Can we parse without tagging?

C. Fairon, S. Paumier and P. Watrin

Center for Natural Language Processing – CENTAL – University of Louvain
{fairon,paumier,watrin}@tedm.ucl.ac.be

Abstract

Syntactic parsing is a major area of NLP which has been widely studied with the help of many approaches. Usually, parsers take in input tagged texts, that is to say texts whose lexical units have been annotated with informations such as lemma, grammatical code, gender and number. In this paper, we present a parsing method that can work on untagged texts as well as on tagged ones. We then compare results obtained on specialized texts in their raw and tagged version in order to determine if tagging is absolutely necessary.

Introduction

Syntactic parsing is a major problem in Natural Language Processing and has been widely studied with the help of different methods, such as statistical parsing (Charniak, 1997; Collins, 1996; Magerman, 1995) and linguistic-based methods (Roche, 1993; Abney, 1996; Koskenniemi et *al.*, 1992; Ait-Mokhtar & Chanod, 1997). Traditionally, at least for Romance languages, parsers take in input tagged texts in which each lexical unit is given to grammatical and/or inflectional information, such as lemma, grammatical category, gender and number. There are many taggers that use a mix of probabilistic and linguistic information: (Church, 1988), TreeTagger (Schmidt, 1994), Brill's Tagger (Brill, 1994), (Cutting et *al.*, 1992), Xerox part-of-speech tagger, GREYC parser (Giguet & Vergne, 1997), etc. Most of these tag about 95% of all lexical units correctly. This means that there is always a certain percentage of eroneously tagged units which will disrupt the behavior of parsers.

In this paper we will discuss the possibility of parsing texts without prior tagging. Therefore we will apply the same parsing method to two versions of the same text: one which was first disambiguated by automatic tagging and one which was not (in the latter case, a dictionary provides morphological information, but without disambiguation). Results will be compared in order to measure the impact of disambiguation. The method used is a common approach based on precise linguistic descriptions: electronic dictionaries (Silberztein, 1993) and lexicon-grammar tables (Gross, 1975).

The framework

The question of whether or not tagging is needed for parsing arose within the framework of an applied research project which aims at developing a linguistic index engine[1]. In this particular context, indexation is viewed as the ultimate step towards an information extraction process. Parsing is therefore understood in a restrictive way: it is used on texts that resort to a technical sublanguage (as opposed to general language) and is used to analyse sentences that contain predefined verbs[2] in order to extract their various complements. We built extraction graphs wich enable us to correctly extract the syntactic structures described in the linguistic databases of the lexicon-grammar. The following question then arose: should these extraction graphs be applied to a disambiguated text (where each word of the text is associated with one - and only one – POS tag) or could the system rely on a “simple” dictionary lookup procedure that would provides all the possible analyses for any given word but which would offer no disambiguation.

Before discussing these two possibilities, we will first describe the parsing method and the linguistic resources format.

The parsing method

Contrary to many others parsers, our method is based on an exact linguistic description and does not use statistics of any kind. The main part of this description consists of tables that, for each verb, describe the elementary structures in which it can appear. We can automatically generate grammars from such tables, and subsequently, apply them to texts via the advanced pattern matching function offered by the Unitex[3] software. This approach has already been illustrated by several works (Roche, 1993; Senellart, 1999; Paumier, 2003).

Lexicon-grammar tables

Lexicon-grammar tables where introduced by Maurice Gross who analysed completive verbs in French (Gross 1975). These tables have since then been extended to the rest of French simple verbs (Boons et *al.*, 1976; Guillet & Leclère, 1992) as well as to other kind of predicates (nouns, adjectives) and various languages.

The tables consist of a formal description of verbs represented as a matrix, as shown in Figure 1. Each line corresponds to a verb, and each row represents a formal syntactic property[4]. A '+' sign in a cell,

[1] This research was partly financed by Bureau Van Dijk. We wish to thank them for their collaboration.

[2] The full process involves also the analysis of nominal forms. As we do not elaborate on this part in this paper, we will not give more details.

