ($var1, $prep, $var2) ::= COMPLEMENT_CIRCONSTANCIEL($var1, $var2)
                                 indice_de_confiance = 0,8

VMODOBJ($var1, $prep, $var2) ::= COMPLEMENT_OBJET_DIRECT($var1, $var2)
                                 indice_de_confiance = 0,2
```

[1] La variable $var représente soit un mot soit un syntagme, . la variable $prep représente un mot.

[2] L'indice varie de 0 à 1, 1 étant l'indice de confiance maximum.

Cet indice sera déterminé à l'aide d'un apprentissage réalisé à partir d'un corpus de référence[3]. Cet indice exprime la confiance relative à l'information en fonction de l'analyseur et de la typologie de l'énoncé. Cet indice sera recalculé lors de la fusion en fonction des autres indices associés à la même information linguistique fournis par les autres analyseurs et du nombre d'analyseurs pouvant fournir ce type d'information (voir 3.3 Fusion des informations linguistiques). L'indice ainsi recalculé permettra lors de la génération de déterminer si l'information correspondante doit être conservée ou supprimée (voir 3.4 Génération des structures de dépendance).

La méthode explicitée ci-dessus est basée sur la méthode dite de « vote à la majorité » (plus une information sera commune aux différents analyseurs, plus son indice de confiance augmentera). Chaque indice pourra être vu comme le *vote pondéré* de l'analyseur pour l'information, ce vote étant *adapté* aux différentes possibilités de l'analyseur en fonction de l'énoncé (par exemple, entrée bruitée ou non) lors de l'apprentissage.

3.3 Fusion des informations linguistiques

À ce moment de l'analyse, à chaque résultat fourni par un analyseur est associée une représentation matricielle comprenant toutes les informations linguistiques normalisées extraites. Dans ce module, nous fusionnons toutes ces représentations pour obtenir une représentation matricielle finale contenant toutes les informations extraites de chaque analyseur. Certaines informations linguistiques seront complémentaires, d'autres contradictoires. Il ne s'agit pas simplement de regrouper toutes les informations, il faut également calculer de nouveaux indices de confiance pour chaque information en fonction des indices de confiance fournis par le module de normalisation.

Définition du calcul du nouvel indice associé à l'information i :

$$\underset{\text{Fusion}}{\text{indice(i)}} = \frac{\sum \text{indice(i)}}{\text{Nombre d'analyseurs pouvant fournir l'information i}}$$

Par exemple, calculons le nouvel indice de confiance à associer à l'information : relation SUJET entre les mots x et y, fournie à la fois par l'analyseur de Xelda et par l'analyseur de Vergne. L'indice de fusion associé à SUJET(x,y) est égal à la somme des deux indices de confiance indice(SUJET::xelda)=0,5 et indice(SUJET::vergne)=0,7 divisé par le nombre d'analyseurs pouvant fournir ce type d'information (ici trois), (0,5+0,7+0)/3 = 0,4. Si le troisième analyseur fournit également une information de type SUJET entre les mots x et z, et que l'indice de confiance relatif à cette information est de 0,8. L'indice de fusion associé à SUJET(x,z) est égal à (0+0+0,8)/3 = 0,26.

On peut constater que notre calcul favorise les informations fournies par le plus grand nombre d'analyseurs et ayant des indices élevés. Les nouveaux indices ainsi calculés serviront lors de la phase de génération.

---

[3] Le module d'apprentissage est en cours de réalisation et sera basé sur une méthode de rétro-propagation des indices. Pour le moment, les indices de confiance sont fixés manuellement.

3.4 Génération des structures de dépendance

Le dernier module permet d'obtenir une ou plusieurs structures de dépendance grâce à toutes les informations recueillies. Les structures de dépendance sont générées à partir des informations ayant un indice de confiance élevé et de contraintes linguistiques imposées par l'utilisateur pour éviter les informations contradictoires. Nous pouvons voir ce module comme un module de satisfaction de contraintes comportant 3 règles : une et une seule relation entre deux mots, les informations seront conservées si leur indice de confiance est au-dessus d'un certain seuil et le respect des contraintes linguistiques imposées par l'utilisateur. Notre outil générera donc plusieurs structures résultats pour un énoncé, à chaque structure résultat sera associé un indice de pertinence, moyenne pondérée des indices de confiance présents dans la structure.

4 Première réalisation

Nous avons réalisé une première maquette de notre outil ne comportant que les deux premiers modules : chargement et normalisation en utilisant une représentation classique en arbre pour représenter nos données. Cette maquette ne charge que deux analyseurs : l'analyseur de la plateforme Xelda et l'analyseur développé par Jacques Vergne. Cette première réalisation avait pour but de vérifier nos hypothèses de départ, c'est-à-dire : la possibilité d'extraire facilement de l'information et de la normaliser, ainsi que la possibilité d'intégrer facilement de nouveaux analyseurs à une plateforme d'analyse. Cette expérimentation nous a également permis de distinguer les futurs problèmes.

Le corpus de phrases utilisé provient d'un corpus UNL, Universal Networking Language, contenant une phrase et son hypergraphe UNL associé. Ce corpus était constitué de quarante phrases courtes, neuf mots par phrase en moyenne (minimum 3 mots, maximum 56 mots), décrivant de nombreux phénomènes linguistiques : coordination, négation, relative, etc. Par exemple :

- Une tulipe est plus belle qu'une rose.
- Il sait que tu ne viendras pas et il ne le regrette pas.

Dans un premier temps, nous avons donc traité ces quarante phrases avec l'analyseur de Xelda et avec l'analyseur développé par J. Vergne. Ce premier traitement, appliqué à toutes les phrases du corpus, nous a permis de déterminer quelques règles de normalisation à associer aux informations fournies par les analyseurs. Pour cette expérimentation, nous avons 15 règles de normalisation associées à chaque analyseur concernant essentiellement les relations syntaxiques.

Pour intégrer les deux analyseurs à notre outil, nous avons donc réalisé leurs grammaires de récupération à associer au module de chargement et leurs modules de normalisation ont été générés automatiquement en fonction des règles de normalisation associées à chaque analyseur. Notre outil a extrait toutes les informations linguistiques fournies par les résultats d'analyse. Avec seulement 15 règles de normalisation, notre outil a déterminé 78% des liaisons entre mots (relations non étiquetées) et 67% de ces relations ont pu être étiqueté. Les premiers résultats normalisés fournis pour notre outil sont encourageants, même s'ils restent limités à quelques phrases et quelques règles de normalisation, mais surtout les hypothèses de départ ont pu être vérifiées.

Mais plusieurs problèmes se sont présentés, tout d'abord, la lourdeur de la représentation arborescente utilisée. En effet, nous nous sommes vite rendu compte de l'utilité de changer de représentation pour les deux premiers modules mais surtout pour le futur module de fusion et d'adopter une représentation de type matricielle, plus maniable et plus adaptée à nos besoins (voir 3.2 Normalisation de l'information). Le second problème concerne le besoin d'informations linguistiques supplémentaires dans le cas des analyseurs fondés sur les constituants. Comment trouver les relations entre mots à partir des syntagmes si aucune information n'est pas fournie concernant le gouverneur du syntagme. Dans le cas de l'analyseur développé par J. Vergne, l'information concernant le gouverneur du syntagme est fournie mais pas dans le cas de l'analyseur de Xelda. Nous avons donc besoin d'une base de connaissances linguistiques simple permettant de résoudre ce genre de problèmes.

5 Bilan et perspectives

Cette première expérience nous a permis de vérifier la possibilité de charger des informations linguistiques extraites de résultats d'analyse fondé sur les constituants et les dépendances, mais également la possibilité de normaliser ces informations. Une seconde maquette est en cours de réalisation et sera testée sur 5 analyseurs : l'analyseur de la plateforme Xelda, l'analyseur Xip développé par Rank Xerox, l'analyseur developpé par J. Vergne, l'analyseur du projet Lidia, l'analyseur developpé par J. Chauché. Nous utilisons le corpus amaryllis composé de 140000 fichiers composé chacun d'une douzaine de phrases longues (environ une trentaine de mots). Les phrases sont bien formées et abordent de nombreuses thématiques. Ce corpus a déjà été traité par l'analyseur de la plateforme Xelda et est en cours de traitement par l'analyseur développé par J. Vergne. Cette nouvelle maquette intégrera tous les modules et fournira en résultat une ou plusieurs structures de dépendance.

Remerciements

Je tiens à remercier Rank Xerox et Jacques Vergne pour m'avoir permis d'utiliser leurs analyseurs.

Références

Adda G., Mariani J., Paroubek P., Rajman M., et Lecomte, J. (1999), L'action GRACE d'évaluation de l'assignation des parties du discours pour le Français, in revue *Langues*, Vol. 2(2) : pp 119-129.

Ait-Mokhtar S., Chanod JP. (1997), Incremental finite-state parsing, in *Applied Natural Language Processing 1997*, April 1997, Washington.

Boitet C. (1990), La TAO à Grenoble en 1990. in *École d'été de Lannion sur le Traitement Automatique des Langues Naturelles*. CENT, juillet 1990, 65 p.

Boitet C., Blanchon H. (1995), Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup. in *Machine Translation*. vol. **9(2)** : pp. 99-132.

Monceaux L., Isabelle Robba I. (2002), Les analyseurs syntaxiques : atouts pour une analyse des questions dans un système de question-réponse », Actes de *TALN'2003*, pp.195-204.

Sérasset G., Boitet Ch. (2000), On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter, in *COLING 2000*, Saarebruecken, Germany.

Vergne J., Giguet E. (1998), Regards théorique sur le « Tagging », Actes de TALN'1998, pp 24-33