A Type-Logical Treebank for French

Richard Moot

LaBRI (CNRS), Bordeaux University
Richard.Moot@labri.fr

1 Introduction

Categorial grammars have interesting theoretical advantages, most notably their very clean syntax-semantics interface. In the last decade, research in Combinatory Categorial Grammar has shown that this is not merely a *theoretical* advantage, but that, with the appropriate resources and tools — an annotated treebank, the CCGbank [13], a very efficient parser [10] and a semantic lexicon [4]) — we can use categorial grammars for wide-coverage, deep semantic analysis. Applications of the resulting wide-coverage semantics include natural-language question-answering [5] and computing textual entailments [6].

A key element has been the development of the CCGbank, which has allowed both parameter-optimization for the wide-coverage parser and provided a framework (in types and in derivations) for the semantic applications.

Categorial grammars in the logical tradition [15, 18, 17] have stayed somewhat behind in terms of their application to large-scale linguistic data. The goal of the current paper is to describe the TLGbank, a semi-automatically extracted treebank containing type-logical proofs, created with the explicit goal of making similar wide-coverage parsing and semantics possible in the type-logical context.

2 The French Treebank

The French Treebank (FTB, [1]) is a set of syntactically annotated news articles from the newspaper *Le Monde*. The FTB consists of 12,891 annotated sentences with a total of 383,227 words. The FTB has previously been used to extract lexical-functional grammars [20] and tree adjoining grammars [11].

For the annotation, the FTB uses simple, rather flat trees with some functional syntactic annotation (subject, object, infinitival argument, etc.). Consecutive multiword-expression have been merged in the annotation and neither traces nor discontinuous dependencies have been annotated. Figure 1 shows a fragment of a sentence from the FTB. Verb clusters are treated as a constituents (labeled *VN*) and the arguments of the verb occur as sisters of the verbal cluster (eg. the infinitival argument with functional role *OBJ* in Figure 1).

3 Type-Logical Grammars

This section is a very short introduction to (multimodal) type-logical grammars. More detailed introductions can be found in Section 2.4 of [15] and in Chapter 5 of [17].

The atomic formulas are n (for nouns), np (for noun phrases), pp_x (for prepositional phrases, with x the preposition heading the phrase) and s_x for sentences (distinguishing between several types s_{main} for main, tensed sentence, s_{whq} for a wh-question, s_q for a sentence introduced by (*that*) and further types for passives s_{pass}, infinitives s_{inf}, and past s_{ppart} and present s_{ppres} participles; this is inspired by the FTB annotation, though passives are not annotated as such, and the categorial treatments of [9, 13] implemented using first-order logic [16]).

An intransitive verb is assigned $np\backslash s_{main}$, indicating that it requires a noun phrase to its left in order to form an inflected sentence. Similarly, transitive verbs are assigned the formula $(np\backslash s_{main})/np$, requiring a noun phrase to their right in order to form an intransitive verb.

Table 1 lists (a slightly simplified version) of the most common rules used in the extracted treebank.Section 3.1 sketches some linguistic phenomena requiring additional rules and gives some references as to where to find these rules.

$$\frac{}{w \vdash A}\ Lex \qquad \frac{}{x \vdash A}\ Hyp$$

$$\frac{X \vdash A/B \quad Y \vdash B}{X \circ Y \vdash A}\ /E \qquad \frac{X \vdash B \quad Y \vdash B\backslash A}{X \circ Y \vdash A}\ \backslash E$$

$$\frac{\begin{array}{c} x \vdash B \\ \vdots \\ X \circ x \vdash A \end{array}}{X \vdash A/B}\ /I \qquad \frac{\begin{array}{c} x \vdash B \\ \vdots \\ x \circ X \vdash A \end{array}}{X \vdash B\backslash A}\ \backslash I$$

$$\frac{X[Y] \vdash B \quad Z \vdash B\backslash_1 A}{X[Y \circ Z] \vdash A}\ \backslash_1 E \qquad \frac{\begin{array}{c} x \vdash B \\ \vdots \\ X[Y \circ x] \vdash A \end{array}}{X[Y] \vdash A/\Diamond_1\Box_1 B}\ /\Diamond_1\Box_1 I$$

Table 1. Logical rules for multimodal categorial grammars

We will abbreviate the lexicon rule as $\frac{w}{A}$. The rule for $/E$ simply states that whenever we have shown an expression X to be of type A/B and we have shown an expression Y to be of type B, then the tree with X as its immediate subtree on the left and Y as its immediate subtree of the right is of type A.

An easy instantiation of this rule (with $X :=$ *the*, $Y :=$ *student*, $A := np$, $B := n$) would be the following (the $\backslash E$ rule is symmetric).

$$\frac{\mathit{the} \vdash np/n \quad \mathit{student} \vdash n}{\mathit{the} \circ \mathit{student} \vdash np}\ /E$$

The two rules on the bottom of the figure require some special attention. The $\backslash_1 E$ rule is an *infixation rule*. This rule is used for adverbs (and other VP modifiers) occurring after the verb. Like the $\backslash E$ rule, it takes a B formula as its argument, but infixes itself to the right of any subtree Y of X ($X[Y]$ denotes a tree X with a designated subtree Y[1]). An example is shown below for the VP "*impoverishes the CGT dangerously*". The interest of this rule is that it allows a uniform type assignment for adverbs occurring post-verbally, regardless of other verb arguments.

$$\frac{\dfrac{\mathit{appauvrit} \vdash (np\backslash s)/np \quad \mathit{la} \circ \mathit{CGT} \vdash np}{\mathit{appauvrit} \circ \circ(\mathit{la} \circ \mathit{CGT}) \vdash np\backslash s}\ /E \qquad \mathit{dangereusement} \vdash (np\backslash s)\backslash_1(np\backslash s)}{(\mathit{appauvrit} \circ \mathit{dangereusement}) \circ (\mathit{la} \circ \mathit{CGT})}\ /E$$

Finally, the $/\diamond_1\square_1$ rule is an *extraction* rule, extracting a B constituent from any right branch inside an X constituent. Section 4.3 shows an example.[2]

3.1 Additional Linguistic Phenomena

The rules listed in Table 1 correspond to the most frequently used rules for the type-logical treebank. The additional rules are a) for the product (primarily used for coordination of multiple arguments (as shown in sentence (1) below, where the two verb arguments *np* and *pp* are conjoined, see Section 2.4 of [18]), b) for gapping (as shown in sentence (2) below, where the transitive verb "atteindre" is absent from the second clause; a multimodal solution is proposed in [12]), and c) for some special rules to treat past-perfect quoted speech, as shown in sentence (3) below. The parenthesized sentence is argument of the past participle "ajouté" and, in addition, this argument is discontinuous. The solution is essentially to analyse the entire verb group missing the s argument "a ajouté ... travailliste" as $s_{main}\backslash_1 s_{main}$.

(1) ... augmenter [np ses fonds propres] [pp de 90 millions de francs] et
... increase [np its equity] [pp by 90 million francs] and
[np les quasi-fonds propres] [pp de 30 millions] ...
[np its quasi-equity] [pp by 30 million] ...

(2) Le salaire horaire atteint dorénavant 34,06 francs et le
The wages per hour reach from now on 34,06 francs and the
SMIC mensual brut [tv] 5756,14 francs.
gross minimum monthly wage [tv] 5756,14 francs.

[1] For adverbs, as here, Y is typically the verb, but in principle infixation is possible anywhere (an admitted simplification)

[2] For readers familiar with the displacement calculus [19], the infixation construction $A\backslash_1 B$ corresponds to $\check{}B \downarrow A$ and the extraction construction $A/\diamond_1\square_1 B$ to $\hat{}(A \uparrow B)$

(3) [sl Les conservateurs], a ajouté le premier ministre ..., [sr "ne sont
[sl The Conservatives], has added the Prime Minister ..., [sr " are
pas des opportunistes qui virevoltent d'une politique à l'autre]
not opportunists who flip-flop from one policy to another]

4 Grammar Extraction

Grammar extraction algorithms for categorial grammars follow a general methodology (see, for example, [7, 13], shown as item 2 below) with some additional rules to deal with the quirks of the format of the input treebank. A high-level description of the grammar extraction algorithm used for the FTB is given below.

1. split multiword expressions,
2. binarize the tree, keeping track of the distinction between modifiers and arguments, arguments are assigned formulas based on their syntactic label (eg. np for a noun phrase argument, $np\backslash s_{inf}$ for an infinitival argument, etc.)
3. reattach verb cluster arguments,
4. rearrange coordinations,
5. insert traces in the appropriate places and assign the appropriate formulas to relative pronouns and clitics

Unfortunately, nearly all of these steps require at least some human intervention: the FTB annotation makes the distinction between modifiers and arguments only for certain categories (sentences, infinitive phrases, present participle phrases, but not past participle phrases or noun phrases), meaning that for many major categories this information is not explicitly annotated and needs to be verified manually.

4.1 Verb Clusters

As discussed in Section 2, verb clusters (which include clitics and adverbs) and their arguments are sisters in the FTB annotation trees. Figure 1 shows an example corresponding to sentence (4).

(4) Ils ont déjà pu constater que (...)
They have already been able to note that (...)

In a categorial setting, we obtain a much simpler analysis if these VN arguments are arguments of the embedded verbs instead (in the current case, we'd like the infinitival group to be the argument of the past participle "pu" (of the verb "pouvoir", *can*). At the bottom of Figure 1 we see the rightward branching structure which results from the corpus transformation. Note also how the adverb "déjà" (*already*) is assigned the VP-modifier formula $(np\backslash s_x)/(np\backslash s_x)$ which is parametric for the type of sentence (in essence, this is a formula with an implicit first-order quantifier ranging over the different sentence types, see Section 2.7 of [15]; in the figure, x is instantiated to *ppart*).

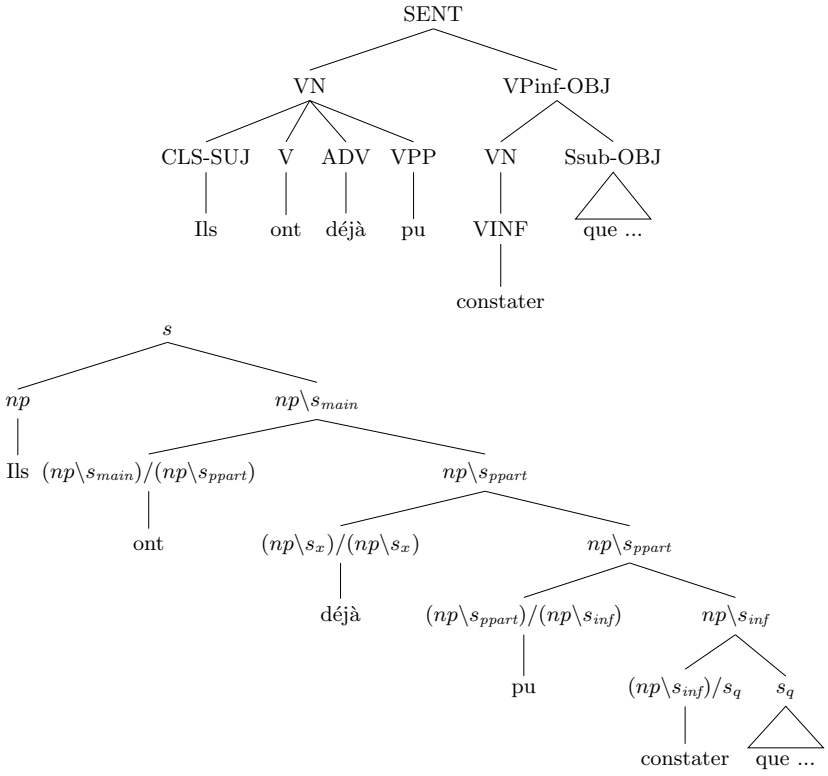

Fig. 1. Rebracketing a verbal group and its arguments

4.2 Coordination and Interpunction Symbols

The sentences below illustrate some of the problems with coordinations which we will discuss in this section.

(5) Elles reprennent et amplifient des programmes existants ou
They resume and amplify programs existing or
en cours d' adaptation
currently being adapted

(6) Les lieux où les deux derniers morts ont été recensés,
The places where the two last deaths have been reported,
lundi 30 décembre, La Yougoslavie et La Colombie, (...)
Monday 30 December, Yugoslavia and Colombia,

Figure 2 shows the FTB syntactic structure of sentence (5). In categorial grammars, conjunctions like "ou" (*or*) are generally assigned instances of the formula $(X\backslash X)/X$ (for a contextually appropriate choice of the formula X). The first

conjunction is of the two transitive verbs (instantiating X with the formula $(np\backslash s_{main})/np$) who share both the subject and the object. For the second coordination it is the adjective and the prepositional phrase which are conjoined (though this is not so clear from the annotation only, where it seems an unlike coordination between an np and a pp). As is standard in categorial grammars, we assign both the adjective and the PP the formula $n\backslash n$ (this is the standard assignment for a PP modifying a noun), turning this seemingly unlike coordination into a trivial instance of the general coordination scheme.

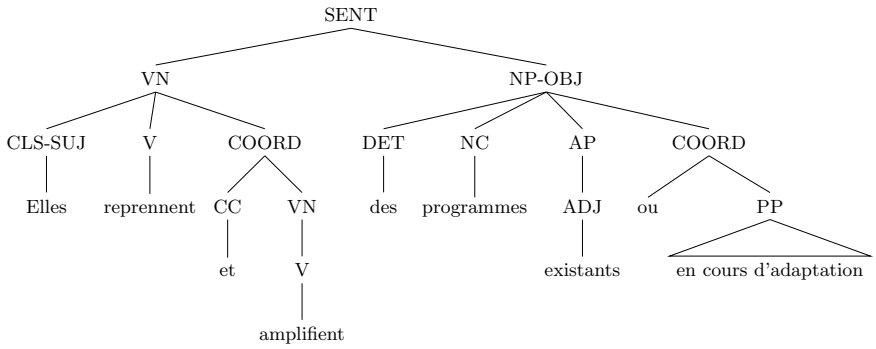

Fig. 2. Coordination

The (somewhat simplified) FTB annotation of sentence (6) of Figure 3 on the next page, shows another problem: appositives, which are treated by assigning a coordination-like formula to the interpunction symbol preceding them (a similar solution is used for parentheticals and for most extrapositions[3]) Additionally, we have to distinguish between the NP-MOD temporal adverb (which modifies the verb "recensés" and the NP-MOD for the appositive (which conjoins to "Les lieux", *the places*)

As the example shows, these cases are difficult to infer from the information provided by the FTB annotation alone, and therefore must be verified manually; in total a bit over 20% of the interpunction symbols — over ten thousand interpunction symbols — are assigned coordination-like categories.

[3] Not all extrapositions can be analysed as coordinations this way. In the example below

(i) A celà s'ajoute une considération générale : (...)
To that adds-itself a general consideration

"A cela" is assigned $s/(s/\Diamond_1\Box_1 pp_a)$ allowing it to function as a long-distance pp argument to "s'ajoute".

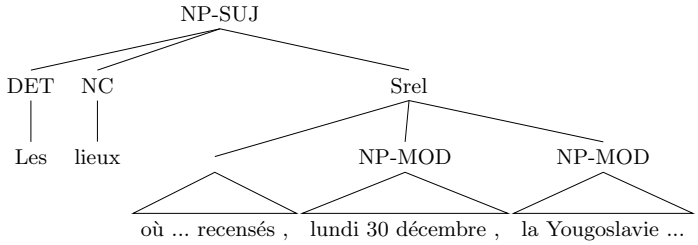

Fig. 3. Appositives

4.3 Traces and Long-Distance Dependencies

As an example of a simple long-distance dependency in the corpus, consider the example below.

(7) Premier handicap auquel il convenait de s'attaquer : l'inflation
First handicap to which it was agreed to attack : the inflation

Figure 4 on the next page shows how the insertion of traces works. In the input structure on the top of the figure, "auquel" (*to which*) is assigned a preposition+pronoun POS-tag and assigned the role of a prepositional object with the preposition "à" (*to*). However, this preposition is an argument of the verb "s'attaquer à" (*to attack*), which occurs much lower in the annotation tree. Since none of these dependencies are annotated in the French Treebank, all relative pronouns, wh-pronouns and clitics — a total of over 3,000 occurrences in the corpus — have been manually annotated with the correct long-distance dependencies. At the bottom of Figure 4, the manually added long-distance dependency is shown.

4.4 Analysis

Categorial grammars, much like lexicalized tree adjoining grammars and other strongly lexicalized formalisms, use very construction-specific lexical entries. This means, for example, that when a verb can be used both as a transitive verb and as an intransitive verb, it will have (at least) two distinct lexical entries. For extracted grammars, this generally means a very high level of lexical ambiguity.

Using the most detailed extraction parameters, the final lexicon uses 1101 distinct formulas (though only 800 of these occur more than once and, 684 more than twice and 570 at least five times).

Using a slightly less detailed extraction (which, for example, distinguishes only pp_{de}, pp_{a} and pp_{par} and uses simply pp for prepositional phrases headed by other prepositions) there are 761 different formulas used in the lexicon (of which only 684 occur more than once, 546 occur more than twice and 471 occur at least five times)

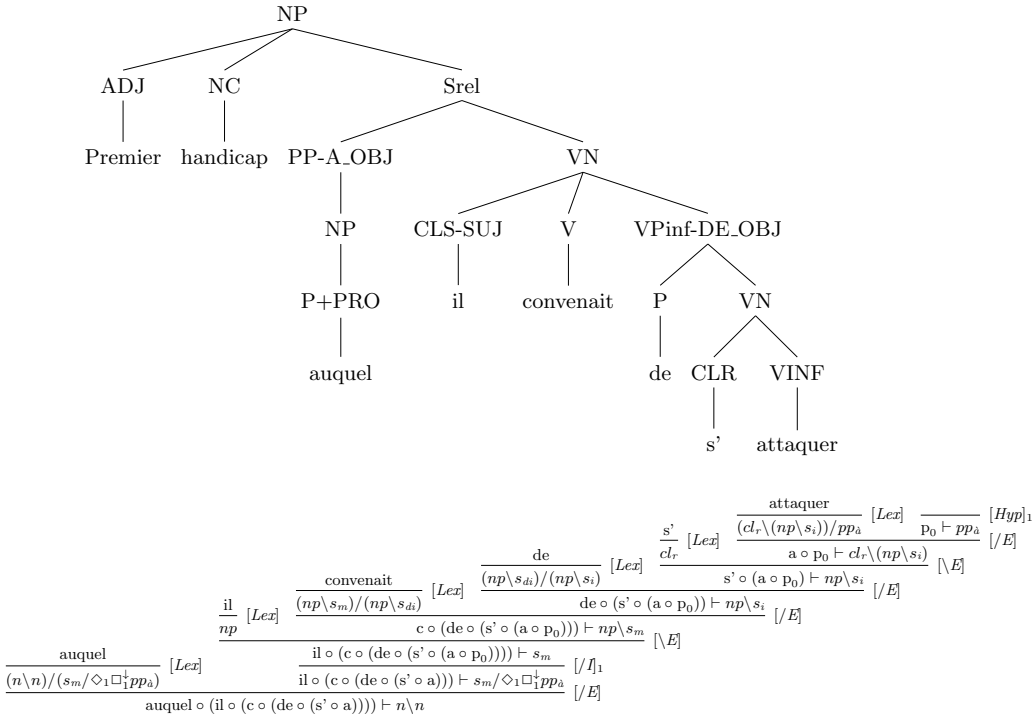

Fig. 4. Adding traces to the output

Even in this second lexicon, many frequent words have a great number of lexical assignments. The conjunction "et" (*and*) has 86 different lexical formulas, the comma "," (which, as we have seen, often functions much like a conjunction) is assigned 72 distinct formulas, the adverb "plus" (*more*) 44 formulas (in part because of possible combinations with "que", *than*), the prepositions "pour", "en" and "de" 43, 42 and 40 formulas respectively, and the verb "est" (*is*) 39 formulas.

Though this kind of lexical ambiguity may seem like a problem when using the lexicon for parsing, well-known techniques such as *supertagging* [2], which assign the contextually most probable set of formulas (supertags) to each word, can be used to reduce the lexical ambiguity to an acceptable level. To give an idea as to how effective this strategy is in the current context and with the reduced lexicon of 761 formulas, when assigning only the most likely formula to each word, 90.6% of the words are assigned the correct formula, when assigning each word all formulas with probability greater than 1% of the most likely supertag (for an average of 2.3 formulas per word), the supertagger assigns 98.4% (complete treebank, using ten-fold cross-validation).

4.5 Comparison With the CCGbank

Apart from the obvious theoretical differences between CCG and type-logical grammars and the different treatment of certain linguistic phenomena — such as extraction — that this implies, it is worth spending some time on some of the less obvious differences between the two treebanks.

Whereas the CCGbank uses a certain number of non-combinatory rules (notably for extraposition and coordination, but also to transform passives $np\backslash s_{pass}$ into adjectives $n\backslash n$ and (bare) nouns n into noun phrases np, the current treebank uses no non-logical rules. As a result, the lexicon of the type-logical treebank does more of the work (and consequently, the taks of the supertagger is more difficult).

If we want to reduce the size of the lexicon in a way similar to the CCGbank, there are two basic options:

- the first option is to allow non-logical rules in the same spirit as the CCGbank,
- the second option, more in line with the general spirit of type-logical grammars, is to exploit the derivability relation and to replace the analysis of passives by a formula F such that $F \vdash n\backslash n$ (see Section 4.4.2 of [18] for a particularly nice solution).

However, we leave the transformation of the proofs in the corpus in these two ways to future research.

5 Tools

To facilite annotation, correction and parsing, several tools have been developed, using a combination of Prolog and TclTk. In addition, several well-known tools have been used for the exploitation of the corpus: the Stanford Tregex tool [14] for browsing and querying the French Treebank (as well as some of its transformations) and the C&C tools [10] for training POS-tag and supertag models using the annotated corpus.

Figure 5 on the next page shows a screenshot of the interface to the supertagger and parser. This "horizontal" interface allows the user to type in sentences and see the resulting semantic output from the parser. The darker-shader percentage of the block to the left of the formula gives a visual indication of the probability assign to the formula (the exact numbers can be seen by moving the mouse over the corresponding area). Apart from some configuration options, this interface is not interactive.

Figure 6 shows a screenshot of the "vertical" interface to the parser and supertagger. This is an interactive interface, allowing the user to select (or type in) the desired formula — to help prevent errors, the current frequency of the chosen formula for the current word is displayed after a manual choice of formula— as well as allowing the user the select the parser rule applications by clicking on one of the premisses for a rule (an additional dialog pops up in case the rule choice is ambiguous). The weight column shows the log-probability of the item.

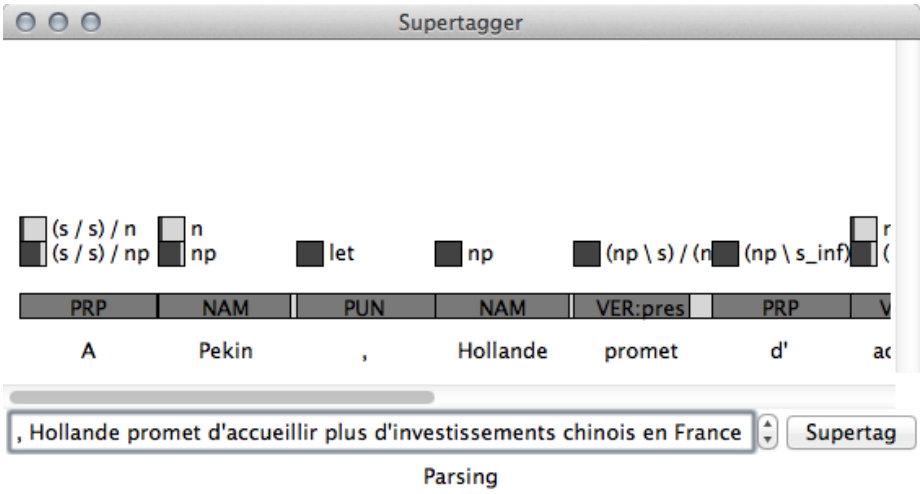

Fig. 5. Screenshot of the supertagger interface

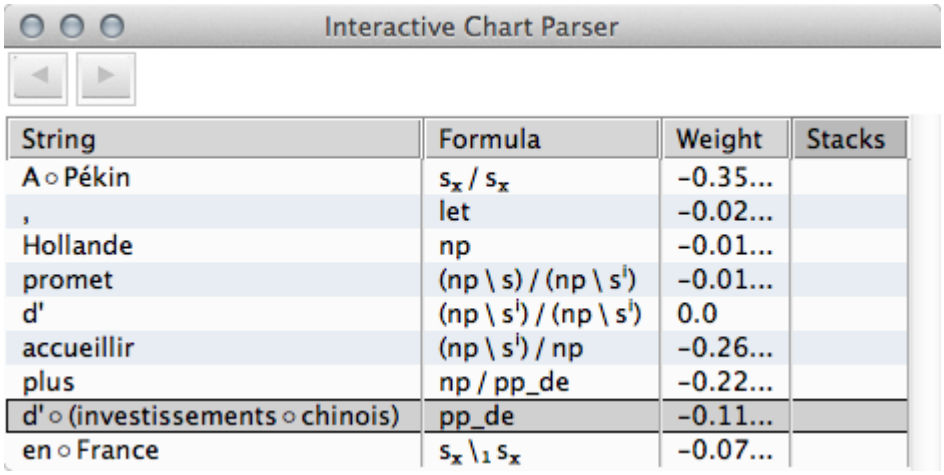

Fig. 6. Screenshot of the interactive parser

6 Bootstrapping

Given that the French Treebank is somewhat small compared to other treebanks and given that the conversion of the FTB to the type-logical treebank was rather labour-intensive, it makes sense to look at more effective and efficient ways of increasing the size of the treebank. The tools described in the previous section,

interfacing with the supertagger and the parser for the core corpus are useful in this respect.

Currently, slightly over 1,600 additional sentences have been annotated (for a total annotated corpus of 14,539 sentences and 421,348 words). Most of these sentences come from the Sequoia treebank [8] and the French Timebank [3]. The observed accuracy of the supertagger for these sentences from the *L'Est Républicain* newspaper is slightly lower than the results reported in Section 4.4: in 88.1% of cases, the best supertag is correct, and 97.6% of cases the correct supertag has probability greater than 1% of the best supertag (compared to 90.6 and 98.4% respectively for the cross-validated results). Part of this difference might be attributed to stylistic differences between the two newspapers (initial experiments with annotating unseen sentences from *Le Monde* seem to confirm this) but it may also be the case that cross-validation gives a somewhat optimistic picture of actual performance on unseen data from other sources (the different training and test sets not being completely independent).

7 Obtaining the Tools and Resources

All tools, as well as the POS-tagger and supertagger models and a semantic lexicon in the style of [4], are available from the author's website under the LGPL licence. The TLGbank, being a derived work, is available under the same licensing conditions as the French Treebank. The Sequoia/L'Est Républicain part of the treebank is available under the LGPL-LR licence.

