Mining Parsing Results for Lexical Corrections

Lionel Nicolas, Jacques Farré, Éric Villemonte de La Clergerie

Mining Parsing Results for Lexical Corrections

Lionel NICOLAS*, Jacques FARRÉ*, Éric DE LA CLERGERIE†

*Team Langages, Laboratory I3S - UNSA + CNRS,
{lnicolas, jf}@i3s.unice.fr

†Project ATOLL, INRIA Rocquencourt,
Eric.De_La_Clergerie@inria.fr

Abstract

Successful parsing depends on the quality of the underlying grammar but also on the correctness of the lexicon that feeds the parser. The development of a lexicon both complete and accurate is an intricate and demanding task. A first step towards the improvement of a lexicon consists in identifying potentially erroneous lexical entries, for instance by using error mining techniques on large corpora (Sagot and de La Clergerie, ACL/COLING 2006) This paper explores the next logical step, namely the suggestion of corrections for those entries. This is achieved by running new analysis on the sentences rejected at the previous step, after having modified the information carried by the identified lexical entries. Afterwards, a statistical computation on the parsing results exhibits the most relevant corrections.

1. Introduction

Obtaining both accurate and wide coverage linguistic resources is a time-consuming and complex task, which can be alleviated through the use of tools. We present one such tool that attempts to correct potentially erroneous lexical entries in a semi-automatic fashion.

We focus on methods to improve the completeness and correctness of a lexicon starting from a set of lexical forms assumed to be non-reliable. Our work extends an *error mining* technique on large corpora which automatically detects suspicious forms (van Noord, 2004). More precisely, we pursue the work begun by (Sagot and de La Clergerie, 2006), whose results are confirmed by ours.

This detection technique relies on the following remark: given a large corpus of valid sentences, the more a form (and indirectly its underlying lemmas) appears (or not) in non-parsable sentences, the more likely (or unlikely) its features are to be incorrect. Nevertheless, we have to mitigate this assumption by considering contexts: a form is even more suspicious if it appears in non-parsable sentences along with forms that appear in parsable ones.

Our work, as well as the previous detection technique, is a mechanism of *feedback on error*. By this term, we mean tools that analyze errors generated by a program (here a parser) in order to improve its quality. For this purpose, the data given as input to the program must be as error-free as possible, in order to ensure that only the program is responsible for the errors. In the present case, our work is based on the errors generated through the processing (in 2005) of a corpus MD extracted from the French newspaper *Le monde diplomatique*. This corpus is composed of 570 000 sentences and 14.5 million words.

The previous detection step has built an ordered list of 5344 suspicious forms with, for each form f, a *suspicion rate* representing how much suspicious f is and a list of sentences (56089 overall) where f appears to be the culprit responsible for the parsing failures. If a form is actually responsible for those failures, and not the grammar,[1] then the information associated with this form (in fact with its underlying lemmas) is likely to be incorrectly or only partially described in the lexicon.

Therefore, by releasing or modifying the constraints on the features carried by the suspicious form, new parses of the sentences have more chances to succeed. The representations of the sentences produced by the new successful parses provide useful grammatical information for the suspicious form. These data typify the conditions in which the parses could be achieved, that is, the expectations of the grammar for the form. By sorting it on various sentences, we are able to provide useful correction hypotheses for the corresponding entries of the lexicon.

Although our examples and results are related to French, the method we present is fully system and language independent.

Practical Context The lexicon we are improving is the *Lefff* (Sagot et al., 2006). Partly built automatically, this morphosyntactic wide coverage French lexicon is under constant development and so far contains more than 520 000 entries. The parser we use is an hybrid TAG/TIG parser based on a grammar generated from a more abstract meta-grammar FRMG, ***Fr**ench **M**eta-**G**rammar*, with 134 highly factorized trees (Thomasset and de La Clergerie, 2005). In 2005, FRMG coupled with the *Lefff* lexicon gave a full parse coverage rate around 41% on the MD corpus.

Related Work Acquisition of lexical knowledge from natural (i.e. not annotated) corpora using some grammatical knowledge has been first studied by Brent (1993) in order to infer the syntactic frames of English verbs. The *Lerner* system designed by Brent is based on two modules: the observation collecting module and the statistical modeling one. The observation module, which has a partial knowledge of the english grammar, only performs a

[1] We suppose that failures caused by the previous parse steps (segmentation, punctuation, detection of named entities, ...) have been identified. We do not take in account the erroneous forms and their associated sentences caused by such errors.

surface analysis.

Noticing that the surface analysis used in *Lerner* disregards a large proportion of sentences and put extra burden on the statistical computations, Horiguchi et al. (1995) suggested that the observation module should rely on a more structurally complex knowledge of the language. They use the results of the unification-based parses of an HPSG-based system in order to acquire lexical entries of unknown japanese content words. However, function words are given manually-coded lexical entries.

The repair of lexical information shortfall for robust parsing, as studied by Grover and Lascarides (2001) and by Crysmann et al. (2002), should also be mentioned.

We first explain how to generate correction hypotheses (Sect. 2.), how to order (Sect. 3.) and how to observe them (Sect. 4.). Then, we present the practical results we obtained and outline the improvements we planned for the future (Sect. 5.).

2. Hypotheses Generation

The main goal of a parser is to check the syntactic correctness of a sentence and to produce one or more representations of it. In general, we intend to avoid overgeneration, that is, to generate as few representations as possible.

We say that a sentence is *ambiguous* for a parser when its parse allows several interpretations. This happens mostly for two reasons:

1. The sentence is naturally ambiguous, and additional information, such as the semantic context, is needed in order to filter the interpretations.

2. The resources (lexicon, grammar...) are not restrictive enough and accept a wider language than intended.

In order to reject sentences that do not belong to the language, we need the lexicon to be as accurate as possible. The most specified a lexical form is, the less combinations with the other constituents of the sentence are allowed, and therefore, the less incorrect interpretations are permitted.

2.1. Causes of Parsing Failure

As we pointed out previously, if a lexical form is actually responsible for a parsing failure, then the features it carries are erroneous. Each form, through its lemmas, has three kinds of information, divided into two sets: firstly the syntactic category (noun, verb ...), secondly the morphological features (number, gender, person, tense, mood ...) and the syntactic features (subcategorization frame, pronomalization, diathesis ...). A parsing failure due to a form is the consequence of a problem related to at least one of those sets.

In the following sections, we focus mostly on errors caused by the syntactic features and category. We will discuss errors related to morphological features or to the grammar later on.

2.1.1. Category Defect

A single lexical form can be associated with several lemmas with distinct syntactic categories. Therefore, the input of the parser must be a lattice of words (Sagot and Boullier, 2005), or DAG, as illustrated by Fig. 1. A successful parse validates at least one path in this DAG.

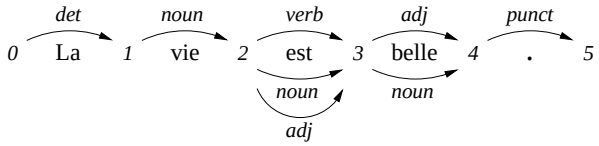

Figure 1: DAG structure given in input to a parser

However, the lexicon may not contain all the homonyms of a form, and this can lead to parsing failures. For instance, the French word “fiche” is a noun and a flexion of the verb “ficher”. If no entry with a “noun” category is present, the sentence “Ma fiche contient une erreur”/*My card contains an error* will be represented by the single sequence *ma/possessive-pronoun fiche/verb contient/verb une/det erreur/noun*. Hopefully, there should not be any grammatical rule to accept it, and its parse should fail.

2.1.2. Overspecification

Usually, grammar rules are enriched with feature equations, in order to further constrain the language recognized by the grammar backbone (Abeillé, 1993). These equations are also used to establish syntactic dependencies between forms in the computed representations of the sentence (see Sect. 2.2.2.).

As told before, having the best possible specification (restriction) of lexical forms helps reduce ambiguity and produce less representations for a sentence. Nonetheless, if they are too restrictive, i.e. overspecified, then feature unification on the syntactic backbone might fail. This issue arises with scarce uses of a form for which some features should have been marked as optional. For instance, the exhaustive enumeration of all the possible uses of a verb is a tedious task, because of polysemy, optional arguments, possible verbal alternations (“to buy something” will give “something is bought”), and various realizations of the arguments. This statement can be extended to the other syntactic categories, provided that they are also given subcategorization frames. Consequently, some aspects may sometimes be marked as mandatory when, in some cases, they are only optional.

2.2. Re-parsing Non-parsable Sentences

The previous step of detection of suspicious form only keeps the non-parsable sentences. Thus, their parsing rate is null. If changing some features of a suspicious form f allows their parsing rate to increase noticeably, it seems natural to assume that f was responsible for the previous failures. Our objective is to identify which changes are the cause of the rate increase. Rather than testing all possible combinations of changes, which is exponential and thus inefficient, we rely upon the ability of our parser to handle underspecified forms.

Once new parsing results on various sentences are obtained, we are able to extract and compute correction hypotheses (see Sect. 2.2.2.).

2.2.1. Joker Generation and Use

In order to collect new parsing results from initially non-parsable sentences, we introduce special underspecified lexical forms called *jokers*. In the current approach, each joker only carries a syntactic category, excluding *closed* categories (pronoun, determinant, ...). Thus, a joker is not constrained with fixed morphological or syntactic informations, and always fulfills the grammar requirements. Its substitution to the suspicious form in a sentence clearly enhances the chances of successful parse. Nevertheless, it can introduce a noticeable ambiguity because no feature-filtering can be done on the joker.

Since, so far, we are not able to know which kind of error (overspecification or category defect) is responsible for the parsing failures, we consider both simultaneously.

In order to handle an overspecification, we use a joker with the same syntactic categories as the exchanged suspicious form. Doing so, we allow the parser to explore the same grammar rules as the ones used for the initial sentences without being bothered by feature equations.

In order to deal with a category defect, we use jokers with syntactic categories distinct from the ones of the suspicious form exchanged. Thus, the parser will explore new grammar rules. We choose these categories according to the informations provided by a stemmer or by a part-of-speech tagger like TREETAGGER (Schmid, 1999).

One could ask why we do not use a single joker without any syntactic category, hence without any information? Indeed, such a joker would simultaneously cover all the situations described above. The reason is that it would introduce a substantial ambiguity, leading, in most cases, to parsing failures caused by timeouts or memory shortage, or to an overgeneration of interpretations. In the first case, we do not collect any information; in the second case, the huge amount of data prevents its efficient analysis. Our solution allows us to avoid in most cases these problems, while keeping the number of jokers to test within reasonable limits.

During our experiments, we tested 2.05 jokers per suspicious form (10 978 all in all), resulting in 117 655 parses.

2.2.2. Syntactic Signature Extraction

Assuming that a suspicious form has been correctly identified by the error mining step, exchanging it with jokers in the associated sentences allows some parses to succeed. In fact, we observe that the success rate for the modified sentences increases in a coordinated way with the suspicion rate of the suspicious form exchanged (see Sect. 5.).

The parser we use returns the set of all possible interpretations for the parsed sentence as a shared dependency forest (see Fig. 2). This forest is available in XML format, allowing easy manipulation. In such forests, vertices represent the lemmas and edges represent the syntactic dependencies between lemmas. Every dependency has a governor source lemma and a governee target lemma. All the informations about the form, the category and the anchored grammatical production are provided for each lemma. A dependency always receives a kind and a label. The label often (but not always) indicates the syntactic function of the target (subject, object, ...). The kind provides information about the nature of the target (argument, adjunct, co-anchor, ...). In order to manage ambiguities, additional information, local to the governor, links lemmas and dependencies to one or more interpretations.

For instance, the French sentence "la vie est belle" (Fig. 2) has various interpretations. This is due to the ambiguity of "est" as a copula verb, an apposed noun and an adjective. Furthermore, we also have the ambiguity of "belle" as an adjective and a noun.

The forests contain incoming and outgoing dependencies to and from the joker. We will use the terms :

- *syntactic signature* for the set of dependencies around the joker for a given interpretation of a sentence,
- *group of signatures* for the set of all possible syntactic signatures for a given sentence.

Those signatures represent the conditions in which the parse could be achieved; in other words, it represents the data the grammar would have accepted for the form. Because of the ambiguity we introduced, the parser can produce several interpretations and thus several signatures. Among these interpretations, there is one that fits the true meaning of the sentence better than any other. Therefore, its corresponding signature contains the most relevant and interesting data. This is the data we need in order to infer the corrections we should apply to the lexicon.

Within only one group (produced from one sentence) we are unable to promote the signatures: there is no way to differentiate the relevant signatures from the ones that are just a consequence of the ambiguity we introduced.

The variability of contexts represented by several groups of signatures (produced from several sentences) bring us a solution to this problem: this variability implies the diversification of non-relevant signatures, which contrasts with the stability of the relevant ones that correspond to the correct sense of the form. Thus, a clear repetition of particular signatures spanning several groups of signatures obtained from different sentences undeniably suggests one or more schemes expected by the grammar for the form.

3. Ordering and Promoting the Signatures

We promote/devalue the signatures through a simple two-step statistical computation.

First step: local distribution of points

The interest we have for a group of signatures depends on its size: the more signatures it contains, the less interest it has, since it is probably related to several *permissive* syntactic skeletons, as shown by the interpretations represented in Fig. 2. So, for each group, we compute a score $P = c^n$ with c being a numerical constant in $]0, 1[$ (eg. 0.95) and n the size of the group.

All the signatures of a group are equally interesting, thus, we spread the score of the group equally between them. In the end, each signature receives a score $p_g = \frac{P}{n} = \frac{c^n}{n}$, which depends twice on the size n of group g.

Second step: global score computation

Once the previous step is complete, we add the scores received by a same signature σ in the different groups where

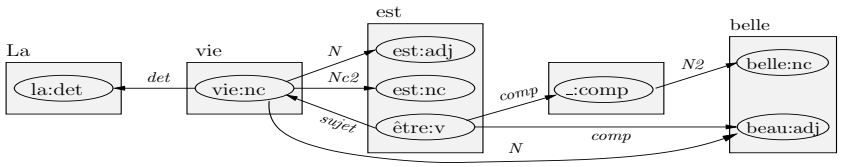

Figure 2: a shared forest of dependencies.

it appears in order to compute its global score $s_\sigma = \Sigma_g p_g$. Thus the best signatures, the ones that appear in various groups and in small groups, receive a greater score s_σ.

4. Signature Study

The technique is a supervised one: the system suggests corrections that may be explored through an adequate interface and the lexicon maintener decide to apply (or not) these suggestions Although the classification resulting from the previous computation is the most important point when studying the signatures, there are side aspects that we also need to take into account.

4.1. Inclusion Between Signatures

If a signature A includes a signature B (all the information carried by B are also in A), the A's added information may represent some optional aspects of the common information shared by both signatures. If both A and B have high scores s_A and s_B, we need to consider this phenomenon in order to avoid an incomplete correction if $s_B > s_A$, an overspecified correction if $s_A > s_B$.

This is why an inclusion graph should be computed and observed when analyzing signatures, in order to avoid fixing only partially an error.

4.2. Lexicon and Grammar Synchronization

The lexical changes suggested by this technique are not always unquestionable. Indeed, this technique allows the grammar to express its expectations about the suspicious forms. If the grammar is not fully correct, neither will be the generated interpretations and, consequently, the extracted signatures. Therefore the quality of the corrections depends greatly on the quality of the grammar. Nevertheless, even incorrect signatures provide precious feedback about the shortcomings of the grammar.

This technique is a method that attempts to decrease the number of conflicts between a grammar and a lexicon in order to better "synchronize" them.

4.3. Impact of Jokers on the Parsing Rate

Unlike programming languages, natural languages are highly flexible and so have to be the grammars describing them. For instance, in French, one can easily use an adjective as a noun and conversely. This fact, combined with the ability of jokers to prevent conflicts during feature unification, can lead to successful parses of sentences even with a wrongly categorized joker. Thus, an irrelevant joker can still induce a set of signatures.

Therefore, it is important to observe the parsing rate a joker has implied before studying the signatures it has induced.

5. Results

In addition to the validation of our technique, our experiments strengthen the relevance of the detection technique with new results not present in the original paper by Sagot and de La Clergerie (2006).

5.1. Precision of the Error Mining Technique

The curve of Fig. 5.1. shows an obvious correlation between the best parsing rates achieved after having introduced jokers and the suspicion rates of the forms exchanged. Thus, it confirms the validity of the detection technique that provided the suspicious forms.

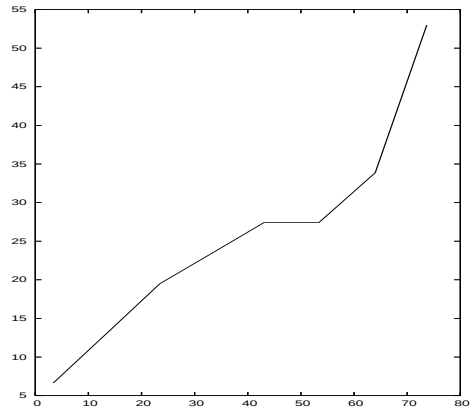

Figure 3: Parsing rates in percentages (Y-axis) according to the suspicion rates (X-axis) after joker introduction

The values used for this curve are averages after grouping the suspicious forms by range of suspicion rate. Without this grouping, the curve would present many variations making its observation difficult. There are two parasite phenomena that explain this variability:

- Some forms were suspected when the grammar was actually the true culprit, so exchanging them with jokers does not bring any parsing rate increase. Indeed, some forms are naturally related to some specific syntactic constructions, like for instance, the subject inversion in presence of some adjectives (as in "Rares sont ceux qui . . . "/*Rare are those that . . .*). Thus their suspicion rate has been unfairly increased.

- Introducing a joker can increase timeout and memory shortage rates noticeably, even if we impose a 40-word limit on the parsed sentences.

Because those two phenomena are not linked to specific suspicion rates, they can be observed at any level. Grouping the forms by range reduces their effects on the curve.

If a sentence associated with an highly suspicious form can not be parsed after several joker introductions (time-outs and memory shortages are not considered as failure), it seems reasonable to think that those failures were caused by shortcomings of the grammar. This provides precious data to analyze with a process such as a grammar induction mechanism.

5.2. Assessing Signature Quality

The study of the best jokers and best signatures has confirmed some doubts we had: the technique is still "young". Various phenomena could not allow us to correctly quantify the quality of the signatures. Nevertheless, we already know how to face most of them (see Sect 6.).

In the end, in many cases, the technique has proved to be very relevant and instructive and has permitted us to perform improvements to our tools, and not only to the lexicon. For instance, we found correct signatures, such as for "prospères"/*prosperous*. We can observe its use as an attributive adjective (shown by joker syntactic category and the signature), whereas it was only described as a verb in the *Lefff*. For the verbal form "révéler"/*to reveal*, the signatures illustrate the expectation of a copula argument as in "ce choix pourrait se révéler catastrophique."/*This choice could appear disastrous*. A reflexive aspect was missing in order to cover prepositional construction.

Even if incomplete, our approach to automatically suggest corrections for a lexicon has proved to be viable and we are willing to develop it further in order to obtain a fully functional tool. Once some improvements are achieved, new computation campaigns will be performed.

6. Future Improvements

During our experiments, we have established a list of problems to solve and solutions to fix them.

It is very frequent to be able to apply several grammar rules to a sequence of words, especially when the syntactic category of a word changes, like when we try various jokers. Nevertheless, rules are not used with the same frequency and, consequently, the resulting signatures do not carry the same amount of information. Once such data is obtained, we shall use this probabilistic aspect to balance the scores of the signatures.

Signatures should be cleaned to avoid irrelevant adjuncts on suspects. However, evaluating the importance of a given adjunct is difficult. For instance, an adverb modifying an adjective joker strengthens the adjectival hypothesis but a prepositional attachment does not bring much information about noun, adjective and verb jokers.

We often observe families of suspicious lemmas with a root stem in common with similar signatures, like for the various flexions of a verb. In these cases, the problems directly concern the root stem and not the forms. We could group those forms in order to increase the number of associated sentences. Still, we have to keep in mind that some problems are only related to some specific forms.

We should group signatures that represent a common syntactic phenomenon under various aspects by establishing a list of equivalent combinations of dependencies. For instance, the subject and other verbal arguments with various realizations (nominal, clitics, relatives and pronouns, ...) or a transitive verb used in a passive construction.

The lack of morphological information in the signatures has lead, in few cases, to signatures equivalent to the data present in the lexicon and thus did not help to understand the problem. The next logical step for the signature model will be to assimilate this available information.

7. Conclusion

Our experiments strengthen the ability of the detection algorithm to correctly identify true suspicious forms. Exchanging them with jokers has proved to noticeably increase the parsing rate.

They also validates our mechanism for automatically suggesting lexical corrections for the suspicious forms (i.e. to their underlying lemmas) and bringing precious feedback on several shortcomings of the grammar.

Still, work remains to be carried out in order to refine the quality of the suggested corrections by distinguishing what is relevant from what is not. We have suggested some means to this end.

Acknowledgement. The authors are grateful to Sylvain Schmitz and Angelica Lim for their valuable comments and suggestions.

References

Abeillé, Anne, 1993. *Les nouvelles syntaxes, grammaire d'unification et analyse du français*. Armand Colin.

Brent, Michael, 1993. From grammar to lexicon: unsupervised learning of lexical syntax. *Comput. Linguist.*, 19(2):243–262.

Crysmann, Berthold et al., 2002. An integrated architecture for shallow and deep processing. In *Proceedings of the 40th Annual Meeting of the ACL*.

Grover, Claire and Alex Lascarides, 2001. XML-based data preparation for robust deep parsing. In *Meeting of the Association for Computational Linguistics*.

Horiguchi, K., K. Torisawa, and J. Tsujii, 1995. Automatic acquisition of content words using an HPSG-based parser. In *Proceedings of NLPRS'95.*.

Sagot, Benoît and Pierre Boullier, 2005. From raw corpus to word lattices: robust pre-parsing precessing. In *Proc. of L&TC*. Poznan, Pologne.

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

Sagot, Benoît and Éric de La Clergerie, 2006. Error mining in parsing results. In *Proceedings of ACL/COLING'06*. Sydney, Australia: Association for Computational Linguistics.

Schmid, Helmut, 1999. Probabilistic part-of-speech tagging using decision trees. *IMS-CL.*.

Thomasset, Fran cois and Éric de La Clergerie, 2005. Comment obtenir plus des méta-grammaires. In *Proceedings of TALN'05*. Dourdan, France: ATALA.

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