[3] Unitex is an LGPL-licensed software, available at: http://igm.univ-mlv.fr/~unitex/

[4] Usually, a lexicon-grammar table contains verbs that have the same main structure (for instance, *N0 V N1*). For convenience sake, we proceeded differently in our experiments. This is why there are properties referring to *N4* or *N5*, which is not standard in the lexicon-grammar frame. However, this detail has no

respectively '-', means that the verb can, respectively cannot, appear in the corresponding structure. For instance, the property *N0 V N1* is verified for both *alimenter* and *allumer*; we can therefore have sentences like:

N0 alimente N1
N0 allume N1

But the property *N0 V en N2* is verified for *alimenter* and not for *allumer*; as a result we can have:

N0 alimente N1 en N2

but not:

N0 allume N1 en N2

V	N0 V N1	N1 est Vpp par N0	N0 V en N2	N0 V à partir de N3	N0 V à N4	N0 V dans N5
abattre	+	+	-	-	-	-
absorber	+	+	-	-	-	-
acheminer	+	+	-	-	-	-
alimenter	+	+	+	+	-	-
allumer	+	+	-	-	-	-
approvisionner	+	+	+	-	-	-
broyer	+	+	+	-	-	-
brûler	+	+	-	-	-	-
chauffer	+	+	-	+	-	-
collecter	+	+	-	-	-	-
comporter	+	-	-	-	-	-
composer	+	+	-	-	-	-
comprendre	+	+	-	-	-	-
concentrer	+	+	-	+	-	-

Figure 1: French verb table

Parametrized graphs

We exploit the content of the table using parametrized graphs, drawn with the help of Recursive Transition Network (RTN) formalism. Such a graph describes linguistic constructions that are relevant to the lexicon-grammar table.

Each construction is described by a path in the parametrized graph as shown in Figure 2. The graph contains parameters. The value of each parameter is given depending on lexical entries and is given in the columns of the table. The parameters are named after the corresponding columns *@A, @B*, etc (*@A*=first column, *@B*=second column, etc). For Figure 1, the variable *@A* corresponds to the verb column.

A table can contain property marks ('+' and '-') and lexical elements. For each line of the table, a graph is then generated by substituting the variables as follows:

- if the variable *@X* refers to '-', the path is removed;
- if it refers to '+', the path is maintained;
- if it refers to a lexical item, it is replaced by this item.

Properties marks + and – are used as conditions in the parametrized graph. We describe a set of possible constructions in the parametrized graph which condition each construction by referring to its corresponding property in the table. An automatic process[5] then generates a graph for each entry of the table which only contains the constructions that are possible for this entry. Figure 3 shows the graph obtained for the verb *allumer*.