8 Conclusions

We have shown how the French Treebank has been semi-automatically transformed into a set of derivations in multimodal type-logical grammars. This is an important first step in training an evaluating wide-coverage type-logical parsers and we hope to see several competitive type-logical parsers in the future.

References

1. Abeillé, A., Clément, L., Kinyon, A.: Building a treebank for French. In: Proceedings of the Second International Language Resources and Evaluation Conference. Athens, Greece (2000)
2. Bangalore, S., Joshi, A.: Supertagging: Using Complex Lexical Descriptions in Natural Language Processing. MIT Press (2011)
3. Bittar, A.: Building a TimeBank for French: A Reference Corpus Annotated According to the ISO-TimeML Standard. Ph.D. thesis, Université Paris Diderot (2010)
4. Bos, J., Clark, S., Steedman, M., Curran, J.R., Hockenmaier, J.: Wide-coverage semantic representation from a CCG parser. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004). pp. 1240–1246. Geneva, Switzerland (2004)

5. Bos, J., Curran, J.R., Guzzetti, E.: The Pronto QA system at TREC-2007: harvesting hyponyms, using nominalisation patterns, and computing answer cardinality. In: Voorhees, E.M., Buckland, L.P. (eds.) The Sixteenth Text RETrieval Conference, TREC 2007. pp. 726–732. Gaitersburg, MD (2007)
6. Bos, J., Markert, K.: Recognising textual entailment with logical inference. In: Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005). pp. 628–635 (2005)
7. Buszkowski, W., Penn, G.: Categorial grammars determined from linguistic data by unification. Studia Logica 49, 431–454 (1990)
8. Candito, M., Seddah, D.: Le corpus Sequoia : Annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In: Proceedings of Traitement Automatique des Langues Naturelles (TALN). Grenoble (2012)
9. Carpenter, B.: Categorial grammars, lexical rules and the english predicative. In: Levine, R. (ed.) Formal Grammar: Theory and Practice. No. 2 in Vancouver Studies in Cognitive Science, University of British Columbia Press, Vancouver (1991)
10. Clark, S., Curran, J.R.: Parsing the WSJ using CCG and log-linear models. In: Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL-2004). pp. 104–111. Barcelona, Spain (2004)
11. Dybro-Johansen, A.: Extraction automatique de grammaires à partir d'un corpus français. Master's thesis, Université Paris 7 (2004)
12. Hendriks, P.: Ellipsis and multimodal categorial type logic. In: Morril, G., Oehrle, R.T. (eds.) Proceedings of Formal Grammar 1995. pp. 107–122. Barcelona, Spain (1995)
13. Hockenmaier, J., Steedman, M.: CCGbank, a coxbrpus of CCG derivations and dependency structures extracted from the Penn Treebank. Computational Linguistics 33(3), 355–396 (2007)
14. Levy, R., Andrew, G.: Tregex and tsurgeon: tools for querying and manipulating tree data structures. In: 5th International Conference on Language Resources and Evaluation (LREC 2006). (2006)
15. Moortgat, M.: Categorial type logics. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, chap. 2, pp. 95–179. North-Holland Elsevier, Amsterdam (2011)
16. Moot, R.: Extended lambek calculi and first-order linear logic. to appear in Springer Lecture Notes in Artificial Intelligence (2013)
17. Moot, R., Retoré, C.: The Logic of Categorial Grammars. Lecture Notes in Artificial Intelligence, Springer (2012)
18. Morrill, G.: Categorial Grammar: Logical Syntax, Semantics, and Processing. Oxford University Press (2011)
19. Morrill, G., Valentín, O., Fadda, M.: The displacement calculus. Journal of Logic, Language and Information 20(1), 1–48 (2011)
20. Schluter, N., van Genabith, J.: Treebank-based acquisition of LFG parsing resources for French. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, Morocco (2008)