Using an incremental robust parser to automatically generate semantic UNL graphs

Núria Gala

Using an incremental robust parser to automatically generate semantic UNL graphs

Nuria Gala
GETA-CLIPS-IMAG
385 av. de la Bibliothèque, BP 53
F-38041 Grenoble cedex 9, France
nuria.gala@imag.fr

Abstract

The UNL project (Universal Networking Language) proposes a standard for encoding the meaning of natural language utterances as semantic hypergraphs, intended to be used as pivot in multilingual information and communication systems. Several deconverters permit to automatically translate UNL utterances into natural languages. However, a rough enconvertion from natural language texts to UNL expressions is usually done interactively with editors specially designed for the UNL project or by hand (which is very time-consuming and difficult to extrapolate to huge amounts of data). In this paper, we address the issue of using an existing incremental robust parser as main resource to enconverting French utterances into UNL expressions.

1 Introduction

UNL is a project of multilingual personal networking communication initiated by the University of United Nations based in Tokyo. The representation of an utterance in the UNL interlingua is a hypergraph where nodes bear *universal words* (interlingual acceptions) with semantic attributes and arcs denote semantic relations. Any natural language utterance can be enconverted (encoded) into a UNL expression that can then be used as a pivot in a variety of possible applications (multilingual information retrieval, automatic translation, etc.).

Enconverting into UNL is thus to be understood as the process by which a UNL expression is generated from the analysis of a natural language utterance. This process can be carried out by different strategies, ranging from fully automatic to fully human enconverting.

Within the UNL project, a number of software tools exist for different languages, mainly dictionnaries and deconverters (for French (Serasset and Boitet, 2000), for Tamil (Dhanabalan and Geeta, 2003), etc.). However, there are a few tools for enconversion (for German (Hong and Streiter, 1999), for Spanish[1], etc.). As they are not full automatic enconverters, these systems have not yet proved to be suitable for dealing with huge amounts of heterogeneous data.

For French, there is currently a version under development of an enconverter that uses the Ariane-G5 platform (Boitet et al., 1982), an environment for multilingual machine translation, for the analysis of the natural language input. However, this approach has several drawbacks. First, the size of the linguistic input that it can process is limited to 200-250 words. Second, the output produced contains all the possible complete linguistic analysis for a sentence (multiple syntactic and logico-semantic trees). This implies an interactive disambiguation step to choose the appropriate linguistic analysis for the enconverter. Such an interactive disambiguation step is not a drawback in itself (it is indeed very useful in the context of automatic translation). The problem rather comes from an efficient disambiguation of huge amounts of analysis in a reasonable time. Finally, the system is not yet multi-platform (the program currently runs only on Macintosh) and the connecting procedures with Ariane-G5 are not very efficient at this time (efforts are currently being done to address this issue).

To cope with all these difficulties and to develop a French enconverter that can generate UNL expressions for large collections of raw corpora, we propose to use the ouputs produced by an existing incremental parser which has already proved robust and efficient for parsing huge amounts of data.

This article is organized as follows: after introducing the UNL language and giving some details on how it represents knowledge in a language-neutral way, we present XIP, an in-

[1] http://www.unl.fi.upm.es

cremental parser, that we will use as the central tool for the enconversion. Then, we describe the mechanism for transforming XIP's outputs into UNL expressions and finally we discuss a preliminary evaluation of the enconverter and our perspectives.

2 The Universal Networking Language (UNL)

2.1 The language

UNL is an artificial language that describes semantic networks. Sentence information is represented by hypergraphs having universal words (UWs) as nodes and relations as arcs. A hypergraph can also be represented as a set of directed binary relations, between UWs in the sentence. Linguistic information is encoded by means of the UWs, the relations that exist between them and the attributes that are associated with them.

2.2 Universal Words

Universal Words represent simple or compound concepts. They denote interlingual acceptions (word senses) for a given lemma.