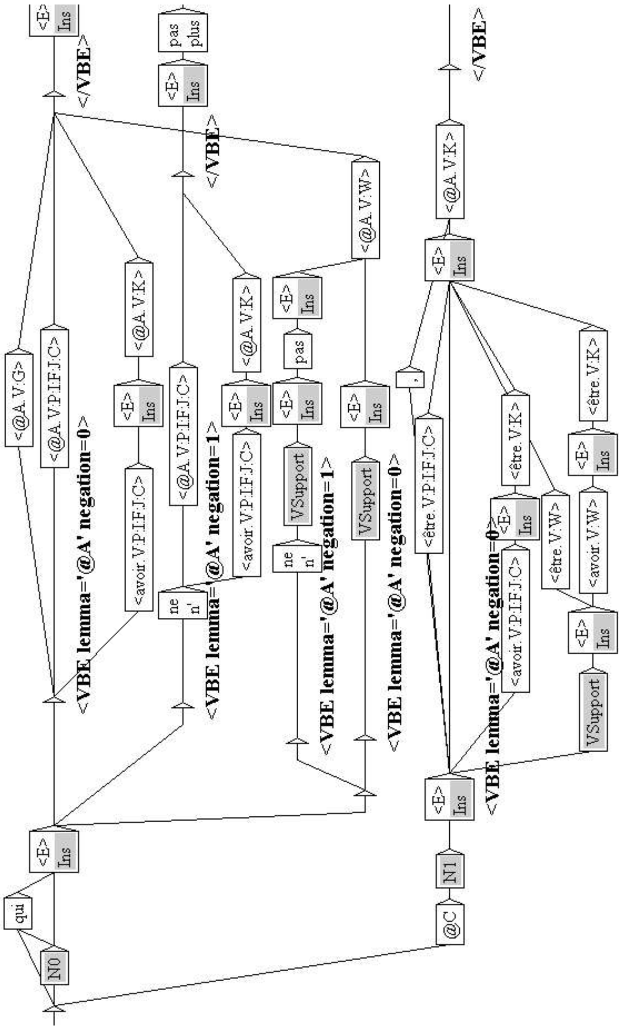

Figure 2: parametrized graph

Parsing

Our method does not distinguish between pattern matching and parsing. Once we have generated graphs, we consider them as patterns. We use the pattern matching function of Unitex to find all matching sequences in a text. If sequences are matched by a graph, then we can say that we have parsed these sequences, because we can insert outputs in the graph, and therefore, tag matching sequences.

impact on the method presented in this paper.

[5] For a complete description of this process, see Unitex manual at: http://igm.univ-mlv.fr/~unitex/manuel.html

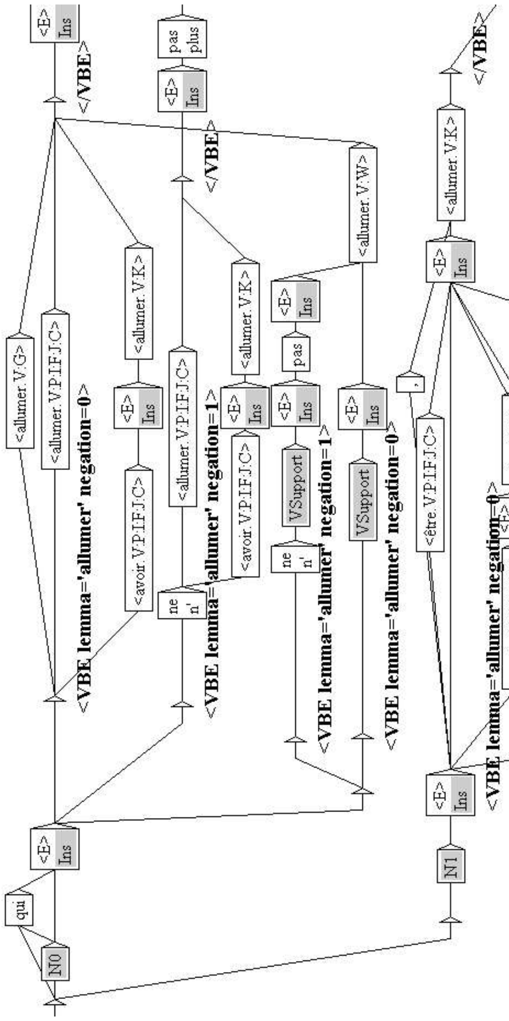

Figure 3: graph obtained for the verb *allumer*

In order to apply this method, we must define all of our patterns, which implies having a description of noun phrases. This is why we use a graph that is an approximative description of noun phrases.

The experiment

We have applied our parsing method on both tagged and untagged versions of the same corpus in order to compare results. In this section we describe our experiment.

Corpus and patterns

Our experiment aimed at extracting information in the area of energy. Our 1,500,000 word corpus is a collection of texts taken from the web. To evaluate our method we selected 5 verbs representative of the most common syntactic structures found in the domain of energy:

- alimenter :	*N0 V N1 en N2 à partir de N3*
- chauffer :	*N0 V N1 à partir de N3*
- consommer :	*N0 V N1*
- contenir :	*N0 V N1*
- produire :	*N0 V N1 à partir de N3*

Naturally, there are other syntactic structures for theses verbs, but for the purpose of our study, we were particularly interested in structures containing complements that were relevant from a semantic point of view.

