Lexik : a maintenance tool for FTAG

Nicolas Barrier, Sébastien Barrier, Alexandra Kinyon

TALaNa – LaTTice
UFRL, University Paris 7
2, pl. Jussieu F-75251 Paris Cedex 05
{nbarrier, sbarrier, kinyon}@linguist.jussieu.fr

Abstract

In this paper we present LEXIK, a tool which allows to maintain and gather data on wide coverage grammars based on the XTAG format. We present the tool, show how it is used within the FTAG project (Abeillé & al. 2000a), and compare it to similar work done on the Xtag grammar for English (Sarkar & Wintner 99).

Introduction

Over the past ten years, FTAG, a wide coverage LTAG has been developed at Talana, building up on the work of (Abeillé, 91). Thanks to the MetaGrammar developed by (Candito 96,99), which allows to generate semi-automatically an LTAG, the number of trees has augmented drastically: from 650 trees for the manually written grammar, we have now reached 5000 elementary trees (cf. Abeillé & al 99,00). Although this has improved the coverage of the grammar, new maintenance issues have appeared :
To remedy this problem, we have developed a tool which we call Lexik. The goal is twofold :

- Insuring consistency in the grammar
- Easily extracting information on a large scale

In the first part of this paper, we review the main characteristic of FTAG and present the problems encountered for maintaining and updating the Grammar. In a second part, we present our tool, as well as its utility. Especially, we compare it to the work presented in (Sarkar & Wintner 99). Finally, we show how this tool is used in other projects.

1. Main characteristics of FTAG

We assume some familiarity with the LTAG formalism. We recall that elementary units of an LTAG are lexicalized constituent trees, which encode all the surface constructions available for a given language. Within FTAG, elementary trees respect the following linguistic well-formedness principles: (Kroch and Joshi 1985, Abeillé 1991, Frank 1992) :

- Strict Lexicalization : all elementary trees are anchored by at least one lexical element, the empty string cannot anchor a tree by itself,
- Surfacism: an elementary tree encodes all word order variations, all basic syntactic phenomena (passive, extraction...) and crossing of phenomena.
- Semantic Consistency : no elementary tree is semantically void (this ensures the compositionality of the syntactic analysis),
- Semantic Minimality : elementary trees correspond to no more than one semantic unit
- Predicate Argument Cooccurrence Principle : the elementary tree is the minimal syntactic structure that includes a leaf node for each realized semantic argument of the anchor(s).

Semantic minimality and consistency imply that function words appear as co-anchors (cf. Figure 1, the relevant syntactic and semantic units are donner-à (give to) and penser-que (think that)).
The elementary trees are combined by substitution or adjunction, and the features of nodes in contact must unify. They thus directly represent all the syntactic rules of the language.

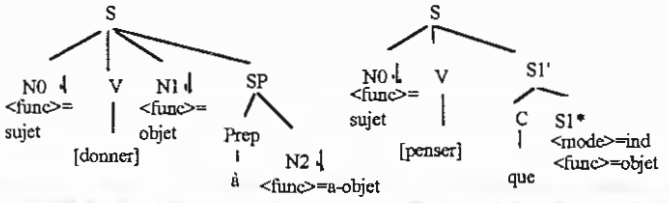


FIGURE 1 : Elementary trees with functional coanchors

1.1. Factorization of Lexicalized Elementary trees

Strict lexicalization at execution time does not prevent from internally compacting the common parts of the elementary trees. This compacting is required for any reasonably sized grammar, since for instance a verbal form may anchor dozens or hundreds of elementary trees. In practice, lexicalized elementary trees are compiled out of three sources of information:

- a set of tree sketches ("pre-lexicalized" structures, whose lexical anchor(s) is not instantiated)
- a syntactic lexicon, where each lemma is associated with the relevant tree sketches
- a morphological lexicon, where inflected forms point to a lemma associated to morphological features

Lexical selection of tree sketches is controlled by features from the syntactic and morphological lexicons, and uses the notion of tree families, grouping sets of tree sketches that share the same (initial) subcategorization frame. The tree sketches of a family show all possible surface realizations of arguments (pronominal clitic realization, extraction, inversion...) as well as all possible transitivity alternations (impersonal, passive, middle..).