An entry in the dictionnary of Universal Words contains, as illustrated in Figure 1, a head word (the French lemma "membre" in this example) followed by a list of morpho-syntactic constraints. The last part of the entry contains the UW itself: a character string (an English-language lemma) between double quotes, which usually contains a list of semantic constraints in brackets.

```
[membre] { CAT(CATN),GNR(MAS,FEM),N(NC) }
"associate(icl>member)";
[membre] { CAT(CATN),GNR(MAS),N(NC) }
"member(icl>human)";
[membre] { CAT(CATN),GNR(MAS) }
"member(icl>part)";
[membre] { CAT(CATN),GNR(MAS),N(NC) }
"member";
```

Figure 1: Semantic ambiguity in Universal Words.

When present, the list of semantic constraints describes conceptual restrictions. For example, the first three entries in Figure 1 define three different acceptions while the last one provides only the lemma and is thus more general.

2.3 Relations

Binary relations are the building blocks for UNL expressions. They link together two UWs in a linguistic utterance and have labels that depend on the roles the UWs play in the sentence. A UNL relation is represented by a headword (the label of the semantic relation) followed by a bracketed expression containing the UWs. The UWs are separated by a comma and decorated with different kinds of linguistic information.

Figure 2 shows the UNL enconversion for the following French sentence:

"Lors de la 29e session de la Conférence générale de l'Unesco, les 186 Etats membres ont ratifié à l'unanimité ce projet."[2]

```
agt(ratify(agt>thing,obj>thing).@entry.@past,
state(icl>nation).@def.@pl)
mod(state(icl>nation).@def.@pl,
member(mod<thing))
qua(state(icl>nation).@def.@pl,186)
man(ratify(agt>thing,obj>thing).@entry.@past,
unanimously(icl>how))
obj(ratify(agt>thing,obj>thing).@entry.@past,
project(icl>plan(icl>thing)))
mod(project(icl>plan(icl>thing)),this)
tim(ratify(agt>thing,obj>thing).@entry.@past,
session(icl>meeting).@def)
mod(session(icl>meeting).@def,29.@ordinal)
mod(session(icl>meeting).@def,
conference(icl>meeting).@def)
mod(conference(icl>meeting).@def,
general(mod<thing))
mod(conference(icl>meeting).@def,UNESCO.@def)
```

Figure 2: UNL expressions.

The UNL expressions in Figure 2 encode relations such as `agt` (agent), `qua` (quantifier), `mod` (modifier), `tim` (instant time), `man` (manner) and `obj` (object).

As can be seen on the figure, the information given by a UNL relation may be very semantically precise : for example, the notion of "time" is composed of six labels, corresponding to an instant time (`tim`), an initial time (`tmf`), a final time (`tmt`), a period (`dur`), a sequence (`seq`) or a simultaneous action (`coo`).

[2] In their 29th General Conference, the 186 member states of the Unesco ratified their unanimous support of the project.

The couple of UWs present in a relation have different kinds of attributes : morphological information (`def`, `pl`, etc.), information about tense (`past`), etc.

2.4 Representation of UNL graphs

The list of UNL relations for a linguistic utterance is represented by a UNL hypergraph (a graph where a node is simple or recursively contains a hypergraph). The arcs bear semantic relation labels and the nodes are UWs with their attributes as showed in Figure 3.

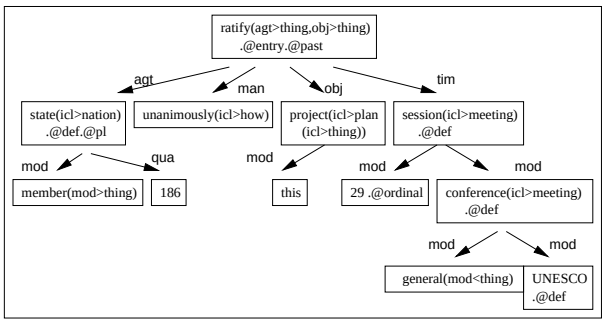

Figure 3: UNL Hypergraph.

UNL hypergraphs must contain one special node, called the entry of the graph (usually the finite verb). This information is encoded with the label `entry` in the list of UNL relations representing the corresponding hypergraph.