Tagged text

As most famous taggers have similar precision scores, we arbitrarily chose to use the TreeTagger because it was easy to install and use in our experiment.

We first applied the TreeTagger to our corpus. A script then rewrites the output of the tagging so that it can be manipulated by Unitex. This rewriting step basically consists of changing the tagset. For example, if we consider the following text:

```
Ce rapport veut en effet alimenter le débat
social
```

The TreeTagger will turn it into:

```
Ce              PRO:DEM       ce
rapport         NOM           rapport
veut            VER:pres      vouloir
en              PRP           en
effet           NOM           effet
alimenter       VER:infi      alimenter
le              DET:ART       le
débat           NOM           débat
social          ADJ           social
```

Finally, the rewriting script will produce the following tagged text:

```
{Ce,ce.PRO} {rapport,rapport.N}
{veut,vouloir.V:P} {en,en.PREP}
{effet,effet.N} {alimenter,alimenter.V:W}
{le,le.DET+Ddef} {débat,débat.N}
{social,social.A}
```

Untagged text

By untagged text we do not mean that we use no linguistic information. We use the DELAF electronic dictionaries for French that are included in Unitex. These dictionaries provide the list of possible tags for each word. For instance, the French word *est* can be considered as a verb (a form of *to be*) or an adjective (*east*). In other words we do not assign a tag to each word *a priori*, as opposed to a tagger that always decides on a particular tag. This ambiguity is illustrated in the graph shown in Figure 4.

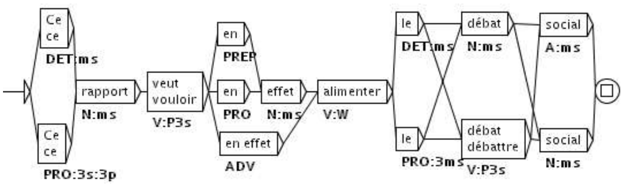

Figure 4: ambiguous text after dictionary lookup

Is this comparison relevant ?

One can object that the comparaison is irrelevant because decisions of the statistical guesser are not based on the same dictionary as the one used for the “no-tagger” approach.

This is not really a limitation since the words that are found in the TreeTagger's lexical resources are also present in the DELAF dictionary which covers a very large part of the lexicon. However there are some differences between the outputs of the TreeTagger and that of the dictionary lookup program:

– some words could be absent from our dictionary (neologisms, proper names, misspelt words) but the TreeTagger will always provide a tag for these. This point will be disscussed below in the “results” section ;
– the dictionary contains a large number of compound lexical forms (over 100.000 compound words); however the tagger cannot analyse compounds. This has no influence on the results since the noun phrase grammar we have developed contains patterns which match the internal structure of compound words (*N de N*, *N à N*, *AN*, *NA*, etc.).

Naturally, this argumentation is valid for any tagger with a comparable level of accuracy.

Results

Precision

Tagged text:	92.28% (598/648)
Untagged text:	88.67% (493/556)

The small difference in precision scores (3.61%) may indicate that tagging is not absolutely necessary. Precision errors are dued to the matching algorithm that gives priority to longest matches. For instance, the French word *est* can be an adjective (*east*) or a verb (a from of *to be*). Consequently, if the matching program considers it as an adjective, it will provide the following erroneous analysis:

```
<N0>L'air</N0> alimentant <N1>la
combustion est injecté par des buses à
la base de la chambre</N1>
```

In the example above, *est* is a verb, and the correct N1 complement is only *la combustion*. Such errors occur when the text is not disambiguated or when a word has been erroneously disambiguated.

Recall

Tagged text:	47.84 % (598/1250)
Untagged text:	39.44 % (493/1250)

The analysis of these results has highlighted two main reasons of this difference (8.4 %):

- Incompleteness of electronic dictionaries: even if they have a very large coverage, these resources cannot be exhaustive. In a domain such as energy, there are many neologisms and technical terms that are not in dictionaries. Therefore an unknown word is a fatal error with a dictionary-based approach, whereas a tagger always gives it a tag. When this tag is correct, the tagger approach is more efficient than the dictionary one.
- Numbers: in its current state, our prototype contains no graph able to handle numbers. This fact blocks the analysis in the same way as unknown words do. On the other hand, the tagger gives tags to numbers so that the grammar can deal with them if they are tagged as determiners.

These poor recall scores are mainly due to complex structures that occur in texts and that are not taken into account in our grammars. For example, verb coordination, anaphora and relatives have not been described (or very briefly). The reason for this is that, within the framekork of indexation, we gave priority to precision in order to extract relevant complements. We did not try to make a complete parsing of our corpus, but only to parse sentences that contain enough information to be useful for indexation.

Conclusion

In this paper, we have presented a parsing method which is based on a precise linguistic description and which can be applied to tagged texts as well as untagged texts. The analysis of the results obtained with and without disambiguation shows that precision is not significantly greater if we use a tagger. On the contrary, we have observed that recall is better with a tagger, because our method is currently blocked by unknown units such as neologisms and numbers.

References

Abney, S. (1996). Partial Parsing via Finite-State Cascades. In, John Carroll Editor, Workshop on Robust Parsing (ESSLLI'96) (pp. 8-15).

Ait-Mokhtar, S. and Chanod, J.-P. (1997). Incremental Finite-State Parsing. In Proceedings of the Fifth Conference on Applied Natural Language Processing.

Boons, J.-P.; Guillet, A. and Leclère, C. (1976). La structure des phrases simples en français: constructions intransitives. Droz, Genève.

Brill, E. (1994). Some Advances in Transformation-Based Part-of-Speech Tagging. In Proceedings of the Twelfth National Conference on Artificial Intelligence, volume 1 (pp. 722-727).

Charniak, E. (1997). Statistical Parsing with a Context-Free Grammar and Word Statistics. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, MIT Press.

Church, K. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In Proceedings of the Second Conference on Applied Natural Language Processing, ACL.

Collins, M. (1996). A New Statistical Parser Based on Bigram Lexical Dependencies. In Proceedings of ACL96.

Cutting, D.; Kupiec, J.; Perdersen, J. and Sibun, P. (1992). A Practical Part-of-Speech Tagger. In Proceedings of the Third Conference on Applied Natural Language Processing, ACL.

Giguet, E. and Vergne, J. (1997). From Part-of-Speech Tagging to Memory-Based Deep Syntactic Analysis. In Proceedings of IWPT-97.

Guillet, A. and Leclère, C. (1992). La structure des phrases simples en français: les constructions transitives locatives. Droz, Genève.

Koskenniemi, K.; Tapanainen, P. and Voutilainen, A. (1992). Compiling and Using Finite-State Syntactic Rules. In COLING'92 (pp. 156—162).

Magerman, D. (1995). Statistical Decision-Tree Model for Parsing. In Proceedings of the 33th Annual Meeting of the ACL.

Paumier, S. (2003). De la Reconnaissance de Formes Linguistiques à l'Analyse Syntaxique. Phd Thesis. University of Marne-la-Vallée.

Roche, E. (1993). Analyse syntaxique transformationnelle du français par transducteurs et lexique-grammaire. Phd Thesis, University of Paris 7.

Schmidt, H. (1994). Probabilistic part-of-speech tagging using decision trees. In Proceedings of the International Conference on New Methods in Language Processing (pp. 44—49), Manchester, UK.

Senellart, J. (1999). Reconnaissance automatique des entrées du lexique-grammaire des phrases figées. In Travaux de Linguistique (vol. 37, pp. 109-121). Duculot, Bruxelles.

Silberztein, M. (1993). Dictionnaires électroniques et analyse automatique de textes: le système INTEX. Masson, Paris.