A lemma selects one or several families (corresponding to one or several initial subcat frames) and with the help of features selects exactly the relevant tree sketches.

Figure 3 shows the canonical elementary tree anchored by *parlait* (talked)[1] and Figure 2 the three sources of information associated with its internal representation.

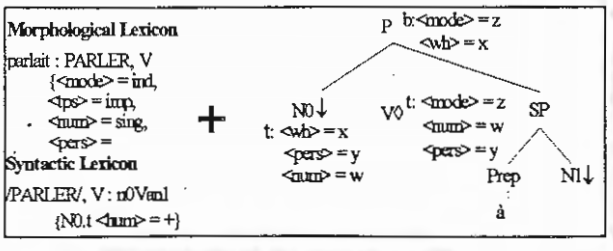


FIGURE 2 : 3 sources of information

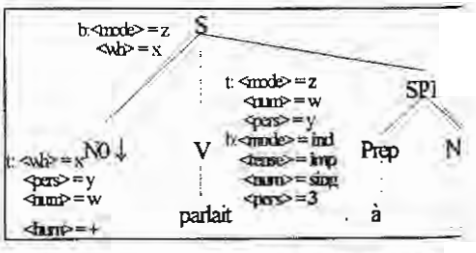


FIGURE 3 : Corresponding elementary trees

[1] Information coming from the lexicon appears in bold characters.

The inflected form *parlait* points to the lemma PARLER, and the lexeme /PARLER/ selects in turn the n0Vanl family, where the preposition appears as a co-anchor (except when argument 1 is cliticized).

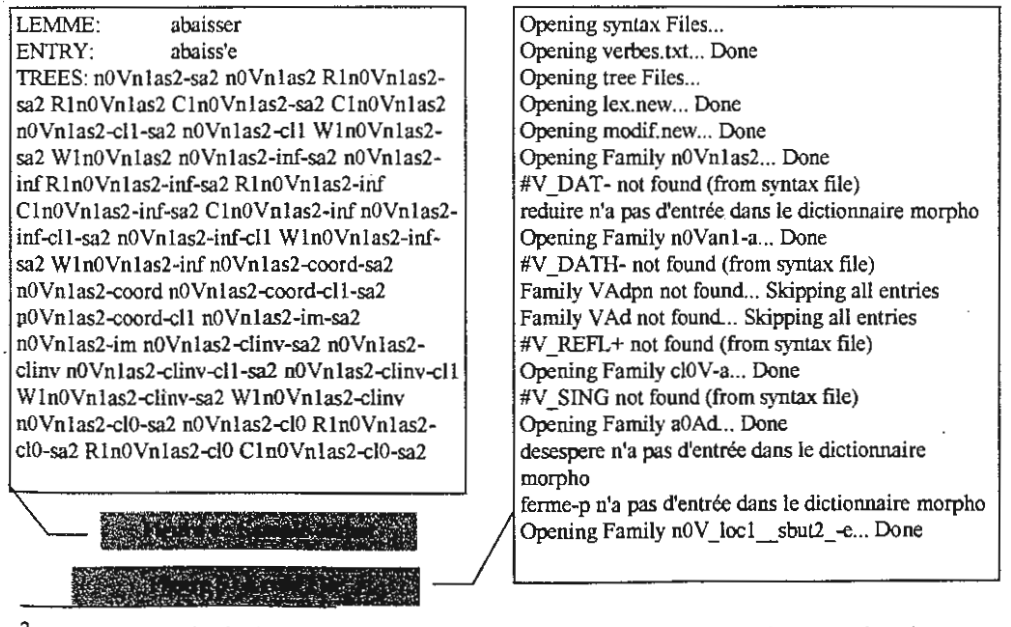
Currently, our morphological lexicon comprises 50000 inflected for ms, our syntactic lexicon has more than 6000 entries, and the bulk of the grammar consist in 5280 tree sketches. Concretely, each family is a file were a set of trees is stored.

