Segmentation in Super-Chunks with a Finite-State Approach

Olivier Blanc[1], Matthieu Constant[2], and Patrick Watrin[2]

[1] University of Munich, CIS, Germany
[2] University of Marne-la-Vallée, IGM, France

Abstract. Since Harris' parser in the late 50s, multiword units have been progressively integrated in parsers. Nevertheless, in the most part, they are still restricted to compound words, that are more stable and less numerous. Actually, language is full of semi-fixed expressions that also form basic semantic units: semi-fixed adverbial expressions (*e.g.* time), collocations. Like compounds, the identification of these structures limits the combinatorial complexity induced by lexical ambiguity. In this paper, we detail an experiment that largely integrates these notions in a finite-state procedure of segmentation into super-chunks, preliminary to a parser. We show that the chunker, developped for French, reaches 92.9% precision and 98.7% recall. Moreover, multiword units realize 36.6% of the attachments within nominal and prepositional phrases.

1 Introduction

Since Harris' parser in the late 50s [1], multiword units have been slowly integrated in parsers [2]. Nevertheless, in the most part, they are still restricted to compound words, that are more stable and less numerous. Actually, language is full of semi-fixed expressions that also form basic semantic units: semi-fixed adverbial expressions (*e.g.* time), nominal collocations. Like compounds, the identification of these structures limits the combinatorial complexity induced by lexical ambiguity.

To study this phenomenon, we implemented an incremental finite-state *chunker* for French[3] based on the notion of *super-chunk*. Super-chunks are different from the notion traditionally associated with chunks [4–7], because adjectival and prepositional attachment has been integrated. From a formal point of view, non-recursivity is verified. Like chunks, a super-chunk stops at its head (*e.g.* the noun in a nominal chunk). Nevertheless, by taking account of multiword units (MWUs), the notion of head is extended to complex structures. For instance, *marge d'exploitation* (trading margin) and *chiffre d'affaires brut* (gross sales turnover) are tagged as a noun at the lexical analysis stage[4] and therefore

[3] Tools presented in this paper are in the most part based on the programs of the Outilex platform [3].

[4] Note that morphosyntactic information is inherited from the lexical head of the MWU (*i.e. marge* and *chiffre*). In addition, that information is augmented with the

are computed as simple words. In that case, the ambiguity reduction is obvious. By analysing the sequence *chiffres d'affaires brut* (gross sales turnover) in a compositional manner, the procedure leads to 24 analyses, that are reduced to one if the collocation is considered as a whole. Moreover, this sole lexical entry permits the resolution of a double attachment (a prepositional one and an adjectival one), which facilitates the identification of the syntactic constituents.

The chunker presented in this paper is part of a larger project of developing a complete parser for French, directly usable by real applications such as information extraction. Our system is composed of three successive stages : (1) lexical segmentation into simple and MWUs ; (2) identification and tagging of super-chunks ; (3) attachment in constituents. An illustration of this incremental procedure is given in the table 1. In this paper, we will only focus on the two first stages that are both based on finite-state resources.

We will first describe the lexical segmentation module with the description of the lexical resources used; we will show how part of them has been automatically learnt and how they have been applied to texts. Then, we will present the super-chunk segmentation module inspired by [4] and next, the disambiguation process. Finally, an evaluation of the performances of our chunker will be made and its interest for resolving lexical attachments will be shown.

2 Lexical segmentation

The lexical segmentation is a key part of our chunker. It takes as an input a text segmented in sentences and in tokens. It is entirely based on Lexical Resources (LRs) either developed by linguists or automatically learnt from raw texts. These resources are either in the form of morpho-syntactic dictionaries or in the form of lexicalized local grammars.

2.1 Manually constructed lexical resources

The lexical module includes a large-coverage morpho-syntactic dictionary of inflected French forms. This dictionary has been developed between the mid-80's and the mid-90's by linguists at the University of Paris 7 [8, 9]. It is composed of 746,198 inflected simple forms and 249,929 inflected compounds (including 245,436 compound nouns). This dictionary is a set of lexical entries, each of them being composed of an inflected form, a lemma, a part-of-speech, morphological information (*e.g.* gender, number), syntactic information (*e.g.* internal structure of MWUs) and semantic information (*e.g.* human feature for nouns). It is of a great interest because of its fine-grained linguistic precision and the large amount of MWUs. These MWUs are compound words of the following types:

– nouns: *pomme de terre* (potato), *faux témoignage* (perjury)

syntactic internal structure of the MWUs (*i.e. noun-preposition-noun* and *noun-preposition-noun-adjective*).

Level	Example
Text	Le groupe de télécommunications néerlandais KPN a annoncé avoir acquis une participation de 77,5% dans le troisième opérateur allemand de téléphonie mobile E-Plus.
Lexical	Le [$_N$ groupe de télécommunications] néerlandais KPN a annoncé avoir acquis une participation de 77,5% dans le troisième [$_N$ opérateur allemand de téléphonie mobile] E-Plus.
Chunk	Le [$_N$ groupe de télécommunications] [$_{XA}$ néerlandais] KPN a annoncé [$_{XVI}$ avoir acquis] une participation de 77,5% dans le [$_{XA}$ troisième] [$_N$ opérateur allemand de téléphonie mobile] E-Plus.
	[$_{XN}$ Le groupe de télécommunications] [$_{XA}$ néerlandais] [$_{XN}$ KPN] a annoncé [$_{XVI}$ avoir acquis] [$_{XN}$ une participation] de [$_{XN}$ 77,5%] dans [$_{XN}$ le troisième opérateur allemand de téléphonie mobile E-Plus].
	[$_{XN}$ Le groupe de télécommunications] [$_{XA}$ néerlandais] [$_{XN}$ KPN] [$_{XV}$ a annoncé avoir acquis] [$_{XN}$ une participation] [$_{XP}$ de 77,5%] [$_{XP}$ dans le troisième opérateur allemand de téléphonie mobile E-Plus].
Sentence	[$_{N0}$ Le groupe de télécommunications néerlandais KPN] [$_V$ a annoncé avoir acquis] [$_{N1}$ une participation de 77,5% dans le troisième opérateur allemand de téléphonie mobile E-Plus].

Table 1. Global process

– prepositions: *au milieu de* (in the middle of), *à cause de* (because of)
– adverbs: *par ailleurs* (moreover), *en pratique* (in practice)
– conjunctions: *bien que* (although), *pendant que* (while)

This large-coverage dictionary is compressed in the form of an FST in order to be efficiently applied to the text.

Our lexical resources also contain a library of lexicalized local grammars. Local grammars [10] are Recursive Transition Networks (RTNs) [11] and theoretically recognizes algebraic languages. They are of great interest for representing local lexical and syntactic constraints in a simple and compact way. We use them mostly to describe MWUs. They can define syntactic classes such as noun determiners and even syntactico-semantic classes such as time adverbials. Linguistic descriptions are in the form of Finite-State Graphs on an alphabet made of terminal and non-terminal symbols. A terminal symbol is a lexical mask, *i.e.* an underspecified lexical entry (some features are missing) equivalent to a feature structure representing a set of lexical entries: *e.g.* the lexical mask <*noun+p*> matches all nouns in the plural. Finally, a non-terminal symbol is a reference to another graph. A graph is a transducer and its output is the annotation assigned to structures described in the graph. An example of a local grammar is given in figure 1[5]. This grammar describes time adverbials and recognizes structures like

[5] The local grammars are drawn using the graph editor of the Unitex platform [12].

en mars 2007 (in March 2007) and *cinq minutes plus tard* (five minutes later). The sequences recognized by this graph are tagged as time adverbs (`ADV+time`[6]). Strings between < and > are lexical masks: for instance, <*minute*> stands for the inflected forms whose lemma is *minute*. Greyed vertices are call to other graphs. For example, `Dnum` and `month` are graphs that respectively recognizes numerical determiners and the names of months.

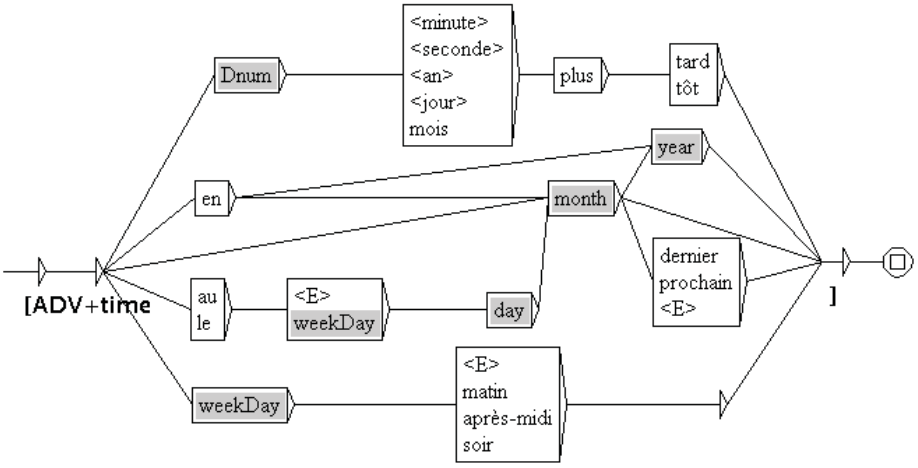

Fig. 1. Local grammar of time adverbials

Practically, the lexical module includes a network of 190 graphs. Local grammars recognize sequences of the following types:

- nouns: function names [*ministre anglais de l'Agriculture* (English minister of Agriculture)]
- prepositions: locative prepositions [*à dix kilomètres au nord de* (ten kilometers north of)]
- determiners: numerical determiners [*vingt-sept* (twenty seven), *des milliers de* (thousands of)], noun determiners [*dix grammes de* (ten grams of)]
- adverbs: time adverbials [*en octobre 2006* (in october 2006)]

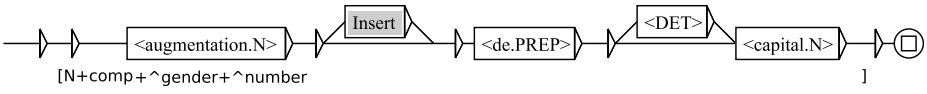

Fig. 2. Collocation: *rise of capital*

[6] The output is in bold and is displayed under the vertices of the graph.

2.2 Nominal collocations

The lexical analyzer also uses a set of nominal collocations. Collocations are sequences of words that co-occur more often than usually expected. Their automatic extraction from raw corpora has been the focus of many papers during the last decade. Nominal collocations can contain a preposition and therefore could be useful for preposition attachment in parsing. For instance, the collocation *offre d'emploi* (job offer) forms a basic semantic unit with the internal structure *noun-preposition-noun*. Considering it as a lexical unit would then resolve the prepositional attachment. To extract collocations, we used the approach defined in [13] and [14]. It consists in applying a set of nominal syntactic patterns on a tagged text and then in evaluating statistically identified candidates. Our learning text comprises 1 million words of French broadcast news and is tagged with TreeTagger [15]. Morpho-syntactic ambiguity, the principal source of noise in extraction, is therefore removed. Each word of the text is associated with a unique lemma and a unique part-of-speech. Next, local grammars representing basic nominal structures and their variants are applied to the tagged text. Extracted candidates that have a frequency greater than 5 are then statistically evaluated by computing the *log-likelihood* measure defined by [13] for bigrams and by [16] for trigrams. Finally, the best nominal collocations are kept and each one is assigned an internal syntactic structure. Given this structure, a local grammar is automatically constructed for each collocation. Each local grammar represents potential variations of the corresponding collocation (*e.g.* taking the insertion of a modifier into account). For instance, the local grammar associated with the collocation *augmentation de capital* (rise of capital, *cf.* Figure 2) recognizes the sequence *augmentations exceptionnelles de capital* (exceptional rises of capital). We therefore extracted 1,330 basic canonical bigrams and 163 basic canonical trigrams. Note that the number of extracted collocations could seem rather low. Nevertheless, as we want to obtain a very low error rate in order to have a totally automated process, we put very strong statistic constraints on the extraction computation. Note that, among the extracted collocations, 69.1% of bigrams and 86.5% of trigrams contain a prepositional attachment, which shows the great interest of locating the extracted nominal collocations during the lexical segmentation process. More details on this extraction process can be found in [17].

2.3 LR application

The lexical segmentation module is divided in two stages: (1) dictionary lookup then (2) application of lexicalized local grammars. The dictionary lookup stage enables to associate each token with all its possible linguistic tags and to recognize MWUs. The output of the process is a Text Finite State Automaton (TFSA). Then, local grammars are directly applied to the TFSA, which is then augmented with the analyses of the matching MWUs.

This process also allows for reducing artificial ambiguities by removing very infrequent analyses from the dictionary[7]. For instance, the analysis of *a* as a noun (letter *a*) is removed. In order to avoid the silence caused by the removal of analyses from the dictionary, it is possible to get these analyses given a specific context. To do so, we use local grammars. For instance, the form *par* is only tagged as a preposition, except if it belongs to local golf-specific lexicalized contexts such as *16 au-dessous du par* (16-under) in which context it is also tagged as a noun. Some MWUs also require a specific context to be analyzed as such. For instance, the sequence *en train de* can be interpreted as a preposition if it is followed by an infinitive verb:

Jean est [en train de PREP] dormir (John is sleeping)

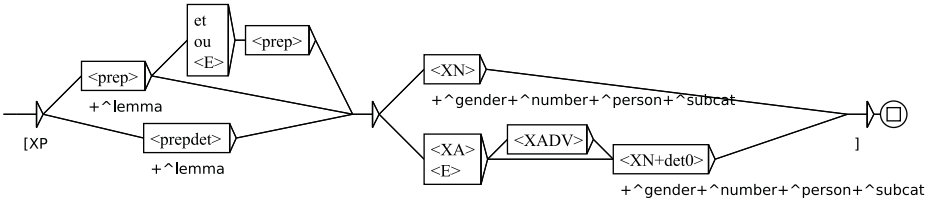

Fig. 3. Prepositional chunk

3 Chunk segmentation

Chunking is also an incremental process: it is in the form of a cascade of FSTs applied on the TFSA, which is then augmented each time a new chunk is found. It is composed of height stages and uses a network of 18 graphs. It consists in identifying:

1. adverbials (`XADV`): simple adverbs or multiword adverbials that have been recognized during the lexical segmentation
2. adjectival chunks (`XA`): adjectives that can be preceded by an adverb
3. nominal chunks (`XN`): simple noun phrases, named entities, some types of pronouns
4. prepositional chunks (`XP`): `XN` preceded by a preposition
5. verbal chunks (cascade of 4 FSTs): passive and active forms of infinitive, past participle, gerund and simple verbal chunks (`XVI`, `XVI-passive`, `XVK`, `XVK-passive`, `XVG`, `XVG-passive`, `XV`, `XV-passive`)

In general, the identified chunks inherit morpho-syntactic properties from their head as it is shown in figure 3 that represents an `XP`. `XP` inherits the gender and the number of its head (`^gender` and `^number`).

[7] Actually, we use a system of priorities.

Once the cascade of FSTs was applied on the TFSA, the latter is cleaned. The cleaning process consists in removing the transitions whose labels do not belong to the chunking level (*e.g.* nouns, verbs, adjectives, ...). It keeps only paths of the TFSA that go from the initial state (beginning of sentence) to the final state (end of sentence).

The chunking process applied to the sequence *au sujet d'un attentat terroriste* produces the TFSA given in the figure 4.

4 Incremental disambiguation

The chunk segmentation produces a set of possible analyses in chunks in the form of a TFSA for each sentence of the input text. In order to reduce or remove ambiguity, the chunker includes an incremental disambiguation module composed of three optional stages.

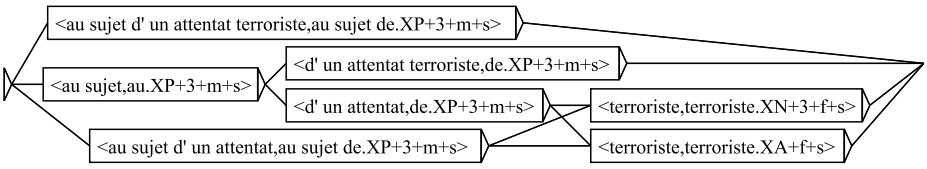

Fig. 4. TFSA after chunking

4.1 Applying the Shortest Path Heuristic (SPH)

The SPH consists in keeping only the shortest paths of the TFSA. This language-independent heuristic could seem simple and naïve at first sight. But, in practice, it is very efficient because it is based on the idea of preferring multiword expression analyses to sequences of simple analyses. The SPH algorithm is an adaptation of Dijkstra's algorithm to keep all shortest paths of a graph instead of one only.

The application of the heuristic to the TFSA in figure 4 produces a TFSA reduced to one path: *<au sujet d'un attentat terroriste.XP>*.

4.2 Applying hand-crafted rules

Given an instance of ambiguity and specific left and right contexts, the chunker user might want to prefer an analysis to the others. We therefore developed a simple formalism of disambiguation rules. A rule consists of three parts: two contextual parts (left and right) that are represented by local grammars (these two parts can be empty: `EMPTY`); a central ambiguous part that is a list of possible analyses. If the ambiguity is found in the TFSA with the defined left and right

contexts, then the first analysis in the list of ambiguous items is selected. The other analyses are then removed from the TFSA.

For instance,

```
XN_elag.wrtn
  <XP> <XN>
    EMPTY
```

When applied to the TFSA in figure 5, the rule above would keep only the `XP` analysis for the sequence *de lutte contre le terrorisme* (of war against terrorism) in the right context of an `XN` (recognized by the `XN_elag.wrtn` grammar).

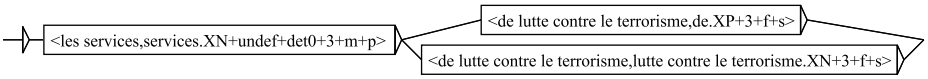

Fig. 5. Chunk ambiguity

4.3 Applying stochastic rules

There are some types of ambiguity that cannot be resolved by manually constructed general rules. A typical example is the `XV-XN` ambiguity (*e.g.* the word *massacre* can be an `XV` or `XN`). We therefore decided to use probabilistic rules automatically learnt from an automatically tagged corpus[8]. Given a word form that can be tagged either as a noun or a verb, the most frequent analysis is prefered. For instance, the form *massacre* would be tagged noun because the analysis noun has a probability of 0.7 while the analysis verb has a probability of 0.3 in our corpus. If no occurrences of a given form have been found in our corpus then the most frequent chunk category is selected (for instance, `XN` would be prefered to `XV`).

Note that all stages are optional because linearization is optional. There are some cases where it is better to keep ambiguities when resolving them is too risky: for instance, if chunking is applied just before an attachment resolution module. The `XN-XV` ambiguity is a typical example of ambiguity that is better resolved at the attachment level.

5 Evaluation and Discussion

Our evaluation process has been carried out on a corpus composed of broadcast news from `http://www.yahoo.fr` web site. This 13,492-word corpus includes 6,901 super-chunks.

[8] The corpus is one year of the newspaper `Le Monde` and has been tagged with TreeTagger.

Our definition of a chunk is different from the standard definition because it integrates the notion of MWU. As a consequence, there exists no reference annoted corpora that is compatible with our definition. The evaluation was therefore carried out manually. The 3-stage process described above was applied to the corpus: (1) lexical segmentation using dictionaries and local grammars presented in section 3; (2) segmentation in super-chunks by applying successively 18 finite-state rules; (3) desambiguation by applying SPH, then 26 hand-built rules and then a statistical module. Precision and recall were evaluated manually by two persons. Table 2 shows the results obtained.

Precision	92.9%
Recall	98.7%

Table 2. Results

From a general point of view, we observed that most of errors are due to incomplete lexical and syntactic resources. That means that improvements can easily be made and will soon be. In practice, we distinguish recall errors and precision ones.

Recall errors are only caused by incomplete LRs: dictionaries and local grammars. First, some compound grammatical words are missing from the dictionary, *e.g. tandis que* (while), *au-dessous de* (below). Moreover, elements are missing from the local grammars. For instance, in the named entity grammar, the sequence *Nouri al Maliki* is not recognized as a unit because the form *al* is unknown and has not been integrated in the grammar as family name prefix. There are some semi-fixed expressions that have not been integrated such as *vers 8h45* (at around 8.45 am). Some complex pronominal structures like *au cours de laquelle* are also missing in the local grammars.

Precision errors can be divided into four classes:

1. SPH-related errors

Lexical ambiguity can lead to wrong chunks after computing the SPH heuristic because it tends to keep the longest chunks. For instance, in the sentence *après l'affirmation du quotidien espagnol El Pais* (after the writing of the Spanish newspaper El Pais), there are two possible analyses:

[après l'affirmation XP] [du quotidien espagnol XP] [El Pais XN]

[après l'affirmation XP] [du quotidien XP] [espagnol XA] [El Pais XN]

As *quotidien* and *espagnol* are both adjectives or nouns, SPH prefers the `[Prep XA N]` analysis to the `[Prep N] [XA]` one.

2. Wrong decision based on probabilistic rules

For example, in the sentence *La côte Est et les villes de New York ...*, there are two possible analyses of the chunk *Est*: either an `XV` (to be) or an `XA` (East). Although *Est* is `XA` in this context, the statistical module prefers the analysis `XV` because *est* is more often tagged as a verb in our training corpus.

3. Errors caused by the application of disambiguation rules

That kind of errors is fortunately infrequent. They mainly concern the `XP`–`XN` ambiguity due to the lexical ambiguity of *de* which can be either a determiner or a preposition. For instance, a disambiguation rule that is applied at a late stage of the incremental disambiguation process, prefers `XP` to `XN`. This analysis is used to make an arbritrary decision. In the example *qui n'a pas fourni de plus amples détails* (who didn't provide more details), the chunk *de plus amples détails* is an *XN*.

4. Dictionary coverage

Some missing compound structures in the dictionary cause errors. For instance, *en outre* is a compound adverb but is missing from the dictionary. Therefore, the compositional analysis is chosen in the sentence *ils ont en outre pris plusieurs centaines de personnes en otage* (they took several hundreds of hostages). It is then chunked as follows

[ils XN] [ont XV] [en outre pris plusieurs centaines XP] [de personnes XP] [en otage XP]
instead of
[ils XN] [ont en outre pris XV] [plusieurs centaines XN] [de personnes XP] [en otage XP]

where *en outre* is an adverb inserted in a verbal chunk.

In addition to recall and precision evaluation, we also estimated the impact of MWUs for lexical attachment. We observed the actual realization of 36.6% of the lexical attachment, with no error, within noun phrases and prepositional ones.

We also applied our chunker on the same corpus without integrating MWU resources. Our chunker then becomes a standard chunker. The corpus passes from 6,901 super-chunks to 7,503 chunks (around 8% augmentation). We then observe a slight rise of precision by using MWUs: the number of errors falls from 600 to 485. Recall is also slightly better: 116 analyses are missing without using MWUs vs. 89 with MWUs. All these figures show the great interest of using super-chunks instead of only standard chunks. First, as the number of chunks decreases, the combinatorial ambiguity is reduced and further processes should be eased. Moreover, as MWUs form "islands of non-ambiguity", they are useful to reduce internal ambiguity within chunks and therefore to reduce precision errors.

6 Conclusion and perspectives

In this paper, we presented a chunking technique based on a significant augmentation of the lexical level by taking into account larger sequences (*i.e.* MWUs). By using this approach, we search for optimizing the disambiguation process on

the one hand and computing a part of the lexical attachment within noun and prepositional phrases on the other hand.

To evaluate the relevancy and the efficiency of our assumption, we carried out an experiment on a broadcast news corpus from the web. Results led us to a double conclusion:

- Our disambiguation procedure reaches excellent recall and precision rates without the use of any tagger;
- a significant amount of attachments within noun and prepositional phrases are actually resolved by the use of a large-coverage set of MWUs, and therefore do not have to be computed at the syntactic level.

Future work will focus on the improvement of the super-chunker by improving the lexical and syntactic resources and by integrating a more sophisticated statistical disambiguation module (*e.g.* use of Hidden Markov Models). We wish to extend it in order to process less stable textual data such as spoken texts or emails. Moreover, we would like to compare the super-chunker with standard chunkers and to evaluate its impact when it is integrated in a parser.

References

1. Joshi, A., Hopely, P.: A parser from antiquity. Natural Language Engineering **2**(4) (1997)
2. Nivre, J., Nilsson, J.: Multiword units in syntactic parsing. In Dias, G., Lopes, J.G.P., Vintar, S., eds.: Proceedings of the Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications, Language and Resource Evaluation Conference. (2004) 39–46
3. Blanc, O., Constant, M.: Outilex, a linguistic platform for text processing. In: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. (2006) 73–76
4. Abney, S.P.: Partial parsing via finite-state cascades. Natural Language Engineering **2**(4) (1996) 337–344
5. Karlsson, F., Voutilainen, A., Heikkila, J., Anttila, A.: Constraint Grammar: A language-independent system for parsing unrestricted text. Volume 4 of Natural Language Processing. Mouton de Gruyter (1995)
6. Federici, S., Montemagni, S., Pirelli, V.: Shallow parsing and text chunking: A view on underspecification in syntax. In: Proceedings of the ESSLLI'96 Workshop on Robust Parsing. (1996)
7. Ait-Mokhtar, S., Chanod, J.P.: Incremental finite-state parsing. In: Proceedings of the fifth Conference on Applied Natural Language Processing ANLP'97. (1997)
8. Courtois, B.: Un système de dictionnaires électroniques pour les mots simples du français. Langue Française **87** (1990) 11–22
9. Courtois, B., Garrigues, M., Gross, G., Gross, M., Jung, R., Mathieu-Colas, M., Monceaux, A., Poncet-Montange, A., Silberztein, M., Vivés, R.: Dictionnaire électronique DELAC : les mots composés binaires. Technical Report 56, LADL, University Paris 7 (1997)
10. Gross, M.: The construction of local grammars. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. The MIT Press, Cambridge, Mass. (1997) 329–352

11. Woods, W.: Transition network grammars for natural language analysis. Communications of the ACM **13**(10) (1970)
12. Paumier, S.: De la reconnaissance de formes linguistiques à l'analyse syntaxique. PhD thesis, Université de Marne-la-Vallée (2003)
13. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics **19**(1) (1993) 61–74
14. Daille, B.: Combined approach for terminology extraction: lexical statistics and linguistic filtering. Technical report, Lancaster University (1995)
15. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: International Conference on New Methods in Language Processing, Manchester, UK (1994)
16. Seretan, V., Nerima, L., Wehrli, E.: Extraction of multi-word collocations using syntactic bigram composition. In: Proceedings of th 4^{th} International Conference on Recent Advances in NLP (RANLP-2003). (2003) 424–431
17. Watrin, P.: Une approche hybride de l'extraction d'information : sous-langages et lexique-grammaire. PhD thesis, Université catholique de Louvain (2006)