3 An incremental robust parser

3.1 Overview of the parser

XIP (Ait-Mokhtar et al., 2002; Hagege and Roux, 2002) is a rule-based platform for building robust incremental parsers. It is developped at the Xerox Research Centre Europe (XRCE) and shares the same computationnal paradigm as the PNLPL approach (Jensen, 1992) and the FDGP approach (Tapanainen and Jarvinen, 1997).

At present, various grammars for XIP have been built for English and French. The different phases of linguistic processing are organized incrementally : syntactic analysis is done by first chunking (Abney, 1991) a morphosyntactic annotated input text and then extracting functionnal dependencies (links between the words).

The aim of the system is to produce a list of syntactic dependencies which may be later used in applications such as information retrieval, semantic disambiguation, coreference resolution, etc.

3.2 Incremental approach

A XIP parser, like the French parser (that we will call XIPF hereafter), is composed of different modules that transform and process incrementally the linguistic information given as input. XIPF contains three main modules: one for morphological disambiguation (disambiguation of POS tags depending on contextual information), another one for chunking (marking structural groups) and a last one for dependency calculus (identifying links between words).

Each module may have a number of grammars which are applied one after the other depending on the linguistic complexity of the phenomena present. For example, for French, the identification of verbal phrases comes after the identification of nominal phrases. The different rules in the grammars also apply incrementally. They are organized in levels so that they apply sequentially to enrich stepwise the linguistic analysis. This strategy favors linguistic precision over recall.

3.3 Data representation

Within the XIP formalism, information is represented by means of syntactic trees with terminal nodes or sequences of constituant nodes (such as nominal phrases (`NP`s), finite verbal phrases (`FV`s), etc.). The maximal node for each tree (sentence) is a virtual node called `GROUPE`.

All nodes, lexical (*membre*) or not (`NP`), have a list of features associated with them and describing precise features : typographical (capital letter `[maj:+]`), lexical (proper noun `[proper:+]`), morphological (number `[plu:+]`), syntactic (subcategorization with the preposition "a" `[sfa:+]`) or semantic (time `[tim:+]`).

Since the complete linguistic information of a node is always present, even if it is not displayed in the output, it is simple to manipulate at any time during the analysis. Therefore, the possibility of taking into account different kinds of features at any step of the analysis is a considerable advantage when building a semantic application (the enconversion into UNL expressions).

Indeed, semantic information can be enriched by adding new particular features when necessary (a feature `title` has been added to be applied in titles).

3.4 XIPF output

The final result of the parser (a list of syntactic dependencies) is obtained from the linguistic

processing done by the different modules. Figure 4 shows the XIPF analysis for the French sentence given as example in section 2.3.

```
 SUBJ_NOUN(ratifié,Etats)
VARG_NOUN_DIR(ratifié,projet)
VMOD_LEFT_NOUN_INDIR(ratifié,Lors
de,session)
VMOD_POSIT1_ADV(ratifié,à, l'unanimité)
NMOD_POSIT1_RIGHT_ADJ(Conférence,générale)
NMOD_POSIT1_LEFT_ADJ_NOUN(session,29e)
NMOD_POSIT1_NOUN(session,de,Conférence)
NMOD_POSIT1_NOUN(Conférence,de,Unesco)
NN(Etats,membres)
DETERM_DEF_NOUN_DET(la,session)
DETERM_DEF_NOUN_DET(la,Conférence)
DETERM_DEF_NOUN_DET(l',Unesco)
DETERM_DEF_NOUN_DET(les,Etats)
DETERM_DEM_NOUN_DET(ce,projet)
DETERM_NUM_NOUN(186,Etats)
AUXIL(ratifié,ont)

 0>GROUPE{SC{PP{Lors de NP{la AP{29e}
session}} PP{de NP{la Conférence}}
AP{générale} PP{de NP{l' Unesco}} , NP{les
186 Etats} NP{membres} FV{ont ratifié}} à
l'unanimité NP{ce projet} .}
```

Figure 4: XIPF output.

For this sentence, the parser extracts relations such as subject (SUBJ), verbal subcategorization (VARG), verbal and nominal modification (VMOD, NMOD and NN), determination (DETERM) and verbal auxiliary (AUXIL). The head of the dependency appears as the first element except in the case of a determination relation.

Relations usually have a list of morpho-syntactic features associated with them : the POS tag of the word linked to the head (NOUN in a SUBJ relation, ADJ in a NMOD, etc.), morphological precisions (NUM, DEM) or syntactic features (the position of the adjective, POSIT1, RIGHT, etc.).

The process of dependency extraction is deterministic: the most plausible relation according to the system is extracted. The only exception is that of prepositionnal attachment (VMOD and NMOD): the linguistic information that the parser has is not enough to handle structural ambiguities. In this case, all possible relations appear in the result.

3.5 Parser evaluation

Parsers built with the XIP engine (XIPF) are able to process about 2.000 words/s using 10 Mo of memory footprint (only grammars, without lexicons)[3] .

As for linguistic performance, an evaluation of XIPF subject and object (VARG) dependencies, conducted on French newspapers (Ait-Mokhtar et al., 2001), showed the following precision (P) and recall (R) rates: SUBJECT, P = 93,45 %, R = 89,36 %; OBJECT, P = 90,62 %, R = 86,56 %.

Another evaluation carried out with XIPF+ (Gala, 2003), a second French parser containing more specialized grammars to handle complex phenomena such as punctuation, lists, titles etc., using varied raw corpora from different types and domains[4] gives P = 94 % for subject even in sentences being or containing lists, enumerations etc. and P = 93 % and R = 89,6 % for key words in titles (CLE relation).

4 A French UNL enconverter

4.1 Overview

The principal motivation to create a French enconverter is to easily obtain huge amounts of UNL enconverted corpora which can be subsequently used in other applications (for example, multilingual information retrieval). To achieve this objective, one of the main requirements was also the reusability of existing robust linguistic resources.

The choice of a XIP parser was motivated by several reasons. First, its robustness permits to deal with huge amounts of text (a result is always produced whatever the complexity of the input). Second, its modular architecture facilitates the articulation of different ressources (it is easy to enrich the parser with new lexicons and grammars and to desactivate a particular module when necessary). Finally, the flexibility of the formalism permits to enrich the rules and the features with no harm. We have prefered XIPF+ over the standard XIPF because of its broader linguistic coverage.

The French UNL enconverter is thus a processor that automatically transforms annotations

[3]Obtained with a Pentum III 1 GHz.

[4]About 108.000 words extracted from the Web (end 2000) concerning general newspaper (*Le Monde*) as well as specialized domains such as economics (journal *Les Echos*), science (medecine, physics), law (project of law), etc.

provided by the XIPF+ parser into UNL expressions.

4.2 Remarks on terminology

To avoid ambiguity, we use the term "dependency" to indicate XIPF+ syntactic links of the form `D(x,y)` or `D(x,y,z)`, as shown in Figure 4, and the term "feature" to indicate linguistic information provided by the parser. XIPF+ provides twelve types of dependencies and more than two hundred and fifty features, of the types described in section 3.3 (typographical, morphological, etc.).

As for UNL, we use the term "relation" to denote a semantic link of the form `label(UW1.attributes,UW2.attributes)`, as shown in Figure 2, while an "attribute" corresponds to a UNL annotation. Such an annotation appears to the right of a UW and adds particular linguistic information. The UNL formalism provides about fourty relations and eighty attributes of different types.

4.3 Generation of UNL expressions

The first step of the enconvertion consists in identifying the information provided by XIPF+ that will be translated into UNL relations. There are three kinds of mapping rules performing this task, depending on the input and the result of the transformation: a dependency giving an attribute, a dependency giving a relation, a feature giving a relation.

4.3.1 Dependency to attribute.

The first kind of mapping rules transforms a XIPF+ dependency into a UNL attribute. An example is that of the relation `CLE` (the head of a title). Within UNL it becomes `@title` and it is included as an attribute of the UNL relation containing the head word of the title.

The following example describes a title, its analysis with XIPF+ and its UNL encoversion:

Le Forum Universel des Cultures[5]

```
 CLE(Forum)
NMOD_POSIT1_RIGHT_ADJ(Forum,Universel)
NMOD_POSIT1_NOUN_INDIR(Forum,des,Cultures)
DETERM_CLOSED_DEF_NOUN_DET(Le,Forum)

<title> 0>GROUPE{NP{Le Forum} AP{Universel}
PP{des NP{Cultures}} </title>

 mod(forum.@def.@entry.@title,
universal(mod<thing))
mod(forum.@def.@entry.@title,
culture(icl>abstract thing).@def.@pl)
```

Figure 5: Example of dependency to attribute transformation.

4.3.2 Dependency to relation.

The second kind of mapping rules transforms a XIPF+ dependency into a UNL relation. In some cases, this transformation is not straightforward since a number of lexical and semantic features are to be taken into account (and they are not always provided by the parser). This is the case of dependencies with the verb to be and generally with all verbs denoting a state.

While in the UNL formalism the verb *to be* is considered a copula and does not appear in the semantic representation, the parser produces the syntactic dependencies in which the verb participates and marks the fact of being a copula by means of features (`[copula]` as lexical feature and `SPRED` -predicative- as syntactic feature) as illustrated on the example below :

Le Forum est Universel.[6]

```
SUBJ(être,Forum)
VARG_ADJ_SPRED(être,Universel)

aoj(Universal,Forum)
```

Figure 6: Example of dependencies involving the verb *to be* and their corresponding UNL relation.

In this case, an `aoj` relation shows the link between the noun in the subject and the adjective. The parser's feature permitting the identification of a copula is thus crucial in order to map precisely a `SUBJ` and a `VARG` into an `agt` and a `obj` or into a single `aoj`.

Table 1 gives a summary of the principal transformations performed by this second kind of mapping rules (as it is shown, in the case of modification, two types of XIPF+ relations produce a UNL `mod`):

[5]The Universal Forum of Cultures

[6]The Forum is Universal.

SUBJ(X[be-],Y)	agt(X,Y)
VARG(X[be-],Y)	obj(X,Y)
SUBJ(X[be+],Y) VARG(X[be+],Z)	aoj(Z,Y)
NMOD(X,Y) or NN(X,Y)	mod(X,Y)

Table 1: XIPF+ dependencies producing UNL relations.

4.3.3 Feature to relation.

The last type of mapping rules identifies particular information encoded as features within the parser's output and transforms them into UNL relations with the appropriate words. This is the case for the notions of quantification and time.

Regarding quantification, this feature, encoded within the dependency `DETERM`, is transformed to produce a `qua` UNL relation between a determiner and a noun.

As for the relations involving the notion of time, the feature `time` encoded by XIPF+ is too general. Therefore, it is not possible to produce the semantically precise UNL relations expressing variations of the concept of time (duration, final time, sequence, etc.). In this case, we have chosen to create an intermediate UNL relation named `time` in order to keep this semantic information.

NMOD(X,Y,Z) DETERM(W[quant+],Z)	qua(Z,W)
NMOD(X,Y,Z[time+])	time(X,Z)

Table 2: XIPF+ features producing UNL relations.

4.4 Accessing the UW base

After identifying the UNL relations, the enconverter retrieves the UWs corresponding to each French word in a relation. UWs are contained into a UW database of 37.901 French lemmas.

The major difficulty here concerns ambiguity, that is, accessing the right acception, since the database usually contains a list of UWs for a given lemma. The ambiguity can be semantic, when a French lemma corresponds to a single English lemma with different acceptions (*cf* Figure 1) or lexical, when a French lemma corresponds to several English lemmas. Here is an example of lexical ambiguity with the pronoun *il* ("he" or "it" in English) :

```
[il] { CAT(CATR) } "he(icl>human)";
[il] { CAT(CATR) } "it(icl>nonhuman)";
```

Figure 7: Lexical ambiguity in Universal Words.

To this date, as the lexico-semantic information provided by the parser is not enough to choose the appropriate UW, the enconverter takes the most general acception (that is, the word sense without a constraint list –the last entry in the list showed in Figure 1). When all acceptions of an entry have such list of constraints, the enconverter chooses the first one.

4.5 Enrichment with lexical information

The final step of the enconversion enriches the rough UNL expressions produced (UNL labels with simplified UWs) with more complete morphological information. A set of rules is thus specialized in translating different linguistic features from the parser into UNL descriptors completing the words in a relation.

Some of this morphological information can also be extracted from the UW base (gender). However, we have preferred to extract a maximum of information from the parser because it produces a contextual analysis of the words appearing in a linguistic utterance.

The features which enrich the UNL output concern definiteness (`@def` or `@indef`), number (`@sg` or `@pl`) and tense (`@past`, `@present`, `@fut`). A few labels (`@ordinal`, `@complete` ...) are absent on the XIPF+ output and therefore not automatically enconverted in the UNL output. Finally, the attribute `@entry` is systematically added to UWs head of their sentence (the verb): `agt`, `varg`, `aoj`, etc.

5 Evaluation

A complete evaluation of a UNL enconverter should take into account the following possible kinds of errors:

- graphs with wrong linguistic information (semantic relations, attributes, etc.),
- missing information (incomplete graph due to missing relations, incomplete decorations, etc.),
- graphs with wrong UWs (wrong acception or wrong lemma).

Since in this article we want to emphasize the use of an incremental robust parser for creating an enconverter, we evaluated errors concerning semantic relations[7], thus the first and the second points which correspond, respectively, to classic evaluation metrics of precision and recall rates.

The enconverter was tested against the first 50 manually enconverted UNL graphs (1.059 words) from a corpus of legal text. The average length of the sentences was about 21 words (21,18). The semantic relations evaluated in this preliminary experiment (322 UNL expressions) were `agt` (44), `obj` (57) and `mod` (221).

Table 3 gives the results obtained for the evaluation of this first version of the enconverter :

Relation	Precision	Recall
`agt(X,Y)`	57 %	80 %
`obj(X,Y)`	51 %	48 %
`mod(X,Y)`	58 %	86 %

Table 3: Results of the evaluation of the first version of the enconverter.

For agents, most errors come from syntactic subjects correctly identified by the parser but presenting semantic features that should had been taken into account to create `aoj` relations. To give an example, in the sentence *"La culture acquiert des formes différentes (...)"*[8], the parser extracts correctly the dependency `subj(acquire,culture)` although it is semantically encoded as `aoj(acquire,culture)` in UNL because the verb "acquire" is considered in this utterance as a verb of state.

In the case of objects, errors on precision concern wrong scope of coordination as well as objects being a whole sentence. As for recall, there are several constructions which may be considered `obj` from a semantic point of view but that the parser identifies as modifiers due to their surface construction with a preposition. For example, *"source de creativité"*[9] is analyzed by the parser as `mod(source,de,créativité)` although UNL encodes `obj(source,creativity)`. Likewise, *"vers l'accés de la diversité culturelle"*[10] is encoded in UNL as `obj(towards,access)`, a kind of relation that the parser does not extract.

A final remark concerns modifiers. As said before, the parser is not deterministic in marking modifiers: all possible combinations between a head word and its dependents are extracted. That is the main reason why precision is low and recall is high.

The average of all these figures gives a global evaluation of the enconverter corresponding to P = 56 % and R = 69 %.

6 Discussion

At this stage of the project, there are a number of conclusions we can draw from the preceding evaluation.

The first one is that the results are rather encouraging in terms of a first rough enconvertion from syntactic XIPF+ information to UNL expressions (`agt`, `obj` and `mod`). However, we are aware that certain cases present considerable difficulties. For example, in addition to the examples presented in the evaluation for verbs of state, subjects with a semantic feature of "patient" are to be enconverted as `obj` and not as `subj` (unfortunately the semantic information needed for this transformation is not yet available within the parser). Thus in *"La réunion continuera jusqu'à ce soir."*[11] the parser extracts a `subj(continuer,réunion)` that might be enconverted as `obj(continue,meeting)` in UNL. All these kinds of complex transformations including particular semantic features are at this point an important bottleneck for the enconverter.

The second conclusion coming from the evaluation (even if not quantitatively analyzed) is that the choice of the UW remains a critical point, as the enconverter has not the possibility of choosing the correct acception giving a configuration. One possibility to consider might be to introduce interactivity with a human to choose the correct UW. The second possibility is related to the improval of the parser: we can consider adding more linguistic information, in the form of semantic classes or semantic features, in order to be able to disambiguate. Having enriched the parser with these semantic features, another possibility to improve the enconverter might be to consider statistical information about collocations.

[7]The presence or absence of the different attributes was not evaluated.

[8]Culture acquires different forms (...).

[9]Source of creativity.

[10]Towards accessing cultural diversity.

[11]The meeting will continue until this evening.

Finally, we are conscious that there would still remain several aspects which would demand to be improved within the parser itself : prepositionnal attachment disambiguation, scope of coordination, complex coreference, etc. Particular strategies may be adapted to handle such difficulties individually (using statistical information, interactive disambiguation, etc.).

7 Conclusion

In this paper we have presented a mechanism for automatically producing UNL expressions using the ouput of a robust parser. After describing the UNL formalism and presenting an incremental parser able to accurately process huge amounts of data, we have shown how one can transform the linguistic information provided by the parser into UNL expressions. We have also presented a first evaluation in an attempt to try to assess the performance of the enconverter.

Our results show that there are still several crucial problems that we need to solve. However, taking into account that this is preliminary work, the results already obtained are encouraging and confirm the possibility of using the reliable linguistic information automatically obtained from an incremental robust parser to create a UNL semantic enconverter for huge amounts of data.

Acknowledgements

The author wants to express her gratitude to E. Blanc and A. Max, as well as to the three anonymous reviewers, for their suggestions and comments on a first draft of the paper.

References

S. Abney. 1991. Parsing by chunks. In *Principle-Based Parsing*, edited by R. Berwick, S. Abney and C. Tenny, 257–278. Kluwer Academic Publishers. Boston.

S. Aït-Mokhtar, J. P. Chanod, and C. Roux. 2001. A multi-input dependency parser. In *Proceedings of International Workshop on Parsing Technologies, IWPT-2001*, 201-204. Beijing, China.

S. Aït-Mokhtar, J. P. Chanod, and C. Roux. 2002. Robustness beyond shallowness : Incremental Deep Parsing. *Special Issue of the Natural Language Engineering Journal on Robust Methods in Analysis of Natural Language Data*, vol. 8(3), 121–144, Cambridge University Press.

C. Boitet, P. Guillaume, and M. Quézel-Ambrunaz. 1982. ARIANE-78, an integrated environment for automated translation and human revision. In *Proceedings of Conference on Computational Linguistics, COLING-82*, 19–27, Prague.

T. Dhanabalan, and T. V. Geetha. 2003. UNL Deconverter for Tamil. In *International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies*. Alexandria, Egypt.

N. Gala. Un modèle d'analyseur syntaxique robuste fondé sur la modularité et la lexicalisation de ses grammaires. *Thèse de doctorat*, Université de Paris-Sud, UFR scientifique d'Orsay, France.

C. Hagège, and C. Roux. 2002. A Robust and Flexible Platform for Dependency Extraction. In *Proceedings of 3rd Conference on Language Ressources and Evaluation, LREC-2002*, 520–523, Las Palmas de Gran Canaria, Spain.

M. Hong, and O. Streiter. 1999. Overcoming the language barriers in the Web : the UNL-Approach. In *Multilingual Corpora : encoding, structuring, analysis. 11th Annual Meeting of the German Society for Computational Linguistics and Language Technologies.*. Frankfurt, Germany.

K. Jensen. 1992. PEG: the PLNLP English Grammar. In *Natural Language Processing : the PLNLP approach*, edited by K. Jensen, G. Heidorn and S. Richardson. Kluwer Academic Publishers. Boston.

G. Serasset, and C. Boitet. 2000. On UNL as the future "html of the linguistic content" and the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter. In *Proceedings of Conference on Computational Linguistics, COLING-2000*. Saarbruecken.

P. Tapanainen, and T. Jarvinen. 1997. A non-projective dependency parser. In *Proceedings of Conference on Applied Natural Language Processing, ANLP-97*. Washington.