Maintaining and updating such a large database is difficult : for example, one can generate a large grammar using Candito's tool, but integrate it with manually written tree sketches for idioms (since trees for idioms are not automatically generated). Then one needs to make sure that the features used in those 2 parts of the grammar are identical. Also, while the automatic generation of the grammar insures consistency (i.e. all features are generated automatically from a hand written hierarchy), errors may still propagate in the grammar, but on a larger scale : if a feature has a typo in the hand written hierarchy (ex : aggreement instead of agreement), then this error will be propagated in hundreds of trees when the grammar is generated (with dramatic effects if it remains undetected). Also, consistency between the grammar and the lexicons is an important issue : for example one would like to detect lexical items which refer to trees that do not exist in the grammar (either because of an error or of an update).

Also, we just said that that a verb can anchor several dozens of trees, but we would like to have a more precise measure of this, and be able to answer questions such as : how many trees does verb X anchor ? How many trees on average are anchored for French verbs ?
This is were Lexik comes in.

2. Lexik : presentation of the tool

Lexik allows to lexicalize tree sketches, that is it takes the morphological lexicon, the syntactic lexicon and the tree sketches as input (e.g. figure 2) and outputs on the one hand fully lexicalized trees (figure 3) anchored by each inflected form[2], and on the other hand, if necessary, an error file. A sample output file can be seen on figure 4, a sample error file can be seen on figure 5.



[2] This is done at runtime by the Xtag parser, but in an opaque manner, which prevents error detection and repair

2.1. Consistency issues

The error file outputed by Lexik allows to detect 4 types of errors :

1. Inconsistencies between the morphological and syntactic lexicons (i.e. lemma with no corresponding inflected forms and vice-versa)
2. Organization problems in the grammar (e.g. missing trees or families)
3. Feature problems (e.g. unknown features)

A simple script allows to extract the most common (and hence damaging) errors, which can then be repaired (cf figure 5)

This work on consistency can be compared with that of (Sarkar and Wintner 99), who validate the consistency of feature structures by imposing type discipline. Contrary to us, their approach focuses on features to detect the 4 following kinds of problems :

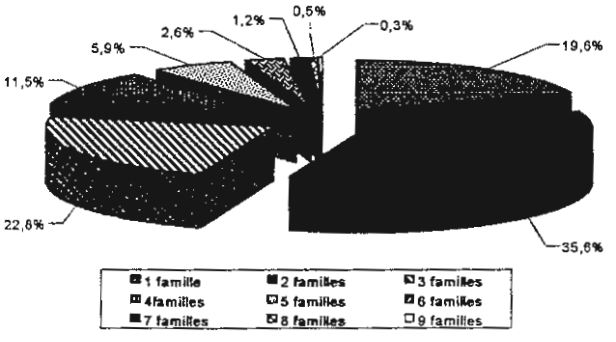
1- ambiguous features (e.g. gen : genitive or gender ?)
2- typos : relpro instead of rel-pro
3- Undocumented features (i.e. used in previous versions of the grammar)
4- type errors : e.g. assign-case is relevant only for verbs, not for nouns

Their tool runs on a wide-coverage LTAG for English (cf Xtag group 95), while ours runs on FTAG for French (cf Abeille & al 99). Since the 2 grammars resort to similar formats, it would be interesting to couple the 2 approaches.

2.2. Gathering information

In addition of detecting errors in the grammar, Lexik allows to gather information that was unavailable previously.

FIGURE 6 : Repartition of verbal lemmas by number of families anchored



Up to now, we could only gather data at the level of families. This allowed to know for instance that the two tree families n0Vn1 (transitive) and n0Vn1-à-n2 (ditransitive) are anchored by two thirds of lemmas (cf NBarrier 99). To have a clearer idea, we extracted 1060 inflected forms of verbs from the 1 million word corpus LeMonde (cf Abeillé & Clément 99) and found that verbs anchor on average 2.8 families / verb (Figure 6), whereas other parts of speech (i.e. nouns, adverbs, adjectives) only anchor between 1 & 2 trees. Only 7 of these verbs anchor 8 families or more[3] (cf SBarrier 99) and only 2 out of these 7 verbs are among the most 100 frequent ones (être *(be)* most often used as an auxiliary, and parler *(talk)*). Intuitively, one could expect that verbs anchoring the more families will also be anchoring the

[3] These verbs are : amuser (amuse), être (be), parler(talk), répendre(spill), revenir (come back), heurter (bump into), dresser (put up)

more trees, and conversely that verbs anchoring the more trees will be verbs anchoring the more families, despite the fact that some verbs anchor only some of the trees contained in a family[4].

