# Deep non-probabilistic parsing of large corpora

**Benoît Sagot and Pierre Boullier**

INRIA, Projet Atoll
Domaine de Voluceau, Rocquencourt, B.P. 105
78153 Le Chesnay, France
{benoit.sagot, pierre.boullier}@inria.fr

**Abstract**

This paper reports a large-scale non-probabilistic parsing experiment with a deep LFG parser. We briefly introduce the parser we used, named SXLFG, and the resources that were used together with it. Then we report quantitative results about the parsing of a multi-million word journalistic corpus. We show that we can parse more than 6 million words in less than 12 hours, only 6.7% of all sentences reaching the 1s timeout. This shows that deep large-coverage non-probabilistic parsers can be efficient enough to parse very large corpora in a reasonable amount of time.

## 1. Introduction

The parsing of large corpora is usually performed with surface stochastic parsers. Indeed, it is usually thought that deep parsers, especially when they do not rely on probabilistic models, are not efficient enough to parse multi-million word corpora in a reasonable amount of time.

However, this paper reports experiments on parsing a large raw French journalistic corpus (5.5 million tokens) with a deep non-probabilistic parser that relies on the LFG formalism. Parsing such a large corpus with a sophisticated formalism such as LFG requires of course a very fast parser. For these experiments, we used SXLFG, an efficient LFG parser described in (Boullier and Sagot, 2005). We were able to parse the whole corpus in only 15 hours of effective parsing time, with only 12.6% of sentences reaching the 1s-timeout.

The LFG for French used in this experiment is still under development. Its current version allows us to give a complete and consistent parse (in the sense of LFG) for 53.4% of all sentences. Moreover, error recovery mechanisms at all levels allows us to build incomplete, inconsistent or partial parses for the remaining sentences.

## 2. Parser, grammar and lexicon

The experiment reported here was performed with the SXLFG parser generator and a large-coverage LFG grammar for French.

SXLFG is a new LFG parser generator (Boullier and Sagot, 2005) that relies on a two-stage architecture: the first step is performed by a context-free parser that gathers all possible constituent structures for the input sentence into a shared parse forest. Then functional structures are evaluated on this forest.

More precisely, the context-free parser that is the core of SXLFG is Earley-like parser that relies on underlying left-corner tables and is an evolution of (Boullier, 2003). The set of analyses produced by this parser is represented by a shared parse forest. In fact, this parse forest may itself be seen as a CFG whose productions are instantiated productions of the CFG backbone. The evaluation of the functional equations is performed during bottom-up traversals of this forest. A disambiguation module, which discards unselected f-structures, may be invoked on any node of the forest, both on its root node (*global disambiguation*) and on selected internal nodes (*partial disambiguation*). The list of individual heuristics that are applied depend on the name of the node (the corresponding non-instantiated non-terminal symbol).

The input of the parser is a DAG of inflected forms (all forms being known by the lexicon, including special forms representing unknown tokens in the raw text). This lattice is converted by the *lexer* in a lexeme lattice (a lexeme being here a CFG terminal symbol associated with underspecified f-structures).

Apart from the use of partial disambiguation, parsing efficiency is achieved thanks to several techniques such as compact data representation, systematic use of structure and computation sharing and lazy evaluation. We also use and heuristic and almost non-destructive pruning during parsing.

Moreover, various robustness techniques are applied both at the constituents level and at the functional level (CFG error recovery, robust computation of functional structures,...). When no f-structure is found, or when the timeout is reached, we can launch an *over-segmentation* mechanism that splits the sentence into smaller parts. This mechanism has 5 possible levels of granularity, so as to ensure that the parser gives an output for all input sentences. These techniques allow to gather in almost all cases (partial) useful information.

The experiment reported here uses approximately the same grammar as in (Boullier and Sagot, 2005), which is an evolution of the grammar developed by Lionel Clément for his XLFG system (Clément and Kinyon, 2001). It is a large-coverage grammar for French which contains 251 rules and 894 functional equations. Recent (yet unpublished) experiments on a smaller journalistic corpus for which a chunked reference is available[1] have led for labeled chunks, with the same grammar and the same parser, to a precision of 73.2% and a recall of 74.5%. This shows that the grammar is large-coverage but must still be improved.

The lexicon we used is the latest version of (Lex-

[1] The "general_lemonde" corpus which is one of the 43 corpora used during the French parsers evaluation campaign named EASy.

ique des formes fléchies du français[2]) (Sagot et al., 2006), which contains morphosyntactic and syntactic information for more than 500,000 entries corresponding to approximately 400,000 different tokens (words or components of multi-word units).

## 3. Corpus and pre-parsing processing

The corpus we parsed in this experiment is a large French journalistic corpus consisting of more than 6 million tokens of the *Monde diplomatique* (a token being defined as a sequence of characters separated by a white space, after having added white spaces around punctuation marks[3]). It is a raw corpus (i.e., it includes all meta-information, footnotes, typographic signs, and so on).

To be able to parse such a raw corpus, we need, as said before, to transform it into a correct input for the parser, i.e., an (ambiguous) lattice of known words. This task was performed with the SXPipe pre-parsing processing chain (Sagot and Boullier, 2005). This system includes sequentially named-entity recognition, tokenization and sentence boundaries detection, lexicon-aware named-entity recognition, spelling correction, and non-deterministic multi-words processing, re-accentuation and un-/re-capitalization.

The result of this processing is a set of 300,000 sentences, each sentence being represented by a word lattice. The average sentence length is 21.3 words, and a repartition of sentences lengths is shown in Figure 1. The whole set of lattices include approximately 7.5 million transitions (the average amount of transitions per input token is 1.2).

Note that no tagging is performed before parsing.

## 4. Results

SXLFG was able to parse the whole 300,000 sentences in approximately[4] 42,000 seconds (11,7 hours).[5]

To get an idea of the ambiguity of the CFG grammar underlying our LFG grammar, Figure 4 shows the median number of CF parses given the number of transitions in the lattice representing the sentence. To illustrate the efficiency of the CFG parser, the highest number of trees in a parse forest (which didn't need error recovery) is as high as $8,8\,10^{45}$. The CFG parsing of the corresponding 143-word sentence needs only 0.05s.

However, SXLFG manages to build very efficiently full or partial parses for most sentences in less time than the 1s timeout that has been set. To show the efficiency of our f-structures computation module, independently from the CFG ambiguity of the grammar, Figure 3 plots the total parsing time, including the evaluation of features structures, against the number of trees produced by the CF parser.

Coverage results are given in Table 1. They show that our grammar is indeed a large-coverage grammar, since more than 60% of the corpus is successfully parsed, despite of the fact that it is a deep LFG grammar.

They show also that only 6.7% of all sentences are not already parsed before the 1 second timeout. A 0.5s timeout would have allowed to parse the whole 6 million word corpus in only 32,500s (9 hours)[6] with only 10.1% of sentences reaching the timeout. A more aggressive 0.1s timeout leads to a total parsing time of 13,000s (3.6 hours), 23.8% of sentences remaining unparsed.

## 5. Conclusion

We have shown that it is possible to parse large raw corpora with a deep non-probabilistic large-coverage parser such as SXLFG, which builds complex and linguistically relevant syntactic structures. Indeed, we were able to parse a French journalistic corpus of more than 6 million words in less than 12 hours with an LFG parser, only 6.7% of all sentences reaching the 1s timeout. More aggressive timeouts lead to even lower total parsing times.

This allows to build in a few hours large corpora with complex syntactic annotation. We are beginning to make use of such corpora to automatically detect erroneous and missing information in our resources, to train statistical taggers and hypertaggers, and to learn syntactic and semantic collocations. These are only some of the possible applications to such corpora.

## 6. References


Boullier, Pierre, 2003. Guided Earley parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT'03)*. Nancy, France.

Boullier, Pierre and Benoît Sagot, 2005. Efficient and robust LFG parsing: SXLFG. In *Proceedings of IWPT'05*. Vancouver, Canada.

Clément, Lionel and Alexandra Kinyon, 2001. XLFG – an LFG parsing scheme for French. In *Proceedings of LFG'01*. Hong Kong.

Sagot, Benoît and Pierre Boullier, 2005. From raw corpus to word lattices: robust pre-parsing processing. In *proc. of the 2nd Language & Technology Conference (LT'05)*. Poznan, Poland. Selected for journal publication.

Sagot, Benoît, Lionel Clément, Éric de La Clergerie, and Pierre Boullier, 2006. The Le*fff* 2 syntactic lexicon for French: architecture, acquisition, use. In *Proceedings of LREC 06*. Genova, Italy. To be published.


[2]Lexicon of French inflected forms

[3]But not around characters such as dots or commas when they are not used as punctuation marks

[4]We can not be very precise for the following reason: the granularity of our time measurement is 10ms. Hence, a sentence with a parsing time of 20ms was parsed in fact in 20 to 29ms. Therefore, we added 5ms to all parsing times lesser than the timeout, which led to a total parsing time of 42,097s. What is sure is that the exact total parsing time is between 40,698s and 43,497s.

[5]We performed this experiment on a on an AMD Athlon 2100+ architecture (1.7 GHz) running Linux.

[6]See footnote 4.

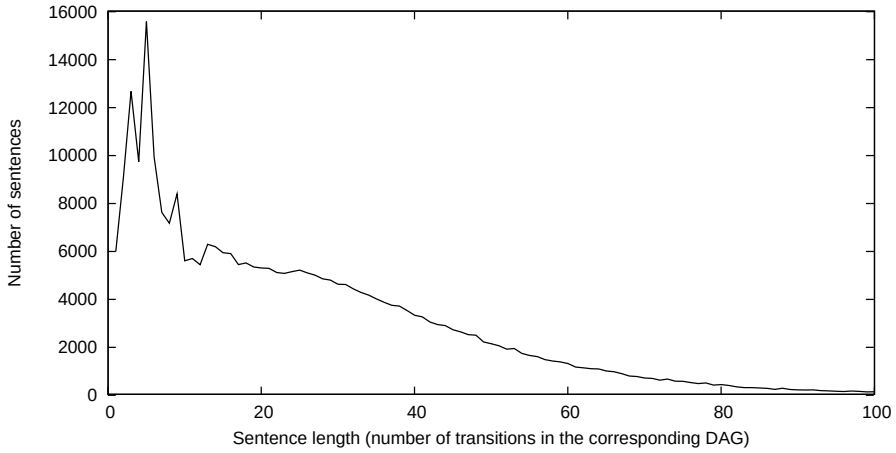


Figure 1: Repartition of sentences of the test corpus w.r.t. their length. We show the cardinal of classes of sentences of length $10i$ to $10(i+1)-1$, plotted with a centered $x$-coordinate ($10(i+1/2)$).

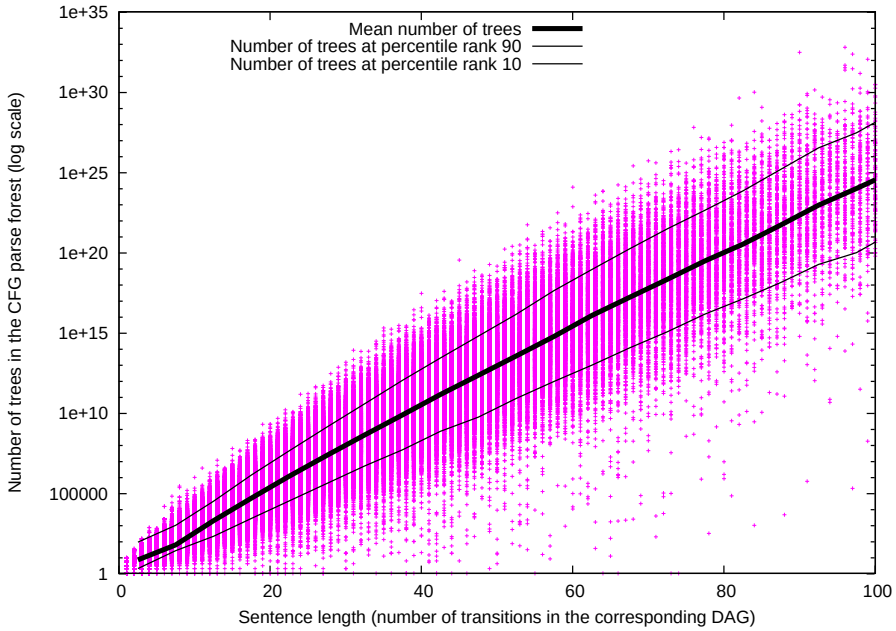


Figure 2: CFG ambiguity (medians are computed on classes of sentences of length $10i$ to $10(i+1)-1$ and plotted with a centered $x$-coordinate ($10(i+1/2)$).

| Total number of sentences | 300,000 | |
|---|---|---|
| Recognized by the CFG backbone | 290,827 | 96.9% |
| CFG parsing required error recovery | 9,173 | 3.1% |
| Complete and consistent f-structure | 181,948 | 60.4% |
| Almost complete and consistent f-structure | 23,055 | 7.7% |
| Partial f-structures | 68,078 | 22.7% |
| No f-structure found (over-segmentation launched) | 6,769 | 2.3% |
| Parser error (to be fixed) | 11 | 0.004% |
| Timeout (1s) | 20,190 | 6.7% |

Table 1: Coverage results for SXLFG on a French journalistic corpus of 5.5 million tokens. Completeness and consistency are standard LFG notions. We say that a sentence is *almost complete and consistent* if all strict sub-structures of the main f-structure (the f-structure associated to the root of the constituency tree) are complete and consistent, but if the main f-structure itself is not.

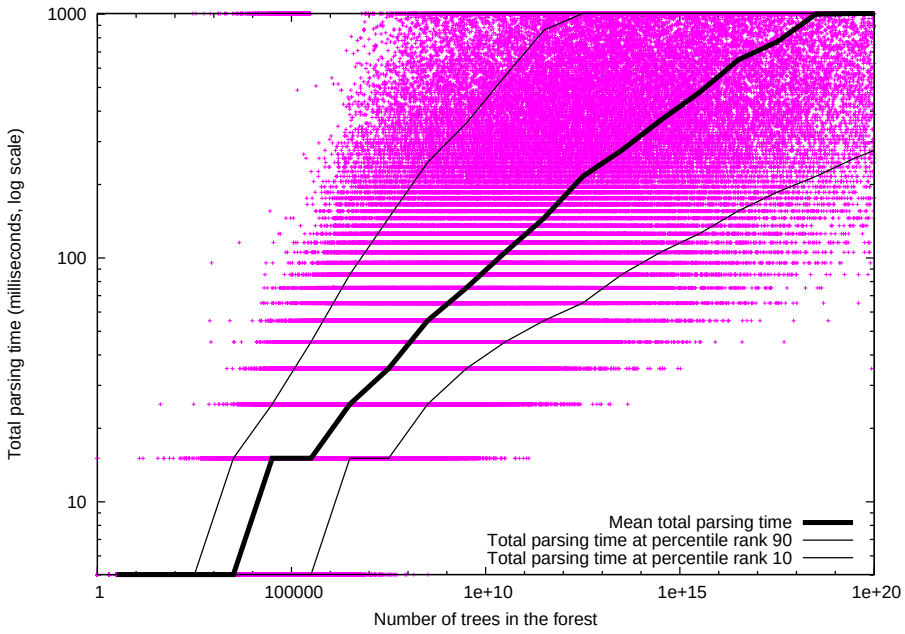


Figure 3: Total parsing time w.r.t. the number of trees in the forest produced by the CF backbone (medians are computed on classes of sentences whose number of trees lies between $10^{2i}$ and $10^{2i+2} - 1$ and plotted with a centered $x$-coordinate ($10^{2i+1}$)). As explained in footnote 4, we added 5ms to all parsing times lesser than the timeout, because of the 10ms granularity of the parsing time measurement.

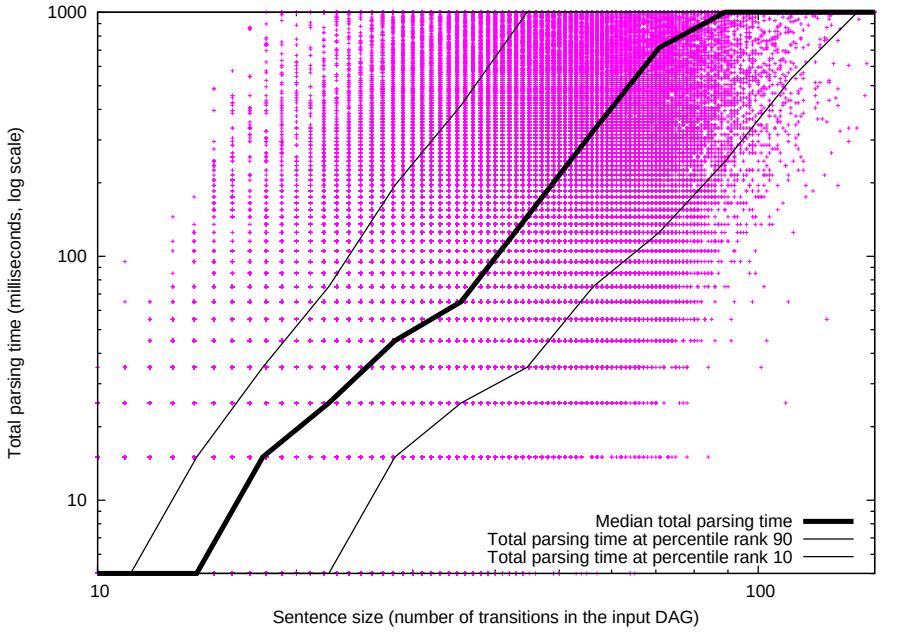


Figure 4: Total parsing time w.r.t. the length of the sentence (number of transitions in the input DAG of the sentence ; medians are computed on classes of sentences whose number of trees lies between $10^{2i}$ and $10^{2i+2} - 1$ and plotted with a centered $x$-coordinate ($10^{2i+1}$)). See also footnote 4. Note that these results measure simultaneously the grammar's characteristics and SXLFG's performance.