A type–logical treebank for French

Richard Moot
CNRS (LaBRI), Bordeaux University

ABSTRACT

This paper describes the TLGbank, a treebank developed in the framework of (multimodal) type-logical grammar. Using the French Treebank as a starting point, a combination of automated and manual techniques are applied to obtain type-logical derivations (parses) corresponding to the phrases of the French Treebank. The TLGbank has been developed with applications to wide-coverage semantics in mind. This means that the TLGbank has richer structure than the original French Treebank, especially where it concerns semantically relevant information such as passives, coordination, extraction and gapping.

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 (Hockenmaier and Steedman 2007), a very efficient parser (Clark and Curran 2004) and a semantic lexicon (Bos *et al.* 2004) – we can use categorial grammars for wide-coverage, deep semantic analysis. Applications of the resulting wide-coverage semantics include natural-language question-answering (Bos *et al.* 2007) and recognising textual entailments (Bos and Markert 2005).

The development of the CCGbank, which has allowed parameter optimization for the wide-coverage parser and provided a framework

(in types and in derivations) for the semantic applications, has been a key element for these applications.

Categorial grammars in the logical tradition initiated by Lambek (1958) (Moortgat 2011; Morrill 2011; Moot and Retoré 2012) 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.

The work described in this paper extends and refines a much earlier version of the TLGbank (Moot 2010b). Lefeuvre *et al.* (2012) and Moot (2012) discuss some initial applications of the treebank to wide-coverage semantics.

2 TYPE–LOGICAL GRAMMAR

This section is a very short introduction to (multimodal) type-logical grammars. For more detailed introductions, see Oehrle (2011), Moortgat (2011, Section 2.4) or Moot and Retoré (2012, Chapter 5).

Although the treebank is annotated using multimodal type-logical grammar, the annotation has been chosen in such a way that derivations in the treebank can easily be translated into derivations of the Displacement calculus (Morrill *et al.* 2011) or of first-order linear logic (Moot and Piazza 2001; Moot 2014). Translations to other versions of categorial grammar are conceivable, but will probably require significantly more work.

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, where we distinguish between several types of sentences/phrases: s_{main} for main, tensed sentence, s_{whq} for a wh-question, s_q for a sentence introduced by "que" (*that*) and further types for passives s_{pass}, infinitives s_{inf},[1] and past s_{ppart} and present s_{ppres} participles; this is inspired by the French Treebank annotation –

---

[1] Like prepositions, s_{inf} is further subdivided into categories for infinitive phrases headed by a preposition: s_{inf_a}, $s_{inf_{de}}$, $s_{inf_{pour}}$, $s_{inf_{par}}$. This allows us to distinguish, for example, between "finir de" (*to* finish *doing something*) and "finir par" (*to* end up *doing something*). The infinitive headed by "pour" occurs in constructions like "trop tôt pour ..." (*too early to ...*).

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

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

$$\frac{\genfrac{}{}{0pt}{}{x \vdash B}{\vdots} \atop \dfrac{X \circ x \vdash A}{X \vdash A/B}} \; /I \qquad\qquad \frac{\genfrac{}{}{0pt}{}{x \vdash B}{\vdots} \atop \dfrac{x \circ X \vdash A}{X \vdash B \backslash A}} \; \backslash I$$

$$\frac{X[Y] \vdash B \quad Z \vdash B \backslash_1 A}{X[Y \circ_1 Z] \vdash A} \; \backslash_1 E \qquad \frac{\genfrac{}{}{0pt}{}{x \vdash B}{\vdots} \atop \dfrac{X[Y \circ x] \vdash A}{X[Y] \vdash A/\Diamond_1 \Box_1 B}} \; /\Diamond_1 \Box_1 I$$

Table 1:
Logical rules for
multimodal categorial
grammars

though passives are not annotated as such in this treebank – and the categorial treatments of Carpenter (1991) and Hockenmaier and Steedman (2007). The different subtypes of s and pp are implemented using first-order variables and unification, following Moot (2014) and Morrill (1994, Section 2.1).

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. In what follows, we will often simply write s instead of s_{main}.

To make this article understandable to the reader not intimately familiar with modern type-logical grammars, all examples in the text use the simplified presentation of Table 1. The intrepid reader interested in the full technical details can find the complete presentation in Appendix A, with further applications in Appendix B.

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 (the $\backslash E$ rule is symmetric).

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

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

The two rules at the bottom row of the table 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. This tree Y can occur at any depth in the tree $X[Y]$, including the root, i.e. Y can be equal to $X[Y]$.[2]) 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.

$$\frac{\dfrac{appauvrit \vdash (np\backslash s)/np \quad la \circ CGT \vdash np}{appauvrit \circ (la \circ CGT) \vdash np\backslash s} \; /E \quad dangereusement \vdash (np\backslash s)\backslash_1(np\backslash s)}{(appauvrit \circ_1 dangereusement) \circ (la \circ CGT) \vdash np\backslash s} \; /E$$

Each occurrence of the introduction rules $/I$, $\backslash I$ and $/\Diamond_1\Box_1$ uses a distinct syntactic variable x which is unique to the proof; therefore, in the case of a proof containing multiple introduction rules, the hypothesis corresponding to an introduction rule can always be uniquely determined by this variable name (we can use any naming convention to ensure this; common choices are $x_0, x_1, \ldots$ or, for shorter proofs, x, y, z).

The $/\Diamond_1\Box_1$ rule is an *extraction* rule, extracting a B constituent from any right branch inside an X constituent.[3] Comparing the rule $/\Diamond_1\Box_1 I$ to the rule $/I$, we can see that $/I$ is the special case of $/\Diamond_1\Box_1 I$ where the context $X[]$ is empty (i.e. where $X[Y]$ is equal to Y). From the point of view of semantics the two rules are the same — both correspond to abstraction over the semantic variable assigned to the B formula which is withdrawn by the rule — but the rule $/\Diamond_1\Box_1 I$ can

---

[2] For adverbs, as here, Y is typically the verb, but in principle infixation is possible anywhere (an admitted oversimplification, which can be remedied by a more sophisticated treatment of mode information).

[3] For readers familiar with the Displacement calculus (Morrill *et al.* 2011), the infixation construction $A\backslash_1 B$ corresponds to $\check{} B \downarrow A$ and the extraction construction $A/\Diamond_1\Box_1 B$ to $\hat{} (A \uparrow B)$.

apply in a larger number of syntactic contexts. As an example, in the following sentence

(1)   l'   argent dont   elle est responsable
       the money for which she is   responsible

the relativizer "dont" is assigned the formula $(n\backslash n)/(s/\lozenge_1\square_1 pp_{de})$ meaning it is looking to its right for a sentence missing a prepositional phrase headed by the preposition "de" (*for*). The subformula $\lozenge_1\square_1 pp_{de}$ should be seen as a special type of pp_{de} formula. Unlike a normal pp_{de} argument, it can occur on *any* right branch, no matter how deeply nested (unlike the rules for $/I$ in Table 1, which apply only when the argument is the immediate right daughter). This means "dont" can take a phrase such as "elle est responsable" (*she is responsible*), where "responsable" is analysed as an adjective which first selects a pp_{de} to its right, as an argument since we can assign it $s/\lozenge_1\square_1 pp_{de}$ as follows.

$$
\dfrac{
\dfrac{elle}{np}\ Lex \quad
\dfrac{
\dfrac{est}{(np\backslash s_{main})/(n\backslash n)}\ Lex \quad
\dfrac{
\dfrac{responsable}{(n\backslash n)/pp_{de}}\ Lex \quad \dfrac{}{x \vdash pp_{de}}\ Hyp
}{responsable \circ x \vdash n\backslash n}\ /E
}{est \circ (responsable \circ x) \vdash np\backslash s_{main}}\ /E
}{
\dfrac{elle \circ (est \circ (responsable \circ x)) \vdash s_{main}}{elle \circ (est \circ responsable) \vdash s_{main}/\lozenge_1\square_1 pp_{de}}\ /\lozenge_1\square_1 I
}\ \backslash E
$$

As shown in the proof, the extraction analysis starts by assuming a pp_{de} hypothesis (corresponding to a pp_{de} gap in a mainstream generative grammar analysis) then derives a sentence s_{main} using the elimination rules. Finally, the introduction rule "binds" the gap: it removes the leaf x corresponding to the pp_{de} hypothesis and binds it semantically. The proof above also shows why the assignment of the simpler formula $(n\backslash n)/(s/pp_{de})$ to the word "dont" doesn't suffice: in the penultimate step of the proof, we have derived $elle \circ (est \circ (responsable \circ x))$ of type s_{main}, whereas for the $/I$ rule to apply we would need a differently bracketed structure such as $(elle \circ (est \circ responsable)) \circ x$, with x the immediate right daughter of the root node.[4] Appendix A gives a

---

[4] To be precise, this example only shows the need for a form of associativity, but slightly more complicated examples like "for which she was responsible in 1992" show that associativity alone is no solution. Examples of this kind have been a driving force in the development of extensions of the Lambek calculus.

detailed treatment of the unabbreviated version of this proof, showing notably how to *derive A* from $\Diamond_1 \Box_1 A$. To summarize, formulas of the form $\Diamond_1 \Box_1 A$ are special types of *A* formulas that can be extracted from deeply embedded positions.

3   THE FRENCH TREEBANK

The French Treebank (FTB, Abeillé *et al.* 2000) 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 phrase structure grammars (Arun and Keller 2005), dependency grammars (Candito *et al.* 2009; Guillaume and Perrier 2012), lexical-functional grammars (Schluter and van Genabith 2008), and tree adjoining grammars (Dybro-Johansen 2004).

For its annotation, the FTB uses simple, rather flat trees with some functional syntactic annotation (subject, object, infinitival argument, etc.). Consecutive multiword-expressions have been merged in the annotation and neither traces nor discontinuous dependencies have been annotated.

Consider the following sentence from the French Treebank.

(2)   À cette époque, on avait dénombré cent_quarante candidats
      at that time,   we had   counted   hundred-forty candidates
      'At that time, there were 140 candidates.'

Its FTB annotation is shown in Figure 1. We can see that verb clusters are treated as constituents (labelled *VN*) and that the arguments of the verb occur as sisters of this verbal cluster. For example, the object noun phrase in Figure 1 is the sister of the *VN*. However, as we will see in Section 4.3, we obtain a much neater analysis when we treat the object as an argument of "dénombré" (*counted*), which is the past participle of a transitive verb.

4   GRAMMAR EXTRACTION

Grammar extraction algorithms for categorial grammars follow a general methodology – see, for example, Buszkowski and Penn (1990),
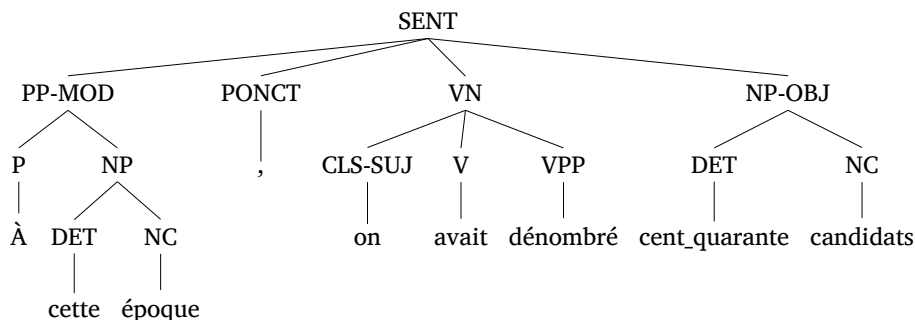
Figure 1: An example sentence from the French Treebank

Moortgat and Moot (2001), Hockenmaier and Steedman (2007) and Sandillon-Rezer (2013), 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 to convert the FTB into the TLGbank 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 (e.g. *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.[5]

All steps are done by a single Prolog tree transformation, then verified and corrected manually (either by writing an ad hoc tree transformation script or by manually editing the output, then verifying that the result remains a valid derivation). Since the FTB annotation makes

---

[5] Subject clitics are treated as normal *np* subjects. Object clitics, such as the object clitic "l" in "Marie l'aime" (Marie him-clitic loves, *Marie loves him*) are assigned the formula $(np \backslash s)/((np \backslash s)/\Diamond_1 \Box_1 np)$ following the analysis of Moot and Retoré (2006). By assigning these higher-order formulas to the clitics, we can assign a normal transitive verb formula to "aime" (*loves*). Only the reflexive clitic "se" and the clitic "y" in the construction "il y a" (*there is/are*) are treated as arguments of the verb (with formulas cl_{se} and cl_y, respectively).

the distinction between modifiers and arguments only for certain categories (sentences, infinitive phrases, present participle phrases, but not past participle phrases or noun phrases), this information is not explicitly annotated for many major categories (the extraction script treats these cases as modifiers for noun phrases and as arguments for other categories, such as past participle phrases). In addition, all forms of the verb "être" (*to be*) with a past participle as argument have been manually changed to passive whenever this was a passive construction.[6]

In Step 4, which harmonizes the annotation of coordinations, many simple coordinations are treated correctly by the extraction script. Special care has been taken of the punctuation symbols, which in many cases are manually given a coordination-like formula assignment, and of gapping, which must be treated manually as well (the treatment of gapping is presented in detail in Appendix B.4).

Finally, relative pronouns are treated by the extraction script as arguments of the immediately following verb, which is correct in many cases but needs to be manually verified for all occurrences.

In sum, after a pass of the extraction script, many constructions are manually verified and corrected. To give an indication of the amount of manual cleanup done: simply running the Prolog script on the treebank results in a lexicon with 5,240 distinct formulas assigned to the words of the lexicon (Moot 2010b) (note that this is without a distinction between passives and past participles), but after cleanup there are 1,101.

From Section 4.1 to Section 4.5, we will treat each of the stages of the extraction algorithm in turn.

4.1   *Splitting multiword expressions*

The French Treebank treats many multiword expressions as single nodes in the annotation. For example, the expression "dépôt de bilan" (*voluntary liquidation*) occurs as "dépôt_de_bilan"; similarly, as shown in Figure 1, numbers such as "cent_quarante" (*140*) are analysed as

---

[6] Not all occurrences of passives are accompanied by a form of "to be": adjectival uses of passive (e.g. in English "books written by Stephen King") are treated automatically, whereas extraposed passive phrases, such as "Elaborated with the greatest discretion, this project...", are handled during the manual correction of coordination/punctuation.

a single word. Though very good solutions exist to detect these automatically in a separate preprocessing step (see, for example, Constant *et al.* 2011), we have decided to split all these into their separate words in order not to have to depend on additional components.

Fortunately, the French Treebank also annotates the internal structure for many of these complex lexical lemmas, so we can find that "dépôt de bilan" has the internal structure [noun, preposition, noun] and use this to automatically annotate the expression according to the basic case discussed below, so this step requires little human intervention.

4.2 *The basic case*

The heart of the algorithm binarizes the trees from the French Treebank and separates the daughters of a node into functors/heads, arguments, and modifiers. This step is done automatically, using a version of the classic "head percolation" table (Magerman 1994) similar to the ones used for other categorial grammar extraction algorithms (Hockenmaier and Steedman 2007; Moortgat and Moot 2001; Moot 2010a).

The automated part of the extraction algorithm recursively descends each node and successively performs each of the different transformations described here, as well as the refinements described in Sections 4.3 to 4.5. Thus, even though these cases are described separately for ease of exposition, they apply together at each node.

For example, the following sentence

(3)   le   score correspondait à peine au   tiers de l'   objectif
      the score corresponded barely to a third of the goal
      mensuel
      monthly
      'the score barely corresponded to a third of the monthly goal'

has the French Treebank annotation shown in Figure 2. In the figure, the multiword expression "à peine" (*hardly*) has already been separated into its component words in the previous step of the algorithm.

The binarization step first selects the head of the constituent (the head percolation table first tries to find a verbal group VN as the head of a sentence SENT) and then combines it first with the sisters to its right, then with the sisters to its left, as shown in the figure below.

Figure 2:
Initial French
Treebank tree.
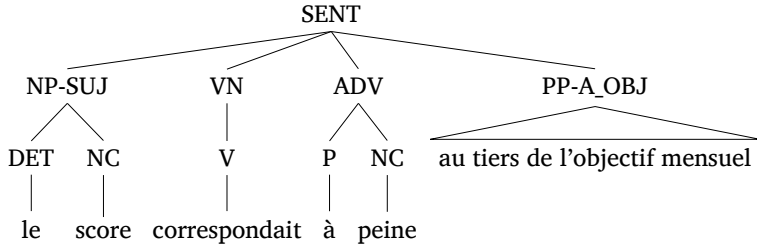
Figure 3:
The tree of
Figure 2 after
binarization.
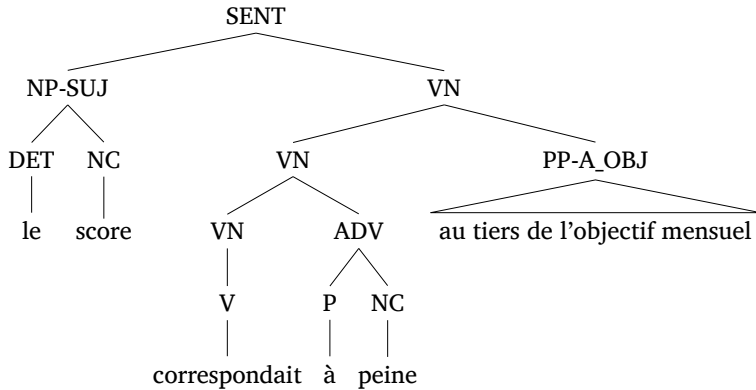
The label of the newly created nodes remains the same; VN in this
case. The resulting tree, shown in Figure 3 has only unary and binary
branches.

Next, a similar table of defaults decides for each binary branch
if the pair of nodes concerned are a functor and its argument or a
modifier and a category it modifies. So in the current example ADV is
treated as a modifier whereas PP-A_OBJ is treated as an argument. A
functor and argument are given the formulas F/A and A, if the argu-
ment occurs on the right, or A and $A\backslash F$ if the argument occurs on the
left, where F is the formula assigned to the parent node and A is the for-
mula corresponding to the syntactic label of the argument node (this
is again performed by looking up the values in a table, which indicates
for example, that NP corresponds to np and PP-A_OBJ corresponds to
pp_a). For modifiers, the modifier is assigned F/F if it occurs on the
left and $F\backslash F$ if it occurs on the right, where F is the formula assigned
to the parent node; the sister node of the modifier will therefore be
assigned the same formula F as the parent node.
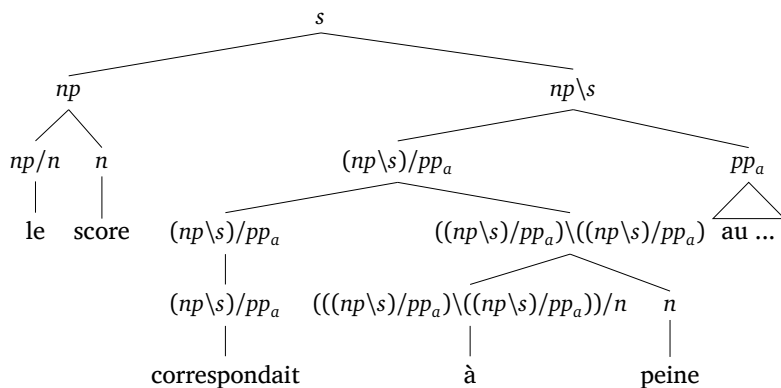
Figure 4: First derivation corresponding to Figure 3, using only elimination rules.

This translates the binarized tree of Figure 3 into the tree shown in Figure 4. This tree gives a full description of a derivation using only the elimination rules $/E$ and $\backslash E$: suppressing the unary modes, we can label each pair of sisters uniquely with one of these rules by looking only at their formulas, either F/A and A or A and $A\backslash F$; the distinction between modifiers and other functors is no longer relevant now, modifiers are simply those formulas where $F = A$.

So far, the extraction algorithm has followed the classic categorial grammar extraction methodology of Buszkowski and Penn (1990) and Moortgat and Moot (2001). However, the tree above gives a rather complicated formula to the modifier "à peine" (*hardly*). Moreover, this formula would change with the formula assigned to the verb it modifies – requiring a different formula for transitive verbs, intransitive verbs, auxiliary verbs, etc. – resulting in unnecessary duplication of lexical entries for all adverbs. As we have seen in Section 2 with the adverb "dangereusement" (*dangerously*), we can choose an infixation solution and treat all adverbs as VP modifiers as shown in Figure 5.

From this tree, we can again obtain a complete derivation, this time using the $/E$, $\backslash E$ and $\backslash_1 E$ rules of Table 1, though we now need the word order of the original sentence to determine the position of the adverb. The $\backslash_1 E$ rule essentially plays the role of the crossing composition rules used for similar situations in the CCGbank (Hockenmaier and Steedman 2007). This simplification is performed automatically whenever a complex verb-modifier formula would be assigned to an adverb.

Figure 5:
A version of
the derivation
of Figure 4
using a simpler
lexical entry
for the adverb
"à peine"
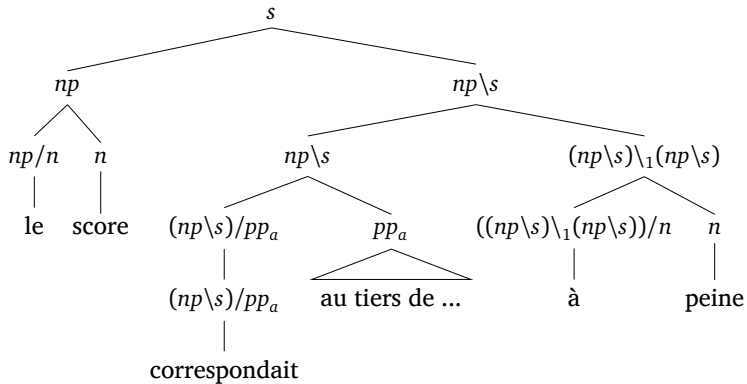
4.3   *Verb clusters*

As discussed in Section 3, verb clusters (which include clitics and some adverbs) and the arguments of verbs are sisters in the FTB annotation trees. While this wasn't a problem for the simple cases treated in the previous section, this becomes problematic in the case of a complex verbal group. Figure 6 shows an example corresponding to sentence (4) (Figure 1 back on page 235 requires a similar treatment).

(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 the 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" (past participle of the verb "pouvoir", *can*). At the bottom of Figure 6 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 Moot 2014 or Moortgat 2011, Section 2.7; in the figure, x is instantiated to *ppart*).

The extraction script automatically rebrackets the verb clusters as indicated above and treats any arguments of the verb cluster as arguments of the final verb in the cluster. This step requires very few manual corrections.

4.4 *Coordination and punctuation symbols*

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

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

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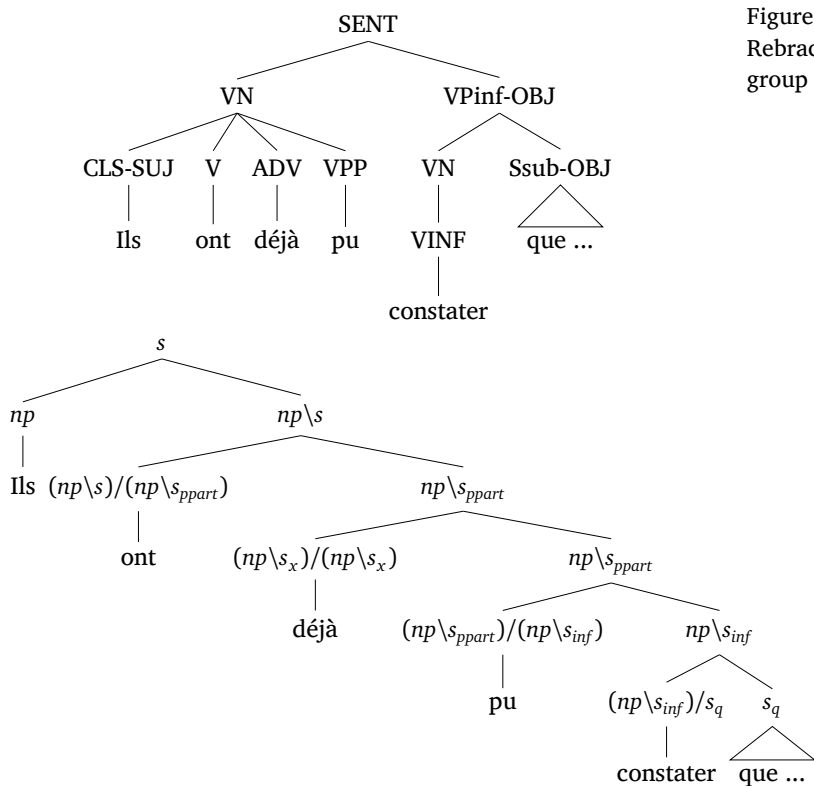
Figure 6:
Rebracketing a verbal group and its arguments

[241   ]

Figure 7:
Coordination
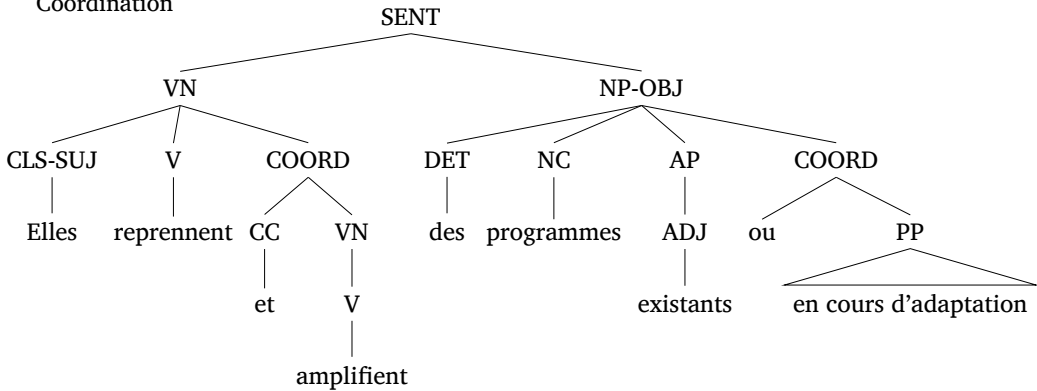
Figure 7 shows the FTB syntactic structure of sentence (5). In categorial grammars, conjunctions like "ou" (*or*) are generally assigned instances of the formula $(X \backslash X)/X$ (for a contextually appropriate choice of the formula X). The first conjunction is of the two transitive verbs (instantiating X with the formula $(np \backslash s_{main})/np$) that 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 to be 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.

The (somewhat simplified) FTB annotation of sentence (6) of Figure 8 shows another problem: appositives, which are treated by assigning a coordination-like formula to the punctuation symbol preceding them (a similar solution is used for parentheticals and for most extrapositions).[7] An additional complication in this example is that we have

---

[7] 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 celà" is assigned $s/(s/\Diamond_1 \Box_1 pp_a)$ allowing it to function as a long-distance pp argument to "s'ajoute", as we have seen for the $s/\Diamond_1 \Box_1 pp_{de}$ argument of "dont" in Section 2.
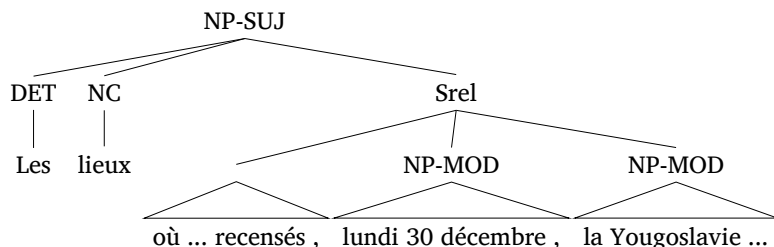
to distinguish between the NP-MOD temporal adverb, which modifies the verb "recensés" (*reported*), and the NP-MOD for the appositive, which conjoins to "Les lieux" (*the places*) with the NP containing "la Yougoslavie" (*Yugoslavia*).

As the example shows, these cases are difficult to infer from the information provided by the FTB annotation alone, and therefore must be annotated manually; in total a bit more than 20% of the punctuation symbols – over ten thousand punctuation symbols – are assigned coordination-like categories. This complicated treatment of punctuation is not necessary for standard phrase structure parsers but given that in a categorial grammar analysis we want coordination-like punctuation to behave *semantically* like coordination, some special treatment of coordination is necessary.

More complex forms of coordination, such as right-node raising and gapping, require a more sophisticated treatment, which is discussed in Appendix B.

4.5   *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:
        first   handicap to which it was agreed to   attack:
        l'inflation
        the inflation

Figure 9 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, the default automatic treatment assigns them as arguments of the next occurring verb. Even though this is a reasonable default, it still produces many errors. In the example above, it would assign the pp_a as argument of the main verb "convenait" (*to agree*), which is a possible assignment for this verb but is incorrect in the current case. As a consequence all relative pronouns, wh-pronouns, and clitics – a total of over 3,000 occurrences in the corpus – have been manually verified and, where necessary, corrected with the appropriate long-distance dependencies. At the bottom of Figure 9, the manually added long-distance dependency is shown (for reasons of horizontal space, the subproof of "de s'attaquer pp_a" has been stretched, as indicated by the dots).

5   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 transitive and intransitive, 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 1,101 distinct formulas, though only 800 of these occur more than once and, 684 more than twice and 570 at least five times. The lion's share of these rare formulas are assigned to frequently occurring words, such as "et" (*and*) and verbs, appearing in unusual syntactic constructions.

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).

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*) has 44 formulas (in part because of possible combinations with "que", *than*), the prepositions "pour" (*for/to*), "en" (*in/while*) and "de" (*of/from*) have 43, 42 and 40 formulas respectively, and the verb "est" (*is*) has 39 formulas.

Although this kind of lexical ambiguity may seem like an important problem when using the extracted lexicon for parsing, well-known techniques such as *supertagging* (Bangalore and Joshi 2011), which assign the contextually most likely set of formulas (supertags) to each word, can be used to reduce the lexical ambiguity to an acceptable level. To give an idea of how effective this strategy is in the current context and with the reduced lexicon of 761 formulas: using the supertagger of Clark and Curran (2004) and assigning only the most
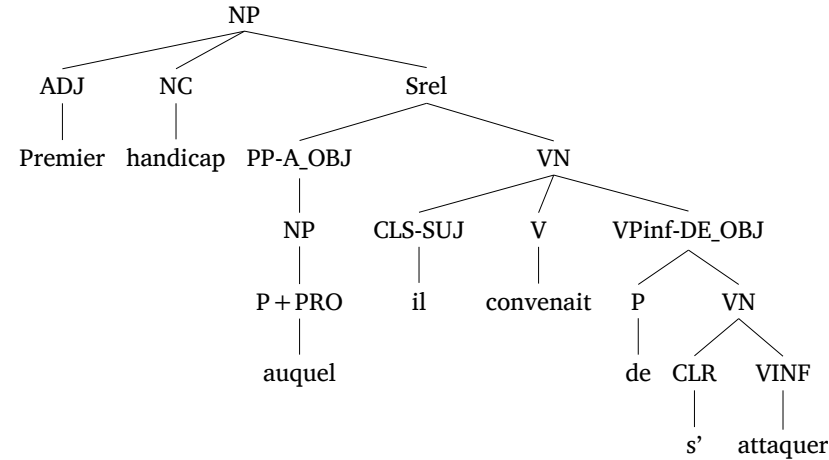
Figure 9:
Adding traces to
the output

$$\frac{\text{auquel}}{(n\backslash n)/(s/\Diamond_1\Box_1 pp_{\text{à}})}\,[\textit{Lex}]$$

$$\frac{\text{il}}{np}\,[\textit{Lex}]\quad \frac{\text{convenait}}{(np\backslash s)/(np\backslash s_{\text{di}})}\,[\textit{Lex}]$$

$$\frac{\text{de}}{(np\backslash s_{\text{di}})/(np\backslash s_{\text{i}})}\,[\textit{Lex}]\quad \frac{\text{s'}}{cl_{\text{r}}}\,[\textit{Lex}]$$

$$\frac{\text{attaquer}}{(cl_{\text{r}}\backslash(np\backslash s_{\text{i}}))/pp_{\text{à}}}\,[\textit{Lex}]\quad \frac{}{\text{p}_0 \vdash pp_{\text{à}}}\,[\textit{Hyp}]_1$$

$$\frac{\text{a}\circ \text{p}_0 \vdash cl_{\text{r}}\backslash(np\backslash s_{\text{i}})}{}\,[/E]$$

$$\frac{\text{s'}\circ(\text{a}\circ \text{p}_0)\vdash np\backslash s_{\text{i}}}{\text{de}\circ(\text{s'}\circ(\text{a}\circ \text{p}_0))\vdash np\backslash s_{\text{i}}}\,[/E]\,[\backslash E]$$

$$\vdots$$

$$\frac{\text{de}\circ(\text{s'}\circ(\text{a}\circ \text{p}_0))\vdash np\backslash s_{\text{i}}}{\text{c}\circ(\text{de}\circ(\text{s'}\circ(\text{a}\circ \text{p}_0)))\vdash np\backslash s}\,[/E]$$

$$\frac{\text{il}\circ(\text{c}\circ(\text{de}\circ(\text{s'}\circ(\text{a}\circ \text{p}_0))))\vdash s}{\text{il}\circ(\text{c}\circ(\text{de}\circ(\text{s'}\circ\text{a})))\vdash s/\Diamond_1\Box_1 pp_{\text{à}}}\,[\backslash E]\,[/I]_1$$

$$\frac{}{\text{auquel}\circ(\text{il}\circ(\text{c}\circ(\text{de}\circ(\text{s'}\circ\text{a}))))\vdash n\backslash n}\,[/E]$$

[245]

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 the correct formula to 98.4% of all words (for the FTB section of the TLGbank, using ten-fold cross-validation).

Supertagging does not solve the problem of data sparseness: for the supertagger, formulas which are seen only once or twice in the training data are not fundamentally different from formulas which do not occur at all. However, since these are exceptional cases, this has little effect on the coverage of the parser: Clark and Curran (2007) use only categories occurring at least 10 times for their parser based on the CCGbank and still obtain 99.58% coverage on unseen sentences.

We will discuss the performance of the supertagger in more detail, especially on sentences *outside* of the French Treebank, while discussing bootstrapping in Section 7.1.

6   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 rules besides the standard combinatory schemata – notably for extraposition and coordination,[8] but also to transform passives $np \backslash s_{pass}$ into adjectives $n \backslash n$ and (bare) nouns n into noun phrases np – the TLGbank uses no non-logical rules. As a result, the lexicon of the type-logical treebank does more of the work. The lexicon is bigger and consequently, the tasks of the supertagger and the parser are more difficult in comparison with the CCG supertagger (Clark and Curran 2007). The supertagger's precision is similar – 98.4% correct in both cases – though the number

---

[8] To give an idea of the form of these rules, there is an extraposition rule transforming "np" (that is, a noun phrase followed by a comma) into a sentence modifier s/s and a set of rules transforming constructions like "X and X" (that is, the word "and" occurring between two expression of the same category X) to X, see Section 2.5.5 of Hockenmaier and Steedman (2005) for more details.

of lexical formulas per word is higher – 2.3 for the TLGbank versus 1.7 for the CCGbank. The number of lexical formulas per word is an important factor for parsing speed.

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

1. the first option is to allow non-logical rules of the same style as those used for the CCGbank,

2. 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 Morrill 2011 for a particularly nice solution).

Since reducing the lexical ambiguity increases parsing speed but adding rules (as in option 1) or complicating the formulas (as in option 2) will reduce it, a careful evaluation of the benefits should be made. We leave to future research the transformation of the TLGbank in these two ways.

7   TOOLS AND RESOURCES

To facilitate 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 (Levy and Andrew 2006) for browsing and querying the French Treebank (as well as some of its transformations), Lefff (Sagot 2010) for lemmatizing and related tasks, the C&C tools (Clark and Curran 2004) for training POS-tag and supertag models using the annotated corpus, and a chart parser strongly inspired by Shieber *et al.* (1995) for parsing with the resulting grammar.

Figure 10 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-shaded percentage of the block to the left of the formula gives a visual indication of the probability assigned 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.
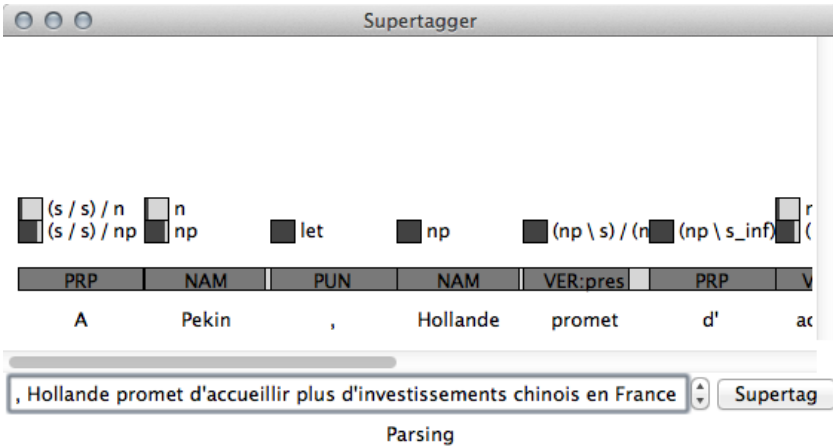
Figure 11 shows a screenshot of the "vertical" interface to the parser and supertagger. This is an interactive interface, allowing the user to select (or type in) the desired formula – to help prevent errors, the current frequency of the chosen formula for the current word is displayed after a manual choice of the formula – as well as allowing the user to select the parser rule applications by clicking on one of the premises for a rule (an additional dialog pops up if the rule choice is ambiguous, which happens infrequently). The weight column shows the log-probability of the item.[9]
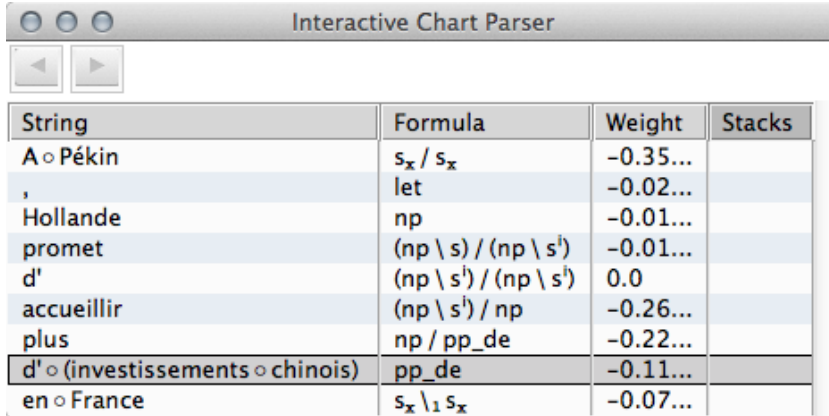
7.1   *Bootstrapping*

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 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

---

[9] The current implementation of the parser is not statistical in the sense that the rule applications do not have a probability assigned to them (the supertags do, so the parser outputs the first parse found for the most probable combination of supertags which allows a parse). However, the source code has the required hooks to add a probability model for the rule applications, whereas the required probabilities can be estimated from the treebank itself.

Figure 11: Screenshot of the interactive parser

String	Formula	Weight	Stacks
A ∘ Pékin	s_x / s_x	−0.35...	
,	let	−0.02...	
Hollande	np	−0.01...	
promet	$(np \backslash s) / (np \backslash s^i)$	−0.01...	
d'	$(np \backslash s^i) / (np \backslash s^i)$	0.0	
accueillir	$(np \backslash s^i) / np$	−0.26...	
plus	np / pp_de	−0.22...	
d' ∘ (investissements ∘ chinois)	pp_de	−0.11...	
en ∘ France	$s_x \backslash_1 s_x$	−0.07...	

words). Most of these sentences come from the Sequoia treebank (Candito and Seddah 2012) and the French Timebank (Bittar 2010). The observed accuracy of the supertagger for these sentences from the *L'Est Républicain* newspaper is slightly lower than the results reported in Section 5: in 88.1% of cases, the best supertag is correct, and in 97.6% of cases the correct supertag has probability greater than 1% of the best supertag (as compared to 90.6% and 98.4% respectively for the cross-validated results). Part of this difference might be attributable 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).

Table 2 shows the accuracy of the Part-of-Speech tagger and of the supertagger for the different sub-corpora. The columns POS and Super list the accuracy of the Part-of-Speech tagger and of the supertagger respectively for the different corpora. Performance degrades gracefully for the different newspaper corpora (the French treebank and more modern articles in *Le Monde* being presumably the most similar, whereas the articles in *L'Est Républicain* from Sequoia and the French Timebank have a slightly reduced supertagger performance) but it shows a somewhat more important reduction for the literary corpus of travelogues in the Pyrenees of Itipy (Lefeuvre *et al.* 2012; Moot 2012).

Table 2:
Supertagger and
Part-of-Speech tagger
performance on the
different sections
of the corpus

Corpus	POS	Super	0.1	0.01	F/w
French Treebank	97.8	90.6	96.4	98.4	2.3
Le Monde 2010	97.3	89.9	95.8	97.9	2.2
L'Est Républicain	97.3	88.1	94.8	97.6	2.4
Itipy/Forbes	95.7	86.7	93.8	97.1	2.6

The 0.1 and 0.01 columns indicate the supertagger's performance when all supertags with probability greater than β ($=0.1$ or 0.01) times the probability of the most likely supertag have been included. The column F/w indicates how many supertags this is per word for $\beta = 0.01$ (for $\beta = 0.1$ this number is around 1.4). We can see that even though the supertagger's performance for the best supertag (in the Super column) reduces steadily – from 90.6 on the main corpus to 86.7 on the Itipy corpus, a 3.9 percentage points difference – when using multiple supertags, this difference is greatly reduced (from 98.4 to 97.1, a 1.3 percentage points difference).

Even in the more difficult context of the Itipy corpus, the parser/supertagger combination (with $\beta = 0.01$) finds a complete analysis for 88.6% of the sentences in this subcorpus. We expect this figure to improve when better search heuristics, such as those described by Clark and Curran (2007), are used to deal with the increased number of formulas per word. To give an indication that even the current parser implementation performs well: the only other parsing statistics I've seen for the Itipy corpus are given by Nguyen (2012), who reports that a total of 18.5% of the sentences in the Itipy corpus were successfully parsed using an off-the-shelf parser.

7.2 *Availability*

All the tools and resources are available from the author under the GNU Lesser General Public License.

```
http://richardmoot.github.io/TLGbank/
```

An unfortunate exception to this is the main part of the Typelogical Treebank itself: being a derived work of the French Treebank, it is available only to those who have a license for the original treebank (contact to author for access to the private Git). The Sequoia part of the treebank and the models derived from the complete treebank are freely available, however.

8   CONCLUSION

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

APPENDIX

A   COMPLETE LOGICAL RULES

Table 3 lists the full set of rules for multimodal categorial grammars. Binary modes i range over $\{\epsilon, 1, 2, 3, l, r\}$ – although we will continue to write $X \circ_\epsilon Y$ as $X \circ Y$ and $A/_\epsilon B$ as A/B, etc. – and unary modes j range over $\{0, 1, l, r\}$.

A.1   *The unary connectives*

The rules for $\Diamond$ and $\Box$ may require some additional explanation for people unused to multimodal type-logical grammars. Whereas the rules for $\bullet$, $/$, and $\backslash$ produce binary trees labelled by indices — with the $\bullet I$, $/E$, and $\backslash E$ rules *constructing* trees (i.e. combining previously derived trees X and Y into a single tree $X \circ_j Y$) and with the $\bullet E$, $/I$, and $\backslash I$ rules *removing* binary branches — the rules for $\Box$ and $\Diamond$ produce and remove unary branches. So the $\Box E$ rule states that if we have previ-

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

$$\frac{X \vdash A/_i B \quad Y \vdash B}{X \circ_i Y \vdash A} \; /E \qquad \frac{X \vdash B \quad Y \vdash B\backslash_i A}{X \circ_i Y \vdash A} \; \backslash E$$

$$\begin{array}{c} x \vdash B \\ \vdots \\ \dfrac{X \circ_i x \vdash A}{X \vdash A/_i B} \; /I \end{array} \qquad \begin{array}{c} x \vdash B \\ \vdots \\ \dfrac{x \circ_i X \vdash A}{X \vdash B\backslash_i A} \; \backslash I \end{array}$$

$$\begin{array}{c} x \vdash A \quad y \vdash B \\ \vdots \\ \dfrac{Y \vdash A \bullet_i B \quad X[x \circ_i y] \vdash C}{X[Y] \vdash C} \; \bullet E \end{array} \qquad \frac{X \vdash A \quad Y \vdash B}{X \circ_i Y \vdash A \bullet_i B} \; \bullet I$$

$$\frac{X \vdash \Box_j A}{\langle X \rangle^j \vdash A} \; \Box E \qquad\qquad \frac{\langle X \rangle^j \vdash A}{X \vdash \Box_j A} \; \Box I$$

$$\begin{array}{c} x \vdash A \\ \vdots \\ \dfrac{Y \vdash \Diamond_j A \quad X[\langle x \rangle^j] \vdash C}{X[Y] \vdash C} \; \Diamond E \end{array} \qquad \frac{X \vdash A}{\langle X \rangle^j \vdash \Diamond_j A} \; \Diamond I$$

ously derived X to be of type $\Box_j A$, then $\langle X \rangle^j$ is of type A; we remove the $\Box_j$ connective and add a unary branch labelled by the index j. Symmetrically, the $\Box I$ rule states that if we have derived $\langle X \rangle^j$ (i.e. we have an initial unary branch labelled j with a daughter subtree X) to be of type A then the tree X by itself is of type $\Box_j A$.

The elimination rules for the product $\bullet$ and the diamond $\Diamond$ may appear a bit odd: they are similar to the disjunction elimination rule in intuitionistic logic and involve an arbitrary formula C. The $\Diamond E$ rule gives instructions on how to use a formula $\Diamond_j A$ once we have derived it (as the subproof of the left premise of the rule) by stating that if we can use a formula A labelled with a fresh variable x to derive any tree X (of any formula C) such that this x corresponding to A occurs as a leaf with a unary branch labelled j as its immediate parent (as indicated by the tree term $X[\langle x \rangle^j]$), then we can conclude that this tree X with the unary branch j and leaf x replaced by Y (the tree corresponding to $\Diamond_j A$) is also a tree

Infixation

$$\frac{V[(X \circ Y) \circ_1 Z] \vdash C}{V[X \circ (Y \circ_1 Z)] \vdash C} \; MA \qquad\qquad \frac{V[(X \circ Y) \circ_1 Z] \vdash C}{V[(X \circ_1 Z) \circ Y] \vdash C} \; MC$$

Extraction

$$\frac{V[X \circ (Y \circ \langle Z \rangle^1)] \vdash C}{V[(X \circ Y) \circ \langle Z \rangle^1] \vdash C} \; MA\Diamond_1 \qquad\qquad \frac{V[(X \circ \langle Z \rangle^1) \circ Y] \vdash C}{V[(X \circ Y) \circ \langle Z \rangle^1] \vdash C} \; MC\Diamond_1$$

Left-node raising/right-node raising

$$\frac{V[(\langle X \rangle^0 \circ Y) \circ Z] \vdash C}{V[\langle X \rangle^0 \circ (Y \circ Z)] \vdash C} \; MA_l\Diamond_0 \qquad\qquad \frac{V[X \circ (Y \circ \langle Z \rangle^0)] \vdash C}{V[(X \circ Y) \circ \langle Z \rangle^0] \vdash C} \; MA_r\Diamond_0$$

In situ binding

$$\frac{V[X \circ \langle Y \rangle^2] \vdash C}{V[\langle X \rangle^r \circ_2 Y] \vdash C} \; I_{2r} \qquad\qquad \frac{V[\langle X \rangle^2 \circ Y] \vdash C}{V[\langle Y \rangle^l \circ_2 X] \vdash C} \; I_{2l}$$

$$\frac{V[X \circ (Y \circ_2 Z)] \vdash C}{V[(X \circ_r Y) \circ_2 Z] \vdash C} \; MA_{2r} \qquad\qquad \frac{V[(X \circ_2 Z) \circ Y] \vdash C}{V[(X \circ_l Y) \circ_2 Z] \vdash C} \; MC_{2l}$$

Quoted speech

$$\frac{V[(X \circ_3 Y) \circ Z] \vdash C}{V[X \circ_3 (Y \circ Z)] \vdash C} \; MA_3 \qquad\qquad \frac{V[Y \circ (X \circ_3 Z)] \vdash C}{V[X \circ_3 (Y \circ Z)] \vdash C} \; MC_3$$

of type C. In other words, $X[\langle x \rangle^j]$ becomes $X[Y]$ as indicated in the rule.

As an example, we show that if a tree Y is of type $\Diamond_j \Box jA$ then this tree is also of type A (for all formulas A and unary indices j), as already alluded to in Section 2.

$$\frac{Y \vdash \Diamond_j \Box_j A \qquad \dfrac{\dfrac{x \vdash \Box_j A}{\langle x \rangle^j \vdash A} \; \Box E}{}}{Y \vdash A} \; \Diamond E$$

If Y is of type $\Diamond_j \Box jA$, then we start the subproof on the right using the hypothesis x of type $\Box_j A$. Then we apply the elimination rule for $\Box$ to produce the tree $\langle x \rangle^j$ of type A. But now, we are immediately in the right configuration to apply the $\Diamond E$ rule (it is the special case

where the context $X[\,]$ is empty) and this allows us to replace $\langle x \rangle^j$ by Y, thereby proving that Y is of type A as required.

Although these patterns of derivability are interesting and can be used to give accounts of case and other forms of subtyping (Bernardi and Moot 2003), our interest here lies in the fact that they give access to structural rules which can rearrange our derived trees in controlled ways. The structural rules are listed in Table 4. The double line for the in situ binding rules indicate that these rules can be applied in both directions: top-to-bottom and bottom-to-top.

   Even though this looks like a rather large list, these are principally instantiations of the well-known universal rule schemata of mixed associativity and mixed commutativity (see Moortgat 2011 and Moot and Retoré 2012 for commentary, and Vermaat 2005 for arguments that these structural rules are truly universal).

   For the grammar engineer, the structural rules give us great flexibility and modularity when designing our grammars (although it could be argued that there is *too* much flexibility to this). However, the account given for different linguistic phenomena follows the conventional wisdom of categorial grammars and, as discussed in the next subsection, our annotation choices have been designed to be compatible with other modern type-logical grammars. So there has been a conscious choice not to create the smallest possible lexicon (at the cost of additional structural rules) but to keep the set of structural rules to the current set of instantiations of well-known schemata.

   The abbreviated proof from Section 2, is repeated below.

$$\cfrac{\cfrac{appauvrit \vdash (np\backslash s)/np \quad la \circ CGT \vdash np}{appauvrit \circ (la \circ CGT) \vdash np\backslash s}\,/E \quad dangereusement \vdash (np\backslash s)\backslash_1(np\backslash s)}{(appauvrit \circ_1 dangereusement) \circ (la \circ CGT) \vdash np\backslash s}\,/E$$

Using the structural rules of Table 4, this proof looks as follows.

$$\cfrac{\cfrac{\cfrac{appauvrit \vdash (np\backslash s)/np \quad la \circ CGT \vdash np}{appauvrit \circ (la \circ CGT) \vdash np\backslash s}\,/E \quad dangereusement \vdash (np\backslash s)\backslash_1(np\backslash s)}{(appauvrit \circ (la \circ CGT)) \circ_1 dangereusement \vdash np\backslash s}\,/E}{(appauvrit \circ_1 dangereusement) \circ (la \circ CGT) \vdash np\backslash s}\,MC$$

Similarly, we can translate proofs which use the $/\Diamond_1\Box_1 I$ rule of Table 1 into proofs using a combination of $\Diamond E$, $\Box E$, $/I$ and the two extraction rules $MA\Diamond_1$ and $MC\Diamond_1$ as shown below.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{elle}{np}\,Lex \quad
\dfrac{
\dfrac{est}{(np\backslash s_{main})/(n\backslash n)}\,Lex \quad
\dfrac{
\dfrac{responsable}{(n\backslash n)/pp_{de}}\,Lex \quad
\dfrac{\overline{x \vdash \Box_1 pp_{de}}\,Hyp}{\langle x\rangle^1 \vdash pp_{de}}\,\Box E
}{responsable \circ \langle x\rangle^1 \vdash n\backslash n}\,/E
}{est \circ (responsable \circ \langle x\rangle^1) \vdash np\backslash s_{main}}\,/E
}{
\dfrac{
\dfrac{
\dfrac{elle \circ (est \circ (responsable \circ \langle x\rangle^1)) \vdash s_{main}}{elle \circ ((est \circ responsable) \circ \langle x\rangle^1) \vdash s_{main}}\,MA\Diamond_1
}{(elle \circ (est \circ responsable)) \circ \langle x\rangle^1 \vdash s_{main}}\,MA\Diamond_1
}
}\,\backslash E
}{
\dfrac{\overline{y \vdash \Diamond_1\Box_1 pp_{de}}\,Hyp \quad (elle \circ (est \circ responsable)) \circ y \vdash s_{main}}{\dfrac{elle \circ (est \circ responsable) \vdash s_{main}/\Diamond_1\Box_1 pp_{de}}{}}\,\Diamond E
}{}\,/I
$$

Given that the goal formula is $s_{main}/\Diamond_1\Box_1 pp_{de}$, we apply the introduction rule for $/$ to obtain a hypothesis y of type $\Diamond_1\Box_1 pp_{de}$, then immediately the $\Diamond E$ rule to obtain a hypothesis x of type $\Box_1 pp_{de}$. Given this hypothesis we can continue the proof using the elimination rules for $/$, $\backslash$, and $\Box$ to derive $elle \circ (est \circ (responsable \circ \langle x\rangle^1))$ of type s_{main}.

Applying the $\Diamond E$ rule immediately will cause our derivation to fail, since it simply substitutes y for $\langle x\rangle^1$ and for the correct application of the $/I$ rule, we need y to be the immediate right daughter of the tree. Our goal is therefore to move the $\langle x\rangle^1$ subterm to the top of the tree and then apply the $\Diamond E$ rule immediately followed by the $/I$ rule. This is where the structural rule of $MA\Diamond_1$ (mixed associativity) comes in: each application of the rule moves the $\langle x\rangle^1$ subterm one step closer to the top, until we can correctly complete the proof.

A.3   *Conversion to other type-logical grammars*

As already indicated when justifying the choice for multimodal categorial grammars for the treebank annotation format in Section 2, the multimodal annotation has been designed to be compatible with other modern instantiations of type-logical grammar, such as the Displacement calculus (Morrill *et al.* 2011) and first-order linear logic (Moot 2014). The phenomena discussed in Section B.1 to Section B.3 can be imported into these calculi by simply removing the $\Diamond_0\Box_0$ prefixes since they operate in an associative base logic, whereas an implementation

of the multimodal gapping solution of Hendriks (1995), on which our analysis of gapping is based, is presented by Morrill *et al.* (2011).

B   ADDITIONAL LINGUISTIC PHENOMENA

The full set of rules from Appendix A allows us to treat a number of additional linguistic phenomena. These analyses, or at least the *ideas* behind them, should be relatively unsurprising to people familiar with linguistic analysis in the tradition of the Lambek calculus and its extensions (Moortgat 2011).

B.1   *Right-node raising*

Right-node raising (and its rare variant left-node raising) are instances of the structural rule of associativity, as is already implicit in the discussion of the examples by Lambek (1958). We need it to analyse sentences such as the following.

(8)   ses bons et   ses mauvais moments
        its good and its bad   moments

(9)   peut et   parfois   doit accompagner ...
        can   and sometimes must accompany   ...

In example (8), we want to analyse both "ses bons" and "ses mauvais" (a determiner and an adjective, which we would like to assign the formulas np/n and n/n respectively) as $np/\Diamond_0\Box_0 n$ (the reader can verify that we cannot derive $np/n \circ n/n \vdash np/n$ since associativity is not globally available). Similarly, "peut" and "parfois doit" in example (9) should be analysed as $(np\backslash s)/\Diamond_0\Box_0(np\backslash s_{inf})$. We can obtain the desired derivations for example (8) by assigning "et" the type $((np/\Diamond_0\Box_0 n)\backslash(np/n))/(np/\Diamond_0\Box_0 n)$ and combining it with the following derivation for "ses bons" (the derivation for "ses mauvais" is similar).

$$
\cfrac{
  \cfrac{x \vdash \Diamond_0\Box_0 n}{} \; Hyp
  \qquad
  \cfrac{
    \cfrac{
      \cfrac{ses}{np/n} \; Lex
      \qquad
      \cfrac{
        \cfrac{bons}{n/n} \; Lex
        \qquad
        \cfrac{\cfrac{y \vdash \Box n}{} \; Hyp}{\langle y \rangle^0 \vdash n} \; \Box E
      }{bons \circ \langle y \rangle^0 \vdash n} \; /E
    }{
      \cfrac{ses \circ (bons \circ \langle y \rangle^0) \vdash np}{(ses \circ bons) \circ \langle y \rangle^0 \vdash np} \; MA_r \Diamond_0
    }
  }{(ses \circ bons) \circ x \vdash np} \; \Diamond E_2
}{ses \circ bons \vdash np/\Diamond_0\Box_0 n} \; /I_1
$$

Left-node raising

Very rarely, for a total of nine times in the entire corpus, we need the symmetric rule of left-node raising. In the example below, we have a conjunction of two combinations of two noun post-modifiers $n\backslash n$: "français Aérospatiale" and "italien Alenia".

(10)   ... des   groupes français Aérospatiale et   italien Alenia ...
          of the groups French Aérospatiale and Italian Alenia
       'of the French group Aérospatiale and Italian (group) Alenia'

By analysing "et" (and) as $((\Diamond_0\Box_0 n\backslash n)\backslash(n\backslash n))/(\Diamond_0\Box_0 n\backslash n)$ we can use the derivability of $n\backslash n, n\backslash n \vdash \Diamond_0\Box_0 n\backslash n$ (which is derivable given the structural rule $MA_l\Diamond_0$ of Table 4) as follows.

$$
\cfrac{
\cfrac{x \vdash \Diamond_0\Box_0 n}{\textit{Hyp}} \quad
\cfrac{
\cfrac{
\cfrac{\cfrac{\cfrac{y \vdash \Box_0 n}{\textit{Hyp}}}{\langle y\rangle^0 \vdash n}\ \Box E \quad \cfrac{\cfrac{\textit{italien}}{n\backslash n}\ \textit{Lex}}{}}{\langle y\rangle^0 \circ \textit{italien} \vdash n}\ \backslash E \quad \cfrac{\cfrac{\textit{Alenia}}{n\backslash n}\ L}{}
}{(\langle y\rangle^0 \circ \textit{italien}) \circ \textit{Alenia} \vdash n}\ \backslash E
}{\langle y\rangle^0 \circ (\textit{italien} \circ \textit{Alenia}) \vdash n}\ MA_l\Diamond_0
}{x \circ (\textit{italien} \circ \textit{Alenia}) \vdash n}\ \Diamond E_2
}{\textit{italien} \circ \textit{Alenia} \vdash \Diamond_0\Box_0 n\backslash n}\ \backslash I_1
$$

Coordination of multiple arguments

The product rules $\bullet E$ and $\bullet I$ are used for coordination of multiple arguments (as shown in sentence (11) below, where the two verb arguments np and pp are conjoined, see Section 2.4 of Morrill 2011).

(11)   augmenter [$_{np}$ ses fonds propres] [$_{pp}$ de 90 millions de
       increase   [$_{np}$ its equity   ] [$_{pp}$ by 90 million
       francs] et   [$_{np}$ les quasi-fonds propres] [$_{pp}$ de 30
       francs] and [$_{np}$ its quasi-equity   ] [$_{pp}$ by 30
       millions]
       million]

We can derive these cases by assigning "et" the following formula.

$$((np \bullet pp)\backslash(np \bullet \Diamond_0\Box_0 pp))/(np \bullet pp)$$

Since we can form the $np \bullet pp$ arguments from both combinations of an np and a pp using the $\bullet I$ rule, we can derive "ses fonds propres

de ... et les quasi-fonds propres de ...” (abbreviated as e in the proof below) as being of type $np \bullet \Diamond_0 \Box_0 pp$ using an application of the $/E$ rule followed by an application of the $\backslash E$ rule. We can then combine this $np \bullet \Diamond_0 \Box_0 pp$ constituent with the verb “augmenter” (abbreviated as a) as follows.

$$
\cfrac{
  e \vdash np \bullet \Diamond_0 \Box_0 pp \qquad
  \cfrac{
    \cfrac{y \vdash \Diamond_0 \Box_0 pp}{} \; Hyp \qquad
    \cfrac{
      \cfrac{
        \cfrac{a}{((np\backslash s)/pp)/np} \; Lex \quad \cfrac{x \vdash np}{} \; Hyp
      }{a \circ x \vdash (np\backslash s)/pp} \qquad
      \cfrac{\cfrac{z \vdash \Box_0 pp}{} \; Hyp}{\langle z \rangle^0 \vdash pp} \; \Box E
    }{
      \cfrac{
        \cfrac{
          \cfrac{(a \circ x) \circ \langle z \rangle^0 \vdash np\backslash s}{a \circ (x \circ \langle z \rangle^0) \vdash np\backslash s} \; MA_r \Diamond_0
        }{a \circ (x \circ y) \vdash np\backslash s} \; \Diamond E
      }{} \; /E
    }
  }{a \circ e \vdash np\backslash s} \; \bullet E
}{}
$$

The $\Diamond_0 \Box_0 pp$ formula allows us to use the right-node raising rule of Section B.1. The proof would be slightly simpler if we assigned the word “augmenter” the formula $(np\backslash s)/(np \bullet pp)$ instead (such an analysis can also be found on page 19 of Morrill 2011). However, since we have already found independent motivation for the right-node raising rules, we have chosen to give the verb the more classical analysis of $((np\backslash s)/pp)/np$.

B.4   *Gapping*

The extraction/infixation rules are used for the analysis of gapping, as shown in sentence (12) below, where the transitive verb “atteindre” is absent from the second clause.

(12)   Le salaire horaire   atteint dorénavant   34,06 francs et
        the wages per hour reach   from now on 34.06 francs and
        le   SMIC mensuel brut   [$_{tv}$] 5756,14 francs.
        the gross minimum monthly wage [$_{tv}$] 5756.14 francs.
        ‘Hourly wages now reach 34.06 francs and the monthly minimum wage 5756.14 francs.’

We use the multimodal approach first proposed by Hendriks (1995) and then advanced by Moortgat (1996). Schematically, the formulas for gapping are of the following form

$$((s/_2 \Box_2 X)\backslash_l (s/_2 X))/(s/\Diamond_1 \Box_1 X)$$

with X being a formula for a verb, for example $X = (np\backslash s)/np$ for a

transitive verb.[10] This formula indicates that first a sentence missing a transitive verb to its right is selected (this is the extraction scheme we have seen before, though no longer restricted to right branches), then a sentence missing a transitive verb to its left, but keeping track of the *position* of this missing transitive verb in the sentence – this is implemented using the l and r modes which indicate whether the extracted verb is on the left or on the right of the current node. Finally, we *insert* a transitive verb at the position of this missing transitive verb on the left.

Even though this may seem like a rather roundabout way of achieving the desired sentence – first moving the transitive verb out, then moving it back into its original place – it has the important advantage of allowing us to get the semantics right; we know the verb from the first sentence and can therefore use it in the semantics, whereas a simpler type such as $(s\backslash s)/(s/\Diamond_1\Box_1 X)$ would not allow us to obtain the correct semantics.

In addition, abstracting away from the mode information and the unary connectives, the current analysis is an instantiation of the universal coordination formula $(Y\backslash Y)/Y$ when we choose $Y = s/X$, giving $((s/X)\backslash(s/X))/(s/X)$.

The extraction part of the gapping proof proceeds as shown below; s abbreviates "le salaire horaire" and f abbreviates "34,06 francs".

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{z \vdash \Box_2((np\backslash s)/np)}{\langle z\rangle^2 \vdash (np\backslash s)/np}\ \Box E \quad f \vdash np}{\langle z\rangle^2 \circ f \vdash np\backslash s}\ /E}{s \vdash np \qquad \qquad \qquad} \quad \cfrac{}{s \circ (\langle z\rangle^2 \circ f) \vdash s}\ \backslash E}{s \circ (\langle f\rangle^l \circ_2 z) \vdash s}\ I_{2l}}{\cfrac{(s \circ_r \langle f\rangle^l) \circ_2 z \vdash s}{s \circ_r \langle f\rangle^l \vdash s/_2\Box_2((np\backslash s)/np)}\ /I_1}\ MA_{2r}
$$

We move the hypothetical $\Box_2((np\backslash s)/np)$ out, but keep track of where

---

[10] More precisely, the instantiation of the schema we need is

$$((s/_2\Box_2((np\backslash s)/np))\backslash_l(s/_2((np\backslash s)/\Diamond_0\Box_0 np)))/(s/\Diamond_1\Box_1((np\backslash s)/np))$$

with the $\Diamond_0\Box_0 np$ permitting right-node raising (associativity) as we have seen it in Section B.1.

we have used it: from the bottom, we started left of f (leaving l as a unary branch there), then right (r).

Consequently, to get back from the top, we first go right (r) and finally left (l), ending up between s and f as required: we can then insert "atteint" of type $(np\backslash s)/np$, removing the trail of l and r during the process, as follows.[11]

$$\frac{\dfrac{(s \circ_r \langle f \rangle^l) \circ_l (et \ldots) \vdash s/_2((np\backslash s)/np) \quad a \vdash (np\backslash s)/np}{\dfrac{((s \circ_r \langle f \rangle^l) \circ_l (et \ldots)) \circ_2 a \vdash s}{\dfrac{((s \circ_r \langle f \rangle^l) \circ_2 a) \circ (et \ldots) \vdash s}{\dfrac{(s \circ (\langle f \rangle^l \circ_2 a)) \circ (et \ldots) \vdash s}{s \circ (\langle a \rangle^2 \circ f) \circ (et \ldots) \vdash s} I_{2l}^{-1}} MA_{2r}^{-1}} MC_{2l}^{-1}}}{} /E$$

B.5   *Quoted speech*

We need some special rules to treat past-perfect quoted speech, as shown in sentence (14) below. The parenthesized sentence is argument of the past participle "ajouté" and, in addition, this argument is discontinuous.

(13)   [$_s$ L'indice   composite (...) a   baissé   de 0,3% en
            [$_s$ the index composite   has descended   0.3% in
            novembre]¸ a   annoncé   mardi   31 décembre le
            November]¸ has announced Tuesday 31 December the
            département du commerce.
            Department of Commerce.
            'The composite index fell 0.3% in November, announced the
            Department of Commerce on Tuesday December 31st.'

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

The solution is essentially to analyse the entire verb group missing the s argument "a ajouté np" as $s_{main}\backslash_3 s_{main}$, the structural rules the

---

[11] The -1 as superscript to the rule names, e.g. in I_{2l}^{-1}, indicates that we apply the structural rules from *in situ binding* section of Table 4 in the "inverse" sense, i.e. bottom-up.

allow this entire group to move to the required position in the final string.

To illustrate this basic idea, we show how the structural rules for quoted speech allow us to derive "a ajouté *np*" (for some *np*) as $s_{main} \backslash_3 s_{main}$.

$$
\cfrac{
  \cfrac{a}{(s/np)/(np\backslash s_{ppart})}\ Lex \quad
  \cfrac{
    \cfrac{x \vdash s}{}\ Hyp \quad \cfrac{\cfrac{ajouté}{s\backslash_3(np\backslash s_{ppart})}\ Lex}{}
  }{
    \cfrac{x \circ_3 ajouté \vdash np\backslash s_{ppart}}{a \circ (x \circ_3 ajouté) \vdash s/np}\ /E
  }\ \backslash E
  \quad \cfrac{np \vdash np}{}
}{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{(a \circ (x \circ_3 ajouté)) \circ np \vdash s}{(x \circ_3 (a \circ ajouté)) \circ np \vdash s}\ MC_3
      }{x \circ_3 ((a \circ ajouté) \circ np) \vdash s}\ MA_3
    }{(a \circ ajouté) \circ np \vdash s\backslash_3 s}\ \backslash I
  }{}
}\ /E
$$

REFERENCES

Anne Abeillé, Lionel Clément, and Alexandra Kinyon (2000), Building a treebank for French, in *Proceedings of the Second International Language Resources and Evaluation Conference*, pp. 87–94, Athens.

Abhishek Arun and Frank Keller (2005), Lexicalization in crosslinguistic probabilistic parsing: the case of French, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 306–313, Ann Arbor, Michigan.

Srinivas Bangalore and Aravind Joshi (2011), *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, MIT Press, Cambridge, Massachusetts.

Raffaella Bernardi and Richard Moot (2003), Generalized quantifiers in declarative and interrogative sentences, *Logic Journal of the IGPL*, 11(4):419–434.

André Bittar (2010), *Building a TimeBank for French: A Reference Corpus Annotated According to the ISO-TimeML Standard*, Ph.D. thesis, Université Paris Diderot.

Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier (2004), Wide-coverage semantic representation from a CCG parser, in *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, pp. 1240–1246, Geneva.

Johan Bos, James R. Curran, and Edoardo Guzzetti (2007), The Pronto QA system at TREC-2007: harvesting hyponyms, using nominalisation patterns, and computing answer cardinality, in E. M. Voorhees and L. P. Buckland,

editors, *The Sixteenth Text REtrieval Conference, TREC 2007*, pp. 726–732, Gaitersburg, Maryland.

Johan Bos and Katja Markert (2005), Recognising textual entailment with logical inference, in *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005)*, pp. 628–635.

Wojciech Buszkowski and Gerald Penn (1990), Categorial grammars determined from linguistic data by unification, *Studia Logica*, 49:431–454.

Marie Candito, Benoît Crabbé, Pascal Denis, and François Guérin (2009), Analyse syntaxique du français : des constituants aux dépendances, in *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Senlis.

Marie Candito and Djamé Seddah (2012), Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical, in *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, pp. 321–334, Grenoble.

Bob Carpenter (1991), Categorial grammars, lexical rules and the English predicative, in Robert Levine, editor, *Formal Grammar: Theory and Practice*, number 2 in Vancouver Studies in Cognitive Science, pp. 168–242, University of British Columbia Press, Vancouver.

Stephen Clark and James R. Curran (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)*, pp. 104–111, Barcelona.

Stephen Clark and James R. Curran (2007), Wide-coverage efficient statistical parsing with CCG and log-linear models, *Computational Linguistics*, 33(4):493–552.

Matthieu Constant, Isabelle Tellier, Denys Duchier, Yoann Dupont, Anthony Sigogne, and Sylvie Billot (2011), Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français, in *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Montpellier.

Ane Dybro-Johansen (2004), *Extraction Automatique de Grammaires à Partir d'un Corpus Français*, Master's thesis, Université Paris 7.

Bruno Guillaume and Guy Perrier (2012), Semantic annotation of the French Treebank with modular graph rewriting, in *Proceedings of the Proceedings of META-RESEARCH Workshop on Advanced Treebanking (LREC'12)*, pp. 14–21, Istanbul.

Petra Hendriks (1995), Ellipsis and multimodal categorial type logic, in Glyn Morrill and Richard T. Oehrle, editors, *Proceedings of Formal Grammar 1995*, pp. 107–122, Barcelona.

Julia Hockenmaier and Mark Steedman (2005), CCGbank: users's manual, Technical report, Department of Computer and Information Science, University of Pennsylvania.

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

Joachim LAMBEK (1958), The mathematics of sentence structure, *American Mathematical Monthly*, 65:154–170.

Anaïs LEFEUVRE, Richard MOOT, Christian RETORÉ, and Noémie-Fleur SANDILLON-REZER (2012), Traitement automatique sur corpus de récits de voyages pyrénéens : une analyse syntaxique, sémantique et temporelle, in *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Grenoble.

Roger LEVY and Galen ANDREW (2006), Tregex and Tsurgeon: tools for querying and manipulating tree data structures, in *5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa.

David M. MAGERMAN (1994), *Natural Language Parsing as Statistical Pattern Recognition*, Ph.D. thesis, University of Pennsylvania.

Michael MOORTGAT (1996), In situ binding: a modal analysis, in Paul DEKKER and Martin STOKHOF, editors, *Proceedings 10th Amsterdam Colloquium*, pp. 539–549, ILLC, Amsterdam.

Michael MOORTGAT (2011), Categorial type logics, in Johan VAN BENTHEM and Alice TER MEULEN, editors, *Handbook of Logic and Language*, chapter 2, pp. 95–179, North-Holland Elsevier, Amsterdam.

Michael MOORTGAT and Richard MOOT (2001), CGN to Grail: extracting a type-logical lexicon from the CGN annotation, *Language and Computers*, 37(1):126–143.

Richard MOOT (2010a), Automated extraction of type-logical supertags from the Spoken Dutch Corpus, in Srinivas BANGALORE and Aravind JOSHI, editors, *Complexity of Lexical Descriptions and its Relevance to Natural Language Processing: A Supertagging Approach*, chapter 12, pp. 291–312, MIT Press, Cambridge, Massachusetts.

Richard MOOT (2010b), Semi-automated extraction of a wide-coverage type-logical grammar for French, in *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Montreal.

Richard MOOT (2012), Wide-coverage semantics for spatio-temporal reasoning, *Traitement Automatique des Languages*, 53(2):115–142.

Richard MOOT (2014), Extended Lambek calculi and first-order linear logic, in Claudia CASADIO, Bob COECKE, Michael MOORTGAT, and Philip SCOTT, editors, *Categories and Types in Logic, Language, and Physics: Essays dedicated to Jim Lambek on the Occasion of this 90th Birthday*, number 8222 in Lecture Notes in Artificial Intelligence, pp. 297–330, Springer, Heidelberg.

Richard MOOT and Mario PIAZZA (2001), Linguistic applications of first order multiplicative linear logic, *Journal of Logic, Language and Information*, 10(2):211–232.

Richard MOOT and Christian RETORÉ (2006), Les indices pronominaux du français dans les grammaires catégorielles, *Lingvisticae Investigationes*, 29(1):137–146.

Richard MOOT and Christian RETORÉ (2012), *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*, number 6850 in Lecture Notes in Artificial Intelligence, Springer, Heidelberg.

Glyn MORRILL (1994), *Type Logical Grammar*, Kluwer Academic Publishers, Dordrecht.

Glyn MORRILL (2011), *Categorial Grammar: Logical Syntax, Semantics, and Processing*, Oxford University Press, Oxford.

Glyn MORRILL, Oriol VALENTÍN, and Mario FADDA (2011), The Displacement calculus, *Journal of Logic, Language and Information*, 20(1):1–48.

Van Tien NGUYEN (2012), *Méthode d'Extraction d'Informations Géographiques à des fins d'Enrichissement d'une Ontologie de Domaine*, Ph.D. thesis, Université de Pau et des Pays de l'Adour.

Richard T. OEHRLE (2011), Multi-modal type-logical grammar, in Robert BORSLEY and Kersti BÖRJARS, editors, *Non-transformational Syntax: Formal and Explicit Models of Grammar*, chapter 6, pp. 225–267, Wiley-Blackwell.

Benoît SAGOT (2010), The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French, in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta.

Noémie-Fleur SANDILLON-REZER (2013), *Apprentissage de Grammaires Catégorielles: Transducteurs d'Arbres et Clustering pour Induction de Grammaires Catégorielles*, Ph.D. thesis, Bordeaux University.

Natalie SCHLUTER and Josef VAN GENABITH (2008), Treebank-based acquisition of LFG parsing resources for French, in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech.

Stuart SHIEBER, Yves SCHABES, and Fernando PEREIRA (1995), Principles and implementation of deductive parsing, *Journal of Logic Programming*, 24(1–2):3–36.

Willemijn VERMAAT (2005), *The Logic of Variation. A Cross-Linguistic Account of wh-question Formation*, Ph.D. thesis, Utrecht Institute of Linguistics OTS, Utrecht University.