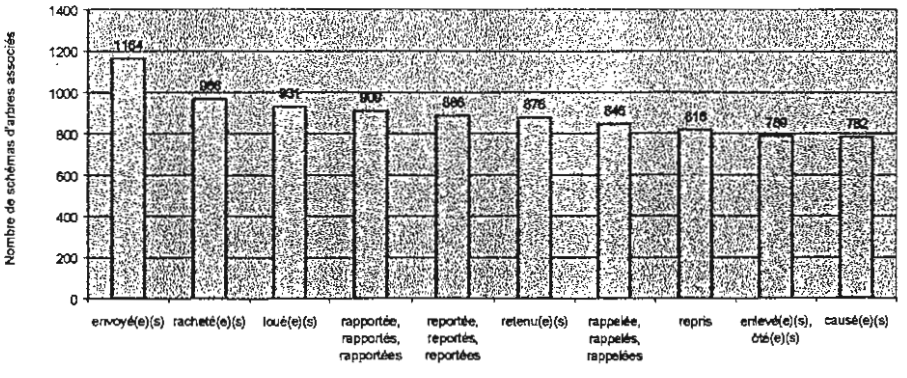
But by going down to the level of trees, Lexik allows to show that this is not the case : it turns out that the inflected form anchoring the more trees (1164) is "envoyés" (past participle for the verb "envoyer"/send) whereas it selects only 3 families. More generally, we have reached the conclusion that the number of families anchored by a given lexical item does not indicate how many trees this item will anchor. Figure 7 illustrates this phenomenon for a few common verbs. We also found that the morphological properties of the item (e.g. past-participles ...) are actually important to predict how many trees an item can anchor.

Lemme	Nombre de familles associées	Forme fléchie retenue	Nombre de schémas d'arbres associés
Amuser	9	Amusés	112
Parler	9	Parlés	333
Répondre	8	Répondus	595
Revenir	8	Revenus	210
Rendre	7	Rendus	452
Parier	6	Parier	93
Louer	5	Loués	931
Envoyer	3	Envoyés	1164
Visiter	1	Visités	78

FIGURE 7 : families and trees anchored by a few common inflected forms

On average, each of the 1060 inflected verbs from LeMonde anchors 139.17 tree schemata (ranging from 1 to 1164). Figure 8 shows the inflected forms which anchor the most trees. It is noticable that all the examples on Figure 8 are past-participles: for exemple for "envoyer" the past-participle anchors 1164 trees, but other inflected forms of this verb (e.g. "envoyons" : Present 1st person plural) anchors only 596 trees. Similarly, if we examine the 2nd most ambigous form (racheté(es) / rebuy), it anchors 966 trees. But "rachetez", which is the 2nd person plural for the same verb in the present, anchors only 498 trees.

FIGURE 8 : Number of trees anchored by the most ambiguous inflected forms



[4] E.g. couter (cost) is a transitive verb which does not passivize, hence it will select all elementary trees in the transitive family, excluding trees for passive.

We also ran Lexik on partial data : we used the same 1060 inflected verbs but kept in the grammar only one tree family n0vn1 for transitive verbs. This family consists in 78 trees. We than ranked the 1060 forms by the number of trees they anchor. It turned out that classes of items bearing morphological similarities appeared : past-participles were at the top of the list (anchoring all 78 trees), followed by infinitivals (anchoring approximately 46 of these trees) and by past participles (anchoring roughly 12 of these trees).

Conclusion

We have presented Lexik : a tool which allows to detect inconsistencies in a wide coverage LTