Mining Parsing Results for Lexical Correction: Toward a Complete Correction Process of Wide-Coverage Lexicons

Lionel Nicolas[1], Benoît Sagot[2], Miguel A. Molinero[3], Jacques Farré[1], and Éric de La Clergerie[2]

[1] Équipe RL, Laboratoire I3S, UNSA + CNRS, 06903 Sophia Antipolis, France, {lnicolas,jf}@i3s.unice.fr
[2] Projet ALPAGE, INRIA Rocquencourt + Paris 7, 78153 Le Chesnay, France, {benoit.sagot, Eric.De_La_Clergerie}@inria.fr
[3] Grupo LYS, Univ. de A Coruña, 15001 A Coruña, España mmolinero@udc.es

Abstract. The coverage of a parser depends mostly on the quality of the underlying grammar and lexicon. The development of a lexicon both complete and accurate is an intricate and demanding task. We introduce a automatic process for detecting missing, incomplete and erroneous entries in a morphological and syntactic lexicon, and for suggesting corrections hypotheses for these entries. The detection of dubious lexical entries is tackled by two different techniques; the first one is based on a specific statistical model, the other one benefits from information provided by a part-of-speech tagger. The generation of correction hypotheses for dubious lexical entries is achieved by studying which modifications could improve the successful parse rate of sentences in which they occur. This process brings together various techniques based on taggers, parsers and statistical models. We report on its application for improving a large-coverage morphological and syntacic French lexicon, the Le*fff*.

Key words: Lexical acquisition and correction, wide coverage lexicon, error mining, tagger, entropy classifier, syntactic parser

1 Introduction

The manual development of a lexicon that is both accurate and wide coverage is a labour-intensive, complex and error prone task, requiring an important human expert work. Unless very important financial and human efforts are put in the balance, the lexicons usually do not achieve the expected objectives in terms of coverage or quality. However, this manual task can be improved through the use of tools which simplify the process and increase its relevance.

We present a set of techniques brought together in a chain of tools which detect missing, incomplete and erroneous entries in a lexicon and proposes relevant lexical corrections.

The methodology implemented in this chain can be summarized as follows:

1. Parse a high number of raw (non tagged) sentences considered as lexically and grammatically valid (law texts, newspapers, etc.) with a deep parser,[4] and spot those that are successfuly parsed and those which ones are not;[5]
2. For each non-parsable sentence, determine automatically, thanks to a statistical classifier, if the parsing failure is caused by a lack of coverage of the grammar or by incompleteness of the morphological and syntactic lexicon;
3. Detect automatically missing, incomplete or erroneous lexical entries. This is achieved by a statistical analysis of non-parsable sentences for which the lexicon has been identified during the previous step as the cause of the parsing failure;
4. Generate correction hypotheses by analyzing the expectations of the grammar about those detected entries when trying to parse the non-parsable sentences in which they occur.
5. Automatically evaluate and rank corrections hypotheses to prepare an easier manual validation.

Although our examples and results are related to French, this set of techniques is system independent, i.e., it can be easily adapted to most existing taggers, classifiers, lexicons and deep parsers, and thus to most electronically described languages.

This chain of tools is one of the starting points of the recently created Victoria project [6], which aims at developing efficiently large-coverage linguistic resources for Spanish and Galician languages, with inter-language links with French resources (incl. the Le*fff* syntactic lexicon, see section 8).

Please note that *some* results presented in section 8 were partly obtained with a previous version of the chain and its architectural model [1]. The differences between both models are presented in details in section 8.

This paper is organized as follows. We first detail step by step the process described above (Sect. 2, 3, 4, 5 and 6). Next, we compare our approach with previous related work (Sect. 7). We expose the practical context and the results we obtained in Sect. 8. Finally, we outline the planned improvements (Sect. 9) and conclude (Sect. 10).

2 Classifying non-parsable sentences

Let us suppose we have parsed a large corpus with a deep parser. Some sentences were successfully parsed, some were not. Sentences that were parsed are both lexically and grammatically covered (even if the parses obtained do not match the

[4] In this paper, we consider only parsers that are able to exploit subcategorization information.

[5] These sentences need to be lexically and grammatically valid in order to ensure that a parsing failure is only due to shortcomings in the parser or of the resources it relies on.

[6] http://www.victoria-project.org (October 2008).

actual meaning of the sentences). On the contrary, and in first approximation, the parsing failure of a given sentence can be caused either by a lack of grammatical coverage or by a lack of lexical coverage.

However, our focus is the improvement of the lexicon. Therefore, we need to apply first a method for determining whether the parser failed on a given sentence because of a problem in the grammar or in the lexicon.

Since syntactic structures are more frequent and less numerous than words, grammatical shortcomings tend to correspond to recurrent patterns in non-parsable sentences, contrarily to lexical shortcomings. Moreover, syntactic problems in lexical entries have no impact on a tagger. This means that we can train a statistical classifier to identify sentences that are non-parsable because of a shortcoming of the grammar; such a classifier needs to be trained with contextual information, e.g., the set of n-grams that constitute the sentence. We built these n-grams using the POS (*part-of-speech*) for open-class forms (i.e., verbs, nouns, etc.) and the form itself for closed-class ones (i.e., prepositions, determiners, etc.). The classifier we used is a maximum entropy classifier [2].

The POS information we used is obtained by two different means. For parsable sentences (i.e., sentences covered by the grammar), POS tags and forms are directly extracted from parsing outputs. Indeed, we are only interested in syntactic patterns covered by the grammar, even if ambiguous parse outputs are used as training. For non-parsable sentences, we simply used a POS tagger. Although taggers are not perfect, their errors are random enough not to blur the global coherence of the classifier's model.

When applied on non-parsable sentences, this classifier identifies two sets of sentences:

- sentences that are non-parsable because of shortcomings in the grammar;
- all other non-parsable sentences, i.e., sentences that are non-parsable because of shortcomings in the lexicon.

3 Detecting lexical shortcomings

The next step of our lexicon improvement process is to detect automatically missing, incomplete or erroneous lexical entries. To achieve this goal, we use two complementary techniques that identify dubious lexical forms and associate them with non-parsable sentences in which they appear, and in which it is suspected they caused the parsing failure.

3.1 Tagger-based approach for detecting shortcomings in short-range lexical information

We call short-range lexical information all information that can be determined by a tagger based on n-grams, such as the POS.

In order to detect problems in the lexicon that concern short-range lexical information, we use a specific POS tagger [3,4]. The idea is the following. Let us

consider a sentence that is non-parsable because of a problem in the lexical entries for one of its form. A tagger might be able to guess for this form relevant short-range information which is missing or erroneous in the lexicon, based on the context in which it occurs. Comparing this "guessed" short-range information with the corresponding information in the lexicon might reveal relevant discrepancies. To achieve this, we apply a POS tagger to the sentence several times; each time, one of the forms that might be concerned by lexical shortcomings (usually, open-class forms) is considered as an unknown word, so as to bypass the tagger's internal lexicon. This allows the tagger to output tags that are compatible with the context of the form, including tags that might lack in the lexicon.

Of course, taggers do make errors. We reduce this problem by two different means. First, we take into account the precision rate $prec_t$ of the tagger for a tag t, as evaluated w.r.t. its training corpus. Second, we smooth the propositons of the tagger by averaging them on all sentences that are non-parsable because of a shortcoming in the lexicon. More precisely, we assign a global *short-range suspicion rate* $S_{\text{sr}}(w)$ to each relevant form w, defined as follows:

$$S_{\text{sr}}(w) = \frac{n_{wt} \cdot prec_t}{n_w} \cdot \log(n_{wt} \cdot prec_t). \quad (1)$$

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 in the non lexically parsable sentences.

3.2 Statistical approach for detecting lexical shortcomings

This lexical shortcomings detection technique, fully described in [5,6], relies on the following assumptions:

- The more often a lexical form appears in non-parsable sentences and not in parsable ones, the more likely its lexical entries are to be erroneous or incomplete [7];
- This suspicion rate $S(w)$ must be reinforced if the form w appears in non-parsable sentences along with other forms that appear in parsable ones.

This statistical computation quickly establishes a relevant list of lexical forms suspected to be incorrectly or incompletely described in the lexicon. The advantage of this technique over the previous one is that it is able to take into account all the syntactic information that is available in the lexicon, provided it is used by the tagger (e.g., subcategorization frames). However, it directly depends on the quality of the grammar used. Indeed, if a specific form is naturally tied with some syntactic construction that is badly covered by the grammar, this form will mostly be found in non-parsable sentences and will thus be unfairly suspected.

This problem can be overcome in at least two ways. First, we exclude from the statistical computation all sentences that are non-parsable because of shortcomings of the grammar (as decided by the classifier defined in the previous

section). Second, as already described in [5], we combine parsing results provided by various parsers that rely on different formalisms and grammars and thus, with different coverage lacks.

4 Generating lexical correction hypotheses: parsing non-parsable sentences

Depending on the quality of the lexicon and the grammar, the probability that both resources are simultaneously erroneous about how a specific form is used in a given sentence can be very low. If a lexically and grammatically valid sentence can not be parsed because of a suspected form, it implies that the lexicon and the grammar could not find an agreement about the role this form can have in a parse for this sentence. Since some suspected forms have been previously detected, we believe some parsing failures to be the consequence of lexical problems about those forms. In order to generate lexical corrections, we study the expectations of the grammar for every suspected form in its associated non-parsable sentences. In a metaphorical way, we could say that we "ask" the grammar its opinion about the suspected forms.

To fulfill this goal, we get as close as possible to the set of parses that the grammar would have allowed with an error-free lexicon. Since we believe the lexical information of a form to have restricted the way it could have been part of a successful parse and led the parsing to a failure, we decrease those lexical restrictions by underspecifying the lexical information of the suspected form. A full underspecification can be simulated in the following way: during the parsing process, each time a lexical information is checked about a suspected form, the lexicon is bypassed and all the constraints are considered satisfied, i.e., the form becomes whatever the grammar wants it to be. This operation is achieved by changing the suspected form in the associated sentences to underspecified ones called *wildcards*.

If the suspected form has been correctly suspected, and if indeed it is the unique cause of the parsing failure of some sentences, replacing it by a wildcard allows these sentences to become parsable. In these new parses, the suspected form (more precisely, the wildcard that replaces it) takes part to grammatical structures. These structures correspond to "instanciated" syntactic lexical entries, i.e., lexical entries that would allow the original form to take part in these structures. *Those instantiated lexical entries are the information used to build lexical corrections.*

As explained in [8], using totally underspecified wildcards introduces too large an ambiguity in the parsing process. This often has the consequence that no parse (and therefore no correction hypothesis) is obtained at all, because of time or memory constraints, or that too many parses (and therefore too many correction hypotheses) are produced. Therefore, we add lexical information to the wildcards to keep the introduced ambiguity below reasonable limits. Unlike other approaches [9,10] which generate all possible combinations of lexical information and test only the most probable, we choose to add only POS to the wildcards and

to rely upon the parsers' ability to handle underspecified forms. The ambiguity introduced by our approach clearly generates a more important number of corrections hypotheses. However, as explained in section 5, this ambiguity can be handled, provided there are enough non-parsable sentences associated with a given suspected form.

In practice, the POS added to a wildcard depends on the kind of lexical shortcoming we are trying to solve, i.e., it is chosen according to the kind of detection technique that suspected the form. So far, we only used the tagger-based detection to validate new POS for a suspected form. Thus, when using this approach, we generate wildcards with the POS given by the tagger to the form. When using the statistical detection approach, we generate wildcards with the different POS present in the lexicon for the suspected form: we want to validate new syntactic structures for the form, without changing its POS.

5 Extracting and ranking corrections

The way correction hypotheses are extracted depends on how they are used. In a previous work [11], the corrections were extracted in the output format of the parser. Such an approach has three important drawbacks:

- One first need to understand the output format of the parser before being able to study the corrections hypotheses;
- Merging results produced by various parsers is difficult, although it is an efficient solution to tackle some limitations of the process (see Sect. 5.2);
- Some parts of the correction might use information that is not easy to relate with the format used by the lexicon (specific tagsets, under- or overspecified information w.r.t. the lexicon, etc.).

We thus developed for each parser a conversion module which extracts the instantiated lexical entry given to the wildcard in a parse and translate it from the output format of the parser to the format of the lexicon.

Natural languages are ambiguous, and so have to be the grammars that model them. Thus, the reader should note that even an inadequate wildcard might perfectly lead to new parses and thus provide irrelevant corrections. In order to take this problem into account and prepare an easier manual validation, the corrections hypotheses obtained for a given suspected form with a given wildcard are ranked according to the following ideas.

5.1 Baseline ranking: single parser mode

Within the scope of only one sentence, there is not enough information to rank corrections hypotheses. However, by considering simultaneously various sentences that contain the same suspected form, one can observe that erroneous correction hypotheses are randomly scattered. On the other hand, correction hypotheses that are proposed for various syntactically different sentences are more likely to be valid.

This is the basis of our baseline ranking metrics, that can be described as follows. Let us consider a given suspected form w. First, all correction hypotheses for w in a given sentence form a *group* of correction hypotheses. This group receives a weight according to its size: the more corrections it contains, the lower weight it has, since it is probably related to several *permissive* syntactic skeletons. Therefore, for each group, we compute a score $P = c^n$ with c being a numerical constant in $]0, 1[$ close to 1 (eg. 0.95) and n the size of the group. Each correction hypothesis σ in the group receives the weight $p_{g\sigma} = \frac{P}{n} = \frac{c^n}{n}$, which depends twice on the size n of group g.

We then sum up all the weights that a given correction hypothesis σ has received in all groups it appears in. This sum is its global *score* $s_\sigma = \Sigma_g p_{g\sigma}$. Thus, the best corrections are the ones that appear in many small groups.

5.2 Reducing grammar influence: multi-parser mode

As it is the case for the statistical detection technique, crossing the results obtained with different parsers allows to improve the ranking. Indeed, most erroneous corrections hypotheses depend on the grammar rules used to reparse the sentences. Since two parsers with two different grammars usually do not behave the same, erroneous corrections hypotheses are even more scattered. On the opposite, it is natural for grammars describing a same language to find an agreement about how a particular form can be used, which means that relevant corrections hypotheses usually remain stable. Corrections can then be considered less relevant if they are not proposed by all parsers. Consequently, we separately rank the corrections for each parser as described in section 5.1 and merge the results using an harmonic mean.

6 Manual validation of the corrections

Thanks to the previous steps, validating the corrections proposed by a given wildcard for a given suspected form is easy. Three situations might occur:

1. There are no corrections at all: the form has been unfairly suspected, the generation of wildcards has been inadequate or the suspected form is not the only reason for its associated parsing failures;
2. There are some relevant corrections: the form has been correctly detected, the generation of wildcards has been adequate and the form is the only reason for (some of) its associated parsing failures;
3. There are only irrelevant corrections: the ambiguity introduced by the wildcards on the suspected form has opened the path to irrelevant parses providing irrelevant corrections; if the grammar does not cover all the possible syntactic structures, there is absolutely no guarantee that we generate relevant corrections.

Consequently, if the aim of the correction process is to improve the quality of a lexicon and not just to increase the coverage of parsers that rely on it, such a process should always be semi-automatic (with manual validation) and not strictly automatic.

7 Related work

To our knowledge, the first time that grammatical context was used to infer automatically lexical information was in 1990 [12]. In 2006 [9,10], error minning techniques like [7] started to be used to detect erroneous lexical forms. The detection technique described in [5,6] and the tagger-based detection technique have been used so far mostly by ourselves [11,1], with convincing results. The idea of a preliminary classification/filtering of non-parsable sentences to improve the detection techniques has also not been considered much so far (Sec. 2).

Wildcard generation started to be refined in [8]. Since then, wildcards have been partially underspecified and limited to open-class POS. In [10], the authors use an elegant technique based on a maximum entropy classifier to select the most adequate wildcards.

Ranking corrections is a task usually accomplished through the use of maximum entropy classifiers like in [9,10]. However, the evaluation of correction hypotheses based on all sentences associated with a given suspected form (see Sect 5.1), without generalizing to the POS of the form, has never been considered so far.

It is worth mentioning that all previous work on correction hypotheses generation has been achieved with HPSG parsers, and that no results have been presented until 2005. Since then, apart from [5], nobody reported on merging results provided by various parsers to increase the relevance of correction hypotheses.

In [9], the author presents his results for each POS. For POS with a complex syntactic behaviour (e.g., verbs), it clearly appears that it is impossible to apply such a set of techniques fully automatically without harming the quality of the lexicon. And the results would be even worse if applied to corpus with sentences that are not covered by the grammar.

8 Results and Discussion

We now detail the practical context in which we performed our experiments. We give some correction examples and explicit for each important element of our chain what is done, what is to be completed and which results could be achieved.

8.1 Practical context

We use and improve a lexicon called the Le*fff*.[7] This wide-coverage morphological and syntactic French lexicon with more than 600 000 entries has been built partially automatically [13] and is under constant development.

In order to improve the quality of our correction hypotheses, we used two parsers based on two different grammars:

[7] Lexique des formes fléchies du français/Lexicon of inflected forms of French. See http://alpage.inria.fr/~sagot/lefff-en.html.

– 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 [14], and compiled into a parser by the DyALog system [15].
– The SxLFG-Fr grammar [16] is an efficient deep non-probabilistic LFG grammar compiled into a parser by SxLfg, a Syntax-based system.

We used a French journalistic corpus from *Le monde diplomatique*. It contains 280 000 sentences of 25 tokens or less for a total of 4,3 million of words.

8.2 Examples of corrections

Here are some examples of valid corrections found:

– *israelien* ("*Israeli*"), *portugais* ("*Portuguese*"), *parabolique* ("*parabolic*"), *pittoresque* ("*picturesque*"), *minutieux* ("*meticulous*") were missing as adjectives;
– *politiques* ("*politic*") was missing as a common noun;
– *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 expecting a mandatory direct object and a mandatory attribute.
– *livrer* ("*to deliver*") did not handle constructions like *to deliver (something) to somebody.*

8.3 Classification of non-parsable sentences

For time reasons, results described in this section have been obtained thanks to a previous version of the classification technique: the POS and forms used for parsable sentences were not extracted from the parser outputs but built thanks to a tagger, just like for non-parsable sentence. Therefore, the model learned by the maximum entropy classifier is not optimal.

We chose to use 3-grams generated from the list of POS and forms for each sentence as well as a start-of-sentence element at its beginning and a end-of-sentence one at its end.

To evaluate the relevance of this technique, we kept 5% of all parsable sentences for evaluating the maximum entropy classifier. Let us recall that this classifier distinguishes sentences that are non-parsable because of shortcomings in the lexicon from all other sentences (parsable or non-parsable because of shortcomings in the grammar). We checked if this classifier was actually classifying parsable sentences in the second class, as they should be. Since there is no difference when generating the 3-grams of parsable and non-parsable sentences, the figures that we get are likely to be close to the actual precision rate of the classifier on non-parsable sentences. These figures are described in table 1.

Session	0	1	2	3
Precision rate	92.7%	93.8%	94.1%	94.9

Table 1. Precision of the non-parsable sentence classification

After 3 correction sessions, the maximum entropy classifier is tagging around 80% of the non-parsable sentences as non-parsable because of shortcomings in the grammar. This sharp contrasts with the figures of table 1 on parsable sentences is an additional clue that this classifier performs satisfyingly.

The precision rate of the classifier raises as expected after each correction session. Indeed, the quality of its training data is improved by each session; in the training data, as explained in section 2, all non-parsable sentences are tagged as non-parsable because of shortcomings in the grammar, even those that are in fact non-parsable because of shortcomings in the lexicon. By correcting lexical shortcomings, the number of parsable sentences increases and many sentences that were incorrectly tagged in the training data become tagged as parsable. Since the quality of the training data could be improved by constructing the n-grams for the parsable sentences from the parser outputs, we believe the precision might increase even higher.

In the end, the 5% error rate (which prevents us from taking into account a few sentences that are non-parsable because of shortcomings in the lexicon) is not a significant problem, given the positive impact of this filtering step on our detection techniques. In addition, since there is no reason for a particular form to be more frequent than average in these incorrectly classified sentences, this can be balanced simply by increasing the size of the corpus.

8.4 Lexical shortcomings detection techniques

The tagger-based technique has evolved a lot recently. The first tests were conducted with a simple preliminary version. At that time, the technique was different on many points.

1. We were only looking for POS shortcomings.
2. We were opening the ambiguity for all open-class forms in a sentence at the same time, which brings unnecessary ambiguity. We now open the ambiguity for one form at a time.
3. We were applying the technique on the whole corpus, which brings a lot of false positives. Even if there might be true positives in the parsable sentences and in non grammatically parsable sentences, it is far more interesting to restrict the detection to the non lexically parsable sentences.
4. We were not considering the error rate associated with each guessed tag when ranking the suspects.

At that time, the results were less convincing as they are today, as far as quality is concerned. However, this beta version of the technique allowed us to correct

182 lemmas in the lexicon. We expect the results of the newly implemented version to be even better.

In practice, our tagger-based technique already exhibits many positive aspects. In particular, the set of sentences that are non parsable because of shortcomings in the lexicon for a given session is a subset of the corresponding set for the previous session. This means that this detection technique only needs to be applied once on a given corpus. We also noticed some drawbacks. First, it can only detect short range lexical shortcomings. Second, we get a non negligible amount of false positives.

The Statistical technique proved relevant from the very beginning and allowed us to correct 72 different lemmas. It detects all kinds of lexical shortcomings, and the ranking it computes is extremely consistent. On the other hand, the grammar must have large enough a coverage to provide a reasonable proportion of parsable sentences; the quality of the detection directly depends on that of the grammar. Moreover, during a session, some suspected forms can prevent other problematic forms from being detected; it is necessary to make several correction sessions for a same corpus until no fairly suspected form arises.

8.5 Correction generation and ranking

The overall accuracy of the correction hypotheses decreases after each correction session. Indeed, after each session, there are less and less lexical errors that need to be corrected: the quality of the lexicon reaches that of the grammar. Since we want to improve efficiently our lexicon, we demonstrate the relevance of the whole process by showing the increase of the 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 2 lists the number of lexical forms updated at 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 2. Lexical forms updated at each session

For all sessions but the second one, all correction sessions are based on the non-parsable sentence classification, the statistical detection and the correction generation. The second session has been achieved only thanks to the tager-based

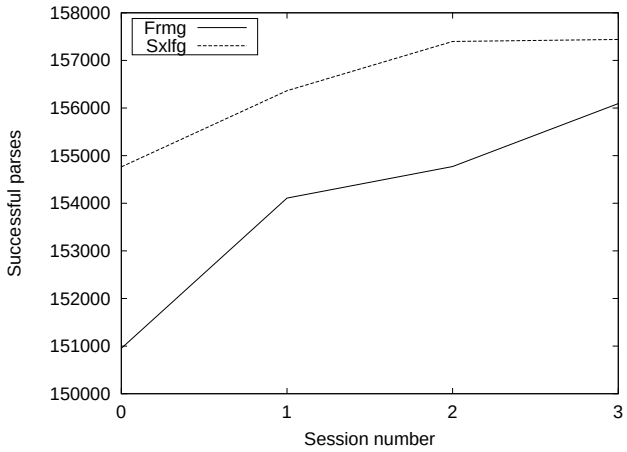

Fig. 1. Number of sentences successfully parsed after each session.

detection technique for identifying POS shortcomings (Sect. 3.1). As expected, we have been quickly limited by the quality of the grammars and the corpus. Indeed, the lexicon and the grammars have been developed together during the last few years, using this same corpus as a testing corpus. Therefore, on this corpus, there was not a huge gap between the coverage of our grammars and the coverage of our lexicon. Further correction and extension sessions only make sense after grammar improvements or if applied on new corpora.

However, the interaction between the grammar and the corpus can lead to complementary information: given a non-parsable sentence, if none of its suspected forms leads to a relevant correction, this sentence can be considered as lexically correct w.r.t. the current state of the grammar. This means that it exhibits shortcomings in the grammar, which can help improving it. Therefore, an iterative process which alternatively and incrementally improves both the lexicon and the grammar can be implemented. This is especially important given the fact that large scale French TreeBanks are rare.

To sum up our results, 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. Those results were achieved within only a few hours of manual work !

9 Future improvements

We are planning the following improvements to continue our research:

– We shall complete the evaluation of all components of the new model and prove their relevance separately.

– In order to pursue the improvement of the lexicon, we will extend our grammars thanks to the corpus of non-parsable sentences which now globally represents shortcomings of the grammars. During this process, we intend to develop some detection techniques to point out more precisely shortcomings in the grammar. The entropy model built by the maximum entropy classifier could be a good starting point.
– Semantically related lemmas of a same class tend to have similar syntactic behaviours. We could use this similarity to guess new corrections for some lemmas in a class where various other more frequent lemmas received the same correction.

10 Conclusion

In conclusion, the process described in this paper has three major advantages.

First, it does allow to improve significantly a morphological and syntactic lexicon within a short amount of time. We showed this thanks to the improvement of the parsing coverage of parsing systems that rely on such a lexicon, namely the Le*fff*. Moreover, our technique contributes to the improvement of deep parsing accuracy, which can be seen as a keystone for many avanced NPL applications.

Second, its iterative application on an input corpus eventualy turns this corpus into a global representation of the shortcomings of the grammar. Such a corpus could be an starting point for the development of a chain of tools dedicated to the improvement of deep grammars.

Third, an important advantage of our process is that it can be fed with raw text. This allows to use as an input any kind of text, including texts produced daily by journalistic sources as well as technical corpora. This is one of the techniques we are using to go on with the improvement the Le*fff*, in particular thanks to the 100 million words corpus of the French project Passage,[8] that combines fragments of the French Wikipedia, of the French wikisource, of the regional daily *L'Est Républicain*, of Europarl and of JRC Acquis.

Acknowledgments. The tagger-based detection technique could be achieved partially thanks to the support of Ministerio de Educación y Ciencia of Spain and FEDER (HUM2007-66607-C04-02), the Xunta de Galicia (INCITE08E1R104022ES, INCITE08PX IB302179PR, PGIDIT07SIN005206PR) and the "Galician Network for Language Processing and Information Retrieval" 2006-2009).

We would like also to thanks Olivier Lecarme, Carine Fédèle and Laurent Galluccio for their valuable comments.

[8] http://atoll.inria.fr/passage/

References

1. Nicolas, L., Sagot, B., Molinero, M.A., Farré, J., Villemonte de La Clergerie, E.: Computer aided correction and extension of a syntactic wide-coverage lexicon. In: Proceedings of Coling 2008, Manchester (2008)
2. Daumé III, H.: 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 2004)
3. Molinero, M.A., Barcala, F.M., Otero, J., Graña, J.: Practical application of one-pass viterbi algorithm in tokenization and pos tagging. Recent Advances in Natural Language Processing (RANLP). Proceedings, pp. 35-40 (2007)
4. Graña, J.: Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural (robust syntactic analysis methods for natural language tagging). Doctoral thesis, Universidad de A Coruña, Spain (2000)
5. Sagot, B., Villemonte de La Clergerie, É.: Error mining in parsing results. In: Proceedings of ACL/COLING'06, Sydney, Australia, Association for Computational Linguistics (2006) 329–336
6. Sagot, B., de La Clergerie, E.: Fouille d'erreurs sur des sorties d'analyseurs syntaxiques. Traitement Automatique des Langues **49**(1) (2008) (to appear).
7. van Noord, G.: Error mining for wide-coverage grammar engineering. In: Proceedings of ACL 2004, Barcelona, Spain (2004)
8. Barg, P., Walther, M.: Processing unkonwn words in hpsg. In: Proceedings of the 36th Conference of the ACL and the 17th International Conference on Computational Linguistics. (1998)
9. van de Cruys, T.: Automatically extending the lexicon for parsing. In: Proceedings of the eleventh ESSLLI student session. (2006)
10. Yi, Z., Kordoni, V.: Automated deep lexical acquisition for robust open texts processing. In: Proceedings of LREC-2006. (2006)
11. Nicolas, L., Farré, J., Villemonte de La Clergerie, É.: Correction mining in parsing results. In: Proceedings of LTC'07, Poznan, Poland (2007)
12. Erbach, G.: Syntactic processing of unknown words. In: IWBS Report 131. (1990)
13. Sagot, B., Clément, L., Villemonte de La Clergerie, É., Boullier, P.: The Lefff 2 syntactic lexicon for french: architecture, acquisition, use. In: Proceedings of LREC'06. (2006)
14. Thomasset, F., Villemonte de La Clergerie, É.: Comment obtenir plus des méta-grammaires. In: Proceedings of TALN'05. (2005)
15. Villemonte de La Clergerie, E.: DyALog: a tabular logic programming based environment for NLP. In: Proceedings of 2nd International Workshop on Constraint Solving and Language Processing (CSLP'05), Barcelona, Spain (October 2005)
16. Boullier, P., Sagot, B.: Efficient parsing of large corpora with a deep LFG parser. In: Proceedings of LREC'06. (2006)