Semi-automated Extraction of a Wide-Coverage Type-Logical Grammar for French

Richard Moot

TALN 2010, Montréal, 19–23 juillet 2010

Semi-automated Extraction of a Wide-Coverage Type-Logical Grammar for French*

Richard Moot
LaBRI (CNRS, Bordeaux) & SIGNES (INRIA Bordeaux SW)
351 cours de la Libération, 33405 Talence, FRANCE
Richard.Moot@labri.fr

Abstract. The paper describes the development of a wide-coverage type-logical grammar for French, which has been extracted from the Paris 7 treebank and received a significant amount of manual verification and cleanup. The resulting treebank is evaluated using a supertagger and performs at a level comparable to the best supertagging results for English.

Résumé. Cet article décrit le développement d'une grammaire catégorielle à large couverture du Français, extraite à partir du corpus arboré de Paris 7 et vérifiée et corrigée manuellement. Le grammaire catégorielle résultant est évaluée en utilisant un supertagger et obtient des résultats comparables aux meilleurs supertaggers pour l'Anglais.

Mots-clés : Extraction de grammaires, grammaires catégorielles, supertagging.

Keywords: Categorial grammar, grammar extraction, supertagging, type-logical grammar.

1 Introduction

Though the development of parsers for the French language is an active area of research — as witnessed, for example, by the participation in the EASy evaluation — currently available grammar and parsers for the French language produce structures (typically shared forests or dependency structures) which are not easily exploitable for semantic tasks. Recently, the first results for wide-coverage semantic analysis for English have begun to emerge (Bos *et al.*, 2004); these developments have been made possible to a large extent by the availability of an automatically extracted grammar which permits an easy mapping of syntactic structures to semantic structures: the CCGbank (Hockenmaier & Steedman, 2007), a treebank for English with annotations in combinatory categorial grammar (CCG).

This paper describes the development of a type-logical treebank for French, which has been developed with is usefulness for such semantic tasks in mind. Using the Paris 7 treebank as a starting point, type-logical formulas are extracted automatically and then verified and corrected manually. The resulting grammar, which is still highly ambiguous because of the large number of lexical categories assigned to each word, is then evaluated using a supertagger, which disambiguates the lexical categories using local information only, and found to perform at a level comparable to the best results for English.

*This research has been partially financed by the conseil régional d'Aquitaine in the context of the ITIPY project.

2 Type-logical Grammar

This introduction to modern type-logical grammars will be necessarily brief and gives only an informal sketch of many of the ideas without going into the formal details. However, this short introduction will be sufficient to help the reader understand the grammar extraction in the next section. The interested reader can find a more detailed introduction and motivations in (Moortgat, 1997).

Type-logical grammars are a grammatical formalism with its roots in formal logic and the theory of types. A type-logical grammar defines a finite set of atomic formulas (typically s for *sentence*, np for *noun phrase*, n for *noun* and pp for *prepositional phrase*) and complex formulas $A \,/\, B$ (which looks to its right for an expression of type B in order to produce an expression of A) and $B \setminus A$ (which looks to its left for an expression of type B in order to produce an expression of type A).

Therefore $(np \setminus s) \,/\, np$ is a formula of the calculus. It would be the formula assigned to a transitive verb. It states that it combines first with a noun phrase to its right to produce an expression of type $np \setminus s$ (the type assigned to an intransitive verb) after which it combines with a noun phrase to its left to form a sentence.

In order to give an account of long-distance dependencies, such as those introduced by relativizers like *que* in French, we assign a formula of the form $(n \setminus n) \,/\, (s \,/\, \Diamond\Box np)$. Here, abstracting over the logical details, this formula indicates that *que* is looking to its right for a sentence s missing a noun phrase *somewhere*, as indicated by the subformula $s \,/\, \Diamond\Box np$, after which is will function as a noun modifier, selecting an n to its left to form an n.

In addition, the *multimodal* type-logical grammar used here (Moortgat, 1997) permits a controlled access to "movement" operations, and allows the assignment of a formula $s\backslash_1 s$ indicating that the adverb is a sentence modifier with mode information 1 allowing it to 'move' towards the head of the phrase.

In order to obtain the semantics for a syntactic analysis in a type-logical grammars we can directly apply the fact that the set of derivations in a type-logical grammar is a proper subset of the set of derivations in intuitionistic logic and thereby obtain the semantics of a derivation simply by means of the well-established Curry-Howard isomorphism between proofs and λ-terms. This means that from a categorial derivation — and an approriated lexicon assigning λ-terms to word-formulas pairs — we can directly obtain a semantic representation in the style of Montague or in a more modern dynamic framework such as DRT (Bos *et al.*, 2004).

3 Grammar Extraction

The Paris 7 treebank (Abeillé *et al.*, 2003) is a corpus containing extracts of the 'Le Monde' newspaper from December 1989 to January 1994. Part of this corpus (12,440 sentences containing a total of 371,029 word tokens, with 25,280 word types) has been given a functional annotation as well. Given that this functional annotation helps the extraction process and reduces the number of manual corrections, the grammar extraction has been defined for this sub-corpus only.

Figure 1 shows a tree from the Paris 7 corpus, a segment of the longer phrase *"La cour a, d'autre part, atténué le montant des amendes que la 11 chambre avait infligées aux autres prévenus"*. This sentence segment suffices to discuss the most interesting cases of the extraction algorithm.

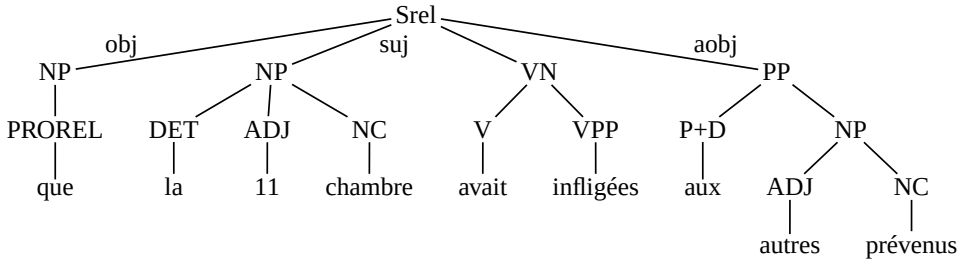

Figure 1: Example tree from the Paris 7 treebank with the extracted formulas indicated below the leaves

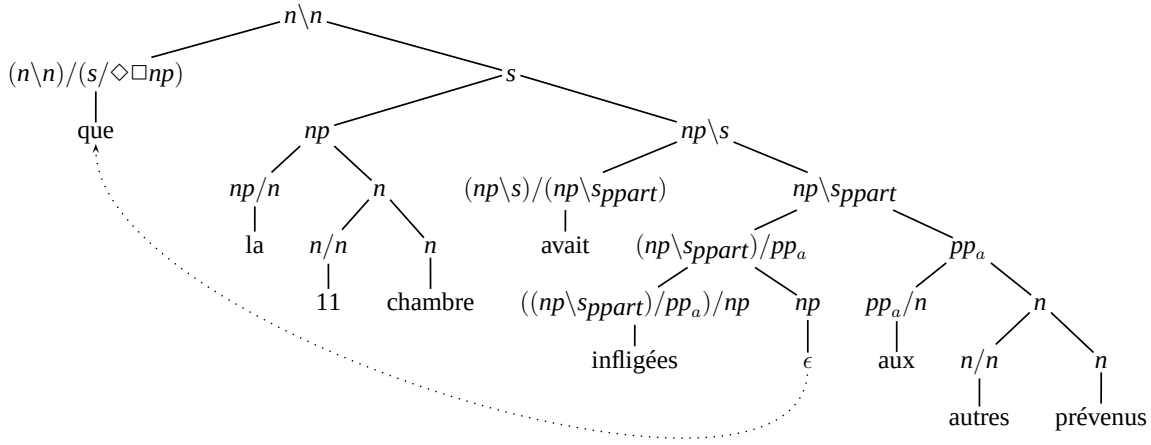

Figure 2: The tree from Figure 1 binarized and with formula information added

The grammar extraction algorithm used follows the general principles of grammar extraction for categorial grammars, as it has been used in other contexts (Buszkowski & Penn, 1990; Hockenmaier & Steedman, 2007; Moot, 2010).

1. the tree is binarized, when necessary nodes are inserted for 'traces' (we will see this in more detail in the example below).
2. in the resulting tree, a distinction is made between heads (functors), arguments and modifiers; a simple table lookup is used to decide between the different cases based on the node labels and their functional annotation, so in a sentence (annotated by SENT) the verb cluster (annotated by VN) is assigned as its head, NP and PP daughters are assigned as arguments[1], whereas adverbs (annotated by ADV and ADVP) would be assigned a modifier role. Similarly, in an NP, adjectives and prepositions will be modifiers.
3. formulas are assigned in a top-down fashion starting with the root node and descending the tree by case analysis; again a table lookup is used to convert syntactic categories to atomic formulas

    - if the parent node has formula A assigned to it and the left (resp. right) daughter is a modifier then the left daughter will receive formula A/A and the right daughter formula A (resp. A and $A\backslash A$ in the case the right daughter was a modifier
    - if the left daughter is an argument and the right daughter is a functor, we look up the corresponding formula B in the table and assign B and $B\backslash A$ to the two daughters; similarly if

[1]unless they have the NP_{mod} functional annotation, as would expressions like "dimanche 5 janvier" or "cette fois" or the PP_{mod} functional annotation, as would expressions like "en Espagne" or "sauf accident"

the left daughter is a functor and the right daughter an argument with formula B then the corresponding formulas will be A/B and B respectively.

- for a leaf with word w and computed formula A, the pair $w - A$ is added to the lexicon.

Figure 2 shows the result of the extraction algorithm for the tree discussed above. I will just comment on some of the less evident aspects of the extracted lexicon.

Firstly, some restructuring has taken place. The prepositional phrase headed by *aux* was assigned in the corpus as an argument of the verb cluster *VN*, where in the extracted lexicon it is an argument of the past participle *infligées*. Again, these cases are very common and this reattachment gives us a more natural lexicon. Otherwise, the binarization is without surprises.

A second important element is the analysis of the relative pronoun *que*. As can be seen in Figure 1, the corpus indicates the *que* is has the role of object in the relative phrase, but in addition it functions as a modifier of the noun *amendes* (absent from the figure). The extraction algorithm adds an *np* 'trace', indicated by ϵ, to the rightmost verb of the first verbal group occurring after the relativizer, though manual verification is often necessary to verify this is the correct position.[2] The relation between the trace and the relativizer is indicated in the figure by a dotted line. Readers familiar with multimodal categorial grammars will recognize the dotted line corresponds to the introduction rule for the implication [/*I*]. All in all, this gives us the category for the relative pronoun *que* which we discussed in Section 2: a noun modifier selecting a sentence missing a noun phrase to its right.

Finally, it should be noted that some of the syntactic categories have subcategories: we distinguish between s when it occurs as a past participle (s_{ppart}) or infinitive group (s_{inf}) making it possible for a verb to admit only specific verb groups as its arguments. This allows us to assign $(np\backslash s)/(np\backslash s_{ppart})$ to the different forms of *avoir* and $(np\backslash s)/(np\backslash s_{inf})$ to the different forms of *vouloir*.

4 Improving the extracted grammar

A first run of the extraction algorithm gives a highly ambiguous grammar with 5.240 distinct formulas which have been assigned at least once to one of the words in the lexicon. Manual inspection of this first treebank TLG_0 reveals that the extracted lexicon contains many formulas which are the result either of inconsistencies in the treebank or of inconsistencies between the way a phenomenon is analyzed in the treebank and the way it would preferably be analyzed in a type-logical grammar.

A first improvement is the reduction of the different formulas assigned to adverbs. One of the prototypical positions for an adverb is directly to the right of the verb it modifies. This means that if we assign the formula A to this verb, the adverb will have formula $A\backslash A$ assigned to it. However, using the multimodal solution sketched in Section 2 permits us to reduce these instances to the formula $s \setminus_1 s$.

In addition, by inspecting the different lexical entries, both for the most frequent words and for the different part-of-speech tags, entries which have been deemed suspect have been manually verified and, where necessary, corrected. Inversely, the list of words assigned to each of the different formulas has been inspected and again formulas which looked inappropriate for the words to which they were assigned have been verified, corrected and made more consistent. To give an indication of the impact of these

[2] Some relatative pronouns like *qui* are easy, whereas others, like *dont* and *laquelle* require more effort.

simplifications, the different formulas for conjunctions have been reduced from 606 in TLG_0 to 188 in TLG.

Taken together, these improvements reduce the number of lexical formulas to a more reasonable, but still quite large, number of 817 distinct formulas. This second treebank, TLG, while undoubtedly still containing a fair amount of errors, provides a good balance between lexicon size and descriptive adequacy.

Some differences with the English treebank CCGbank should be noted. First of all, in the CCGbank, conjunctions (in our case *et*, *ou* and on some occasions the comma) are handled by the parser whereas in the TLG treebank this information is handled by the supertagger, hence the 188 different categories cited above. In addition, to reduce the number of lexical categories, the CCGbank uses a number of non-logical axioms which transform past participles to adjectives (useful in noun phrases like "le risque lié au négoce international", where *lié* is assigned $(n\backslash n)/pp_a$ — instead of its usual $(np\backslash s_{ppart})/pp_a$ — indicating that in this context, it selects a prepositional phrase to its right in order to become a noun modifier). Other non-logical axiom include a rule allowing an n to function as an np, which is used for a noun phrases without a determiner. In order to give an indication of the effects of these simplifications in the current context, a second grammar by automatically simplifying the first grammar in accordance with these strategies. We will refer to this more compact grammar as TLG_c.

5 Evaluation

β	TLG	#/w	TLG_c	#/w
1.0	90.5	1.0	93.5	1.0
0.1	96.4	2.7	97.5	2.5
0.05	97.3	3.1	98.0	2.9
0.01	98.4	4.7	98.8	3.8

Table 1: Supertagger results for the TLG treebanks

In spite of all the reductions made to the treebanks, the resulting lexicon still has a very high number of formulas assigned to each word. In order for the extracted grammar to be more easily parsed, a 'classic' strategy is to use a supertagger which decides, based on the surrounding local context — the words and part-of-speech tags occurring in a two-word window around the current word as well as the previous two formulas or 'supertags' — which is the most likely formula to assign to the current word.

The maximum entropy supertagger developed by Clark & Curran (2004) has been used to evaluate supertagger performance on the TLG treebank. The treebank has been split into two set: a of training data containing 11.196 sentences and 334.525 words and a set of test data, containing 1.244 sentences and 36.504 words. The maximum entropy model has been trained with the Clark & Curran (2004) supertagger using their adaptation of the L-BFGS algorithm to optimize parameter estimation.

Results for the extracted grammars TLG and TLG_c are shown in Table 1.[3] For the first row only the best supertag has been kept, whereas the other rows list the result for a multitagger which keeps all supertag with probability greater than β times the highest probability (Clark, 2002), with a lower β value meaning a larger set of supertags assigned to each word. The left-hand column lists the percentage of the sets of supertags assigned to word-POS tag pairs containing the correct supertag for experiments TLG and TLG_c, with the right-hand column listing the average number of supertags per word. Though we should

[3]Results for the treebank TLG_0 which incorporates none of the improvements described in Section 4 are not shown in the table. Supertagging predicision for this treebank is 79.4 %, which is in line with results reported for automatically extracted TAGs (Chen & Vijay-Sjanker, 2000) both in terms of the number of different supertags and precision.

be careful making direct comparisons between results from different languages and different formalisms, these results indicate a supertagger performance comparable with the supertaggers for English described in (Clark, 2002), though our results have a slightly higher number of supertags assigned to each word for the lower β values.

6 Conclusion and Future Work

In this paper a wide-coverage type-logical gramar for French has been semi-automatically extracted from the Paris 7 treebank and the resulting corpus has been evaluated using a supertagger. The supertagger obtains state-of-the-art performance compared to English supertaggers.

Currently, different parsing strategies are being developped for further evaluation of the results of the supertagger, and early results look promising. Development of a wide-coverage semantic lexicon for this grammar — in the style of Bos *et al.* (2004) — is progressing rapidly and both this lexicon and the trained models for the POS and supertagger will be made available to the research community under the LGPL-LR license.

ABEILLÉ A., CLÉMENT L. & TOUSSENEL F. (2003). Building a treebank for French. In A. ABEILLÉ, Ed., *Treebanks: Building and Using Parsed Corpora*, chapter 10, p. 165–187. Dordrecht: Kluwer.

BOS J., CLARK S., STEEDMAN M., CURRAN J. R. & HOCKENMAIER J. (2004). Wide-coverage semantic representation from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, p. 1240–1246, Geneva, Switzerland.

BUSZKOWSKI W. & PENN G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica*, **49**, 431–454.

CHEN J. & VIJAY-SJANKER K. (2000). Automated extraction of TAGs from the Penn treebank. In *Proceedings of the 6th International Workshop on Parsing Technologies*, Trento, Italy.

CLARK S. (2002). Supertagging for combinatory categorial grammar. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Formalisms*, p. 19–24, Venice.

CLARK S. & CURRAN J. R. (2004). Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL-2004)*, p. 104–111, Barcelona, Spain.

HOCKENMAIER J. & STEEDMAN M. (2007). CCGbank, a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, **33**(3), 355–396.

MOORTGAT M. (1997). Categorial type logics. In J. VAN BENTHEM & A. TER MEULEN, Eds., *Handbook of Logic and Language*, chapter 2, p. 93–177. Elsevier/MIT Press.

MOOT R. (2010). Automated extraction of type-logical supertags from the spoken dutch corpus. In S. BANGALORE & A. JOSHI, Eds., *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*. MIT Press.