Computer aided correction and extension of a syntactic wide-coverage lexicon

Lionel NICOLAS$^\diamondsuit$, Benoît SAGOT$^\clubsuit$, Miguel A. MOLINERO$^\spadesuit$,
Jacques FARRÉ$^\diamondsuit$, Éric DE LA CLERGERIE$^\clubsuit$

$^\diamondsuit$Team RL, Laboratory I3S - UNSA + CNRS, 06903 Sophia Antipolis, France
{lnicolas, jf}@i3s.unice.fr
$^\clubsuit$Project ALPAGE, INRIA Rocquencourt + Paris 7, 78153 Le Chesnay, France
{benoit.sagot, Eric.De_La_Clergerie}@inria.fr
$^\spadesuit$Grupo LYS, Univ. de A Coruña, 15001 A Coruña, España
mmolinero@udc.es

Abstract

The effectiveness of parsers based on manually created resources, namely a grammar and a lexicon, rely mostly on the quality of these resources. Thus, increasing the parser coverage and precision usually implies improving these two resources. Their manual improvement is a time consuming and complex task : identifying which resource is the true culprit for a given mistake is not always obvious, as well as finding the mistake and correcting it.

Some techniques, like van Noord (2004) or Sagot and Villemonte de La Clergerie (2006), bring a convenient way to automatically identify forms having potentially erroneous entries in a lexicon. We have integrated and extended such techniques in a wider process which, thanks to the grammar ability to tell how these forms could be used as part of correct parses, is able to propose lexical corrections for the identified entries.

We present in this paper an implementation of this process and discuss the main results we have obtained on a syntactic wide-coverage French lexicon.

1 Introduction

Increasing the coverage and precision of non-trained parsers based on manually created grammar and lexicon, relies mostly on the improvement of these two resources.

The manual improvement of wide coverage linguistic resources is a labour-extensive, complex and error prone task, requiring an important human expert work.

In order to minimize human intervention, simplify the process and increase its relevance, automatic or semi-automatic tools can be used. We present one such tool, using raw inputs, which detects shortcomings of a lexicon and helps correct them by proposing relevant corrections.

Detecting forms erroneously or incompletely described in the lexicon is achieved by applying two techniques which exhibit *suspicious forms* and associate them with a set of non parsable sentences.

Proposing relevant corrections relies on the following assumption: when studying the expectations of a grammar for a suspicious form in various non-parsable sentences, we can observe expected use patterns for this form. Those patterns can be regarded as possible corrections for the lexicon. In a metaphorical way, we believe the problem to be due to the lexicon[1], and we ask the grammar to express possible corrections for the lexicon.

The set of techniques we present here is fully system and language independent: it can be easily applied to most existing lexica and unification-grammar based parsers. The only condition is to provide lexically and grammatically valid inputs, such as law texts or newspapers, in order to ensure that the rejection of a sentence is only due to errors

[1] We will discuss later lexical forms incorrectly suspected because of errors in other components of the parsing system, notably the grammar.

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 633–640
Manchester, August 2008

in some components on which the parser relies on.

This paper is organized as follows. We start by giving a global view of the whole process (Sect. 2) that we later detail step by step (Sect. 3, 4, 5, 6 and 7). We then compare our work with previously publicated ones (Sect. 8) right before exposing the practical context and the results obtained (Sect. 9). Finally, we outline the planned improvements (Sect. 10) and conclude (Sect. 11).

2 Global view

A lexical form is generally described in a lexicon with one or more entries including different kinds of information: the POS (part of speech), morphologic features, syntactic features and sometimes semantic features.

A form will cause a parsing failure if its description in the lexicon leads to a conflict with the grammar expectations for this form, i.e., if grammar and lexicon do not agree on a particular instance of the form in a given sentence.

For practical reasons, we make a difference between conflicts concerning the POS, that we now call **POS defect**, and conflicts concerning the features, that we now call **overspecification**. POS defect generally happen with homonyms, i.e., with forms related to frequent lemmas while a seldom used one is missing in the lexicon. Overspecification is usually caused by the difficulty of describing exhaustively all the subcategorization frames of a lemma (optional arguments, polysemy, etc.). Therefore, if for a given lemma the most restrictive frames are also the most frequent, some entries can be overspecified and induce such conflicts.

We generate lexical corrections according to the following process.

1. We first detect suspicious forms and associate them with a set of non-parsable sentences in which the form is suspected to be responsible of the sentences' parsing failures.

2. We get as close as possible to the set of parses that the grammar would have allowed with an error-free lexicon. We achieve this goal by underspecifying the suspicious form, i.e., we increase the set of its possible POS (that is, by virtually adding new entries to the lexicon) and/or underspecify the morphological and syntactic informations of a given existing entry. A full underspecification can be simulated in the following way: during the parsing process, each time a lexical information is checked about the suspicious form, the lexicon is bypassed and all the constraints are considered as satisfied. We actually achieved this operation by replacing the suspicious form in the associated sentences with special forms called **wildcards**.

3. If the suspicious form has been correctly detected, such underspecification increases the parsing rate (except for sentences for which the form was not the only problem). In the newly successful parses, the form became whatever the grammar wanted it to be, i.e., it has matched any morphological, syntactic or semantic pattern required. Those patterns are the data we use to generate the corrections. We thus extract the instances of the wildcard in the newly producted parses, and after ranking, we propose them as corrections.

4. Finally, we manually validate and apply the corrections.

We will now explain with details how each step is achieved, starting with the detection of suspicious forms.

3 Detection of suspicious forms

In order to detect erroneous entries in a lexicon, we have developed and implemented two techniques : a shallow technique to identify POS defects and an extended version of an existing parser-based technique to (mostly) identify overspecification. Both provide for the form a suspicion rate and a set of associated non-parsable sentences.

3.1 Tagger-based detection of POS defects

This technique is based on a stochastic tagger. The underlying idea is to generate new POS for forms in the input corpus by using an especially configured stochastic tagger (Molinero et al., 2007). Such a tagger considers every form belonging to open POS (adjectives, common nouns, adverbs, verbs and proper nouns) as unknown. Candidate POS for unknown forms are then proposed by the tagger's guesser and the most likely to be correct are selected by the tagging process itself. Thus, new POS arise for some forms present in the input.

To obtain such a tagger, we have used two training sets. One is a training corpus composed of manually tagged sentences (330k words) extracted from the French Paris 7 Treebank (Abeillé, 2003), and the other one is composed of a small list of lemmas belonging to closed POS (prepositions, determiners, pronouns and punctuation marks). The tagger was modified so that only lemmas present in the second set are considered as known.

After applying the tagger on the input corpus, we extracted the produced pairs of form/POS and checked their presence in the lexicon. Every non present pair has been proposed as POS defect candidate. The emergence of false positives has been smoothed by sorting the POS defect candidates according to the following measure:

$$(n_{wt}/n_w) * log(n_{wt}),$$

where n_{wt} is the number of occurrences of the form w tagged as t and n_w is the total number of occurrences of the form w.

3.2 Parsing-based suspicious forms detection

The technique described hereafter extends the ideas exposed in (Sagot and Villemonte de La Clergerie, 2006), in which suspicious forms are detected through a statistical analysis of the parsing success and errors produced by a parser.

This error mining technique relies on the following idea.

- When the parsing of a sentence known to be lexically and grammatically correct fails, there is no automatic and unquestionable way to decide if this rejection is caused by an error in the lexicon or by a flaw in another component of the parsing system (grammar, etc.).

- Given the parsing results of a large corpus of reliable sentences, the more often a lexical form appears in non-parsable sentences and not in parsable ones, the more likely it is that its lexical entries are erroneous. This suspicion is reinforced if it appears in non-parsable sentences together with forms that appear mostly in parsable ones.

The statistical computation establishes a relevant list of lexical forms that are likely to be incorrectly or incompletely described in the lexicon.

As such, the main drawback of this approach is the dependence to the quality of the grammar used. Indeed, if a specific form is naturally tied with some syntactic construction non-handled by the grammar, this form will always be found in rejected sentences and will thus be unfairly suspected. Nevertheless, we limited this drawback by applying two extensions.

The first, already described in (Sagot and Villemonte de La Clergerie, 2006), mixes the detection results obtained from various parsers with different grammars, hence with different shortcomings.

The second extension detects short-range representative syntactic patterns non-handled by the grammar and filters the non-parsable sentences where they appear. To do so, we reduce every sentence to a single POS sequence through the use of a tagger and train a maximum entropy classifier (Daumé III, 2004) with the different possible trigrams and the corresponding parse failure or success. Even if non-perfect (the tagger or the maximum entropy classifier might be mistaken at some point), this pre-filtering has proved to noticeably increase the quality of the suspicious forms provided.

We will now explain how we manage to permit the parsing process of the associated non-parsable sentences in order to extract afterwise the corrections hypotheses.

4 Parsing originally non-parsable sentences with wildcards

As explained in Sect. 2, in order to generate lexical corrections, we first need to get as close as possible to the set of parses that the grammar would have allowed with an error-free lexicon. We achieve this goal by replacing in the associated sentences every suspicious forms with special underspecified forms called **wildcards**.

The simplest way would be to use totally underspecified wildcards. Indeed, this would have the benefit to cover all kinds of conflicts and thus, it would notably increase the parsing coverage. However, as observed by (Fouvry, 2003), it introduces an unnecessary ambiguity which usually leads to a severe overgeneration of parses or to no parses at all because of time or memory shortage. In a metaphorical way, we said that we wanted the grammar to tell us what lexical information it would have accepted for the suspicious form. Well, by introducing a totally underspecified wildcard, either the grammar has so many things to say that it is hard to know what to listen to, or it has so many things to think about that it stutters and does not say anything at all.

Therefore, we refined the wildcard by introducing some data. For technical, linguistic and readability reasons, we added POS information.

When facing a **POS defect**, we need the parser to explore other grammar rules than those already visited during the failed parses. We thus generate wildcards with different POS than those already present in the lexicon for the suspicious form.

When facing an **overspecification**, we need the parser to explore the same grammar rules without being stopped by unification failures. We thus generate wildcards with the same POS than those already present in the lexicon, but with no feature-level constraints.

When suspicious forms were correctly detected, such exchanges usually increases the parsing rate of the associated sentences. Those parses place the wildcards in grammatical contexts/patterns which clearly express what lexical informations the grammar would have accepted for the suspicious forms.

We will now explain how we extract the correction hypotheses from the newly obtained parses and how we rank them.

5 Extracting corrections hypotheses

The extraction directly depends on how one planes to use the correction hypotheses. In a previous work (Nicolas et al., 2007), we extracted the corrections proposals in the parser's output format. Such a way to process had three important drawbacks :

- one needed to understand the parser's output format before being able to study the corrections;
- merging results produced by various parsers was difficult, although it is an efficient solution to tackle most limitations of the process (see Sect. 6.2);
- some parts of the correction proposals were using representations that are not easy to relate with the format used by the lexicon (specific tagsets, under- or overspecified information w.r.t. the lexicon, etc.).

We therefore developed for each of the parsers used a conversion module in order to extract from a given parse the instantiated lexical entry of each wildcard in the format used by the lexicon.

6 Ranking corrections hypotheses

Natural languages are ambiguous, and so need to be the grammars that model them. For example, in many romance languages, an adjective can be used as a noun and a noun as an adjective.

Consequently, an inadequate wildcard may perfectly lead to new parses and provide irrelevant corrections. We thus separate the correction hypotheses according to their corresponding wildcard before ranking them. Afterwards, the parsing rate induced by each type of wildcard and the associated parsed sentences allows to easily identify which wildcard is the correct one.

When only one parser is used to generate correction hypotheses, ranking correction hypotheses proves straightforward, but, as we will explain, the results heavily depend on the quality of the grammar. We thus put together correction hypotheses obtained thanks to different parsers in order to rank them in a more sophisticated way.

6.1 Baseline ranking: single parser mode

The correction hypotheses obtained after introducing a wildcard are generally irrelevant, i.e., most of them are parasitic hypotheses resulting from the ambiguity brought by the wildcards. Nevertheless, among all these hypotheses, some are valid, or at least close to valid. In the scope of only one sentence, there is no reliable way to determine which corrections are the valid ones. But, if we consider simultaneously various sentences that contain the same suspicious form embedded in different syntactic structures, we usually observe a strong variability of the noisy correction hypotheses. On the opposite, if some stable correction hypothesis is proposed for various sentences, it is likely to be valid, i.e, to represent the correct sense of the form according to the grammar. We thus simply rank correction hypotheses according to the number of sentences that have produced them.

6.2 Reducing grammar influence: multi-parser mode

Using various parsers not only improves the suspicious forms detection (Sect. 3.2), it also allows to merge correction hypotheses in order to minimize the influence of the shortcomings of the grammars. When some form is naturally related to syntactic constructions that are not correctly handled by the grammar, this form is always found in rejected sentences, and therefore is always suspected. Replacing it by wildcards will only produce incorrect corrections or no correction at all because the problem is not related to the lexicon.

Having various sets of non-parsable sentences for a given suspicious form f, and various sets of correction hypotheses for f, one can discard (or consider less relevant) correction hypotheses according to the following three statements:

- If any form in a sentence is actually incorrectly described in the lexicon, then this sentence should be non-parsable for both parsers. Correction hypotheses produced from sentences that are non-parsable for only one parser should be discarded.
- For the same reason, correction hypotheses produced with sentences in which only one parser made f identified as a suspicious form should be avoided.
- Finally, correction hypotheses proposed by only one of both parsers (or proposed much more often by one parser than by the other one) might just be the consequence of the ambiguity of one grammar. Afterall, both grammar describe the same language, they should find an agreement about the uses of a form.

In our experiments, we decided to apply the following ranking scheme: for a given suspicious form, we only keep the corrections hypotheses that are obtained from sentences that were originally non-parsable and parsable after a wildcard introduction for both parsers. Afterwards, we separately rank the correction hypotheses for each parser and merge the results.

We will now explain how we manually validate the ranked correction hypotheses.

7 Manual validation of the corrections

When studying the ranked corrections for a given wildcard, there might be three cases:

1. There are no corrections at all: the form was unfairly suspected or the generation of wildcards was inadequate. It also happens when the erroneous entries of the suspicious form are not the only reasons for all the parsing failures.
2. There are relevant corrections: the form was correctly detected, the generation of wildcards was adequate and the form was the only reason for various parsing failures.
3. There are irrelevant corrections: the ambiguity introduced by the relevant or irrelevant wildcards opened the path to irrelevant parses providing irrelevant corrections.

It is truly important to note that an incorrectly suspected form may perfectly lead to irrelevant corrections brought by the ambiguity introduced. Consequently, unless the grammar used, the detection of suspicious form and the generation of wildcards are perfect, such a correcting process should always be semi-automatic (manually validated) and not automatic.

Now that the whole system has been explained with details, we will expose the similarities and differences of our methods with previously publicated ones.

8 Related works

Since efficient and linguistically relevant lexical and grammatical formalisms have been developed, the acquisition/extension/correction of linguistic ressources has been an active research field, especially during the last decade.

The idea to infer lexical data from the grammatical context first appeared in 1990 (Erbach, 1990). The combination with error mining/detection technique, such as van Noord (2004), begun in 2006 (van de Cruys, 2006; Yi and Kordoni, 2006). Except in our previous work (2007), nobody has combined it with the technique described in Sagot and Villemonte de La Clergerie (2006). The idea of prefiltering the sentences (Sec. 3.2) to improve the error mining performance has never been applied so far.

The wildcards generation started to be refined with Barg and Walther (1998). Since then, the wildcards are partially underspecified and restrained to open class POS. In Yi and Kordoni (2006), the authors use an elegant technique based on an entropy classifier to select the most adequate wildcards.

The way to rank the corrections is usually based on a trained tool (van de Cruys, 2006; Yi and Kordoni, 2006), such as an entropy classifier. Surprisingly, the evaluation of hypotheses on various sentences for a same suspicious form in order to discriminate the irrelevant ones has never been considered so far.

Finally, all the previous works were achieved with HPSG parsers and no results has been exposed until 2005. van de Cruys (2006) expose its results for each POS and one can clearly observe, for complex lemmas like verbs, the impossibility to apply such set of techniques in an automatic way without harming the quality of the lexicon. The results would be even worse if applied to corpus with sentences non-covered by the grammar because no relevant corrections could be generated but irrelevant ones might perfectly be.

9 Results

We now expose the results of our experiments by describing the practical context, giving some correction examples and discussing the effectiveness of the correction process through the parsing rate increases we have obtained.

9.1 Practical context

The lexicon we are improving is called the Le*fff*.[2] This wide-coverage morphological and syntactic French lexicon has been built partly automatically (Sagot et al., 2006) and is under constant development. At the time these lines are written, it contains more than 520 000 entries. The less data an entry has, the more specified it is.

We used two parsers based on two different grammars in order to improve the quality of our corrections.

- The FRMG (***Fr**ench **M**eta-**G**rammar*) grammar is generated in an hybrid TAG/TIG form from a more abstract meta-grammar with highly factorized trees (Thomasset and Villemonte de La Clergerie, 2005).
- The SxLFG-Fr grammar (Boullier and Sagot, 2006), is an efficient deep non-probabilistic LFG grammar.

The corpus used is extracted from the French politics newspaper *Le monde diplomatique*. This corpora is composed with around 280 000 sentences of 25 or less elements and 4,3 million of words in total.

9.2 Examples of corrections

9.2.1 POS corrections

Most of the POS corrections performed were about missing forms or about adjectives that could be used as noun and vice versa. Here are some examples :

- israélien (*israeli*) as an adjective,
- portugais (*portuguese*) as an adjective,
- politiques (*politic*) as a common noun,
- parabolique (*parabolic*) as an adjective,
- pittoresque (*picturesque*) as an adjective,
- minutieux (*meticulous*) as an adjective.

9.2.2 Features corrections

As one can expect, most of the features corrections performed were about lemmas with complex subcategorization frames / features, i.e., essentially verbs.

- "revenir" (*to come back*) did not handle constructions like *to come back from* or *to come back in*
- "se partager" (*to share*) did not handle constructions like *to share something between.*
- "aimer" (*to love*) was described as always expecting a direct object and an attribute.
- "livrer" (*to deliver*) did not handle constructions like *to deliver to somebody.*

9.3 Correction process relevance

As explained earlier (Sect. 7), this process might generate erroneous corrections, especially if general corpora with sentences non-covered by the grammar are used and various correction sessions are made. Globally, the accuracy of the corrections goes decreasing after each session. Indeed, there are less and less lexical mistakes to correct after each session. Anyway, we are more interested in improving efficiently our lexicon. We thus prove the relevance of the whole process by showing the gains of parsing rate obtained during our experiments. One must keep in mind that the corrections are manually validated, i.e, the noticeable increases of parsing coverage (Figure 1) are mostly due to the improvement of the quality of the lexicon.

Table 1 lists the number of lexical forms updated at each session.

Except for the second session, all correction sessions have been achieved with the error mining and the hypothesis generation modules. The second session has been achieved with the POS defect mining module only (Sect. 3.1). We planned to

[2] Lexique des formes fléchies du français/Lexicon of inflected forms of French.

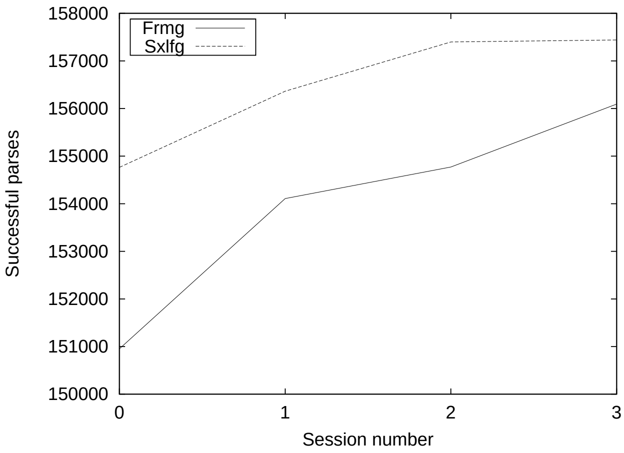

Figure 1: Number of sentences successfully parsed after each session.

Session	1	2	3	total
nc	30	99	1	130
adj	66	694	27	787
verbs	1183	0	385	1568
adv	1	7	0	8
total	1280	800	413	2493

Table 1: Lexical forms updated at each session

interface it with the hypothesis generation module but we could not finish it on time. Nevertheless, the suspicious form list provided was good and simple enough (mostly proper nouns, adjectives and common nouns) to be reviewed without the help of the hypothesis generation module.

As expected, we were quickly limited by the quality of the grammars and by the corpus used. Indeed, the lexicon and the grammars have been developed together for the last few years, using this same corpus for test. Thus, the error mining technique came, after few corrections sessions, to provide us irrelevant suspicious forms. The tagger-based detection of POS defects can only be used once on each corpus. Further correction and extension sessions make sense only after grammar improvements or obtention of new corpora.

Nevertheless, we have already detected and corrected 254 lemmas corresponding to 2493 forms. The coverage rate (percentage of sentences for which a full parse is found) has undergone an absolute increase of 3,41% (5141 sentences) for the FRMG parser and 1,73% (2677 sentences) for the SXLFG parser. Thoses results were achieved in only few hours of manual work on the lexicon !

9.4 Discussion

This set of techniques has two major qualities.

The first one, as one can observe through our results, it allows to improve significantly a lexicon in a short amount of time.

The second one is related to the main drawback of our approach: the dependence to the grammars used. If in a non-parsable sentence, none of the suspicious forms is a true culprit (there are no relevant correction), then this sentence can be considered as lexically correct w.r.t. the current state of the grammar. It thus exhibits shortcomings of the grammar and can be used to improve it.

A cycling process which alternatively and incrementally improves both the lexicon and the grammar can then be elaborated. This data is even more important considering the fact that nowadays, large scale French TreeBank are rare.

10 Future developments

The whole system has globally proved to be mature. Nevertheless, we are planning the following improvements to continue our investigation.

- We need to interface the POS defect mining module with the hypothesis generation one.
- The tagger-based detection of POS defects is still young and can be improved.
- We will refine the wildcard generation in a similar way as done in (Yi and Kordoni, 2006).
- In order to pursue the corrections of the lexicon, we will improve our grammars according to the corpus of failed sentences. It is now globally representative of shortcomings of the grammars, thus we are thinking about developing some detection techniques in order to emphasize cases of error for the grammar. The entropy model built by the maximum entropy classifier should be a good starting point.

11 Conclusion

The path taken, highlighted by the dependence on the grammar, seems to be a promising one. It will allow to develop a cycling process which alternatively and incrementally improves both the lexicon and the grammar.

The correction process of lexicon presented here is now globally mature and has proved to be relevant and effective in practice. Indeed, noticeable improvements of the lexicon could be achieved in a few amount of manual work.

The time spend to validate the corrections generated has also confirmed our doubts about evolving such process to an automatic one.

We will definitively continue the correction sessions after upgrading some components.

Acknowledgement We thank the COLE team (University of Vigo, Spain) for granting us access to their computers.

We would like to thanks Sylvain Schmitz and the reviewers for their valuable comments.

The POS mining technique could be achieved partially thanks to the support of Ministerio de Educación y Ciencia of Spain (HUM2007-66607-C04-02) and the Xunta de Galicia ("Galician Network for Language Processing and Information Retrieval" 2006-2009).

References

Abeillé, Anne. 2003. Annotation morpho-syntaxique. Paper available at http://www.llf.cnrs.fr/Gens/Abeille/guide-morpho-synt.02.pdf, January.

Barg, Petra and Markus Walther. 1998. Processing unkonwn words in hpsg. In *Proceedings of the 36th Conference of the ACL and the 17th International Conference on Computational Linguistics*.

Boullier, Pierre and Benoît Sagot. 2006. Efficient parsing of large corpora with a deep LFG parser. In *Proceedings of LREC'06*.

Daumé III, Hal. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at http://pub.hal3.name/daume04cg-bfgs, implementation available at http://hal3.name/megam/, August.

Erbach, Gregor. 1990. Syntactic processing of unknown words. In *IWBS Report 131*.

Fouvry, Frederik. 2003. Lexicon acquisition with a large coverage unification-based grammar. In *Companion to the 10th of EACL*.

Molinero, Miguel A., Fco. Mario Barcala, Juan Otero, and Jorge Graña. 2007. Practical application of one-pass viterbi algorithm in tokenization and pos tagging. *Recent Advances in Natural Language Processing (RANLP). Proceedings, pp. 35-40*.

Nicolas, Lionel, Jacques Farré, and Éric Villemonte de La Clergerie. 2007. Correction mining in parsing results. In *Proceedings of LTC'07*.

Sagot, Benoît and Éric Villemonte de La Clergerie. 2006. Error mining in parsing results. In *Proceedings of ACL/COLING'06*, pages 329–336. Association for Computational Linguistics.

Sagot, Benoît, Lionel Clément, Éric Villemonte de La Clergerie, and Pierre Boullier. 2006. The Lefff 2 syntactic lexicon for french: architecture, acquisition, use. In *Proceedings of LREC'06*.

Thomasset, François and Éric Villemonte de La Clergerie. 2005. Comment obtenir plus des méta-grammaires. In *Proceedings of TALN'05*.

van de Cruys, Tim. 2006. Automatically extending the lexicon for parsing. In *Proceedings of the eleventh ESSLLI student session*.

van Noord, Gertjan. 2004. Error mining for wide-coverage grammar engineering. In *Proceedings of ACL 2004*.

Yi, Zhang and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open texts processing. In *Proceedings of LREC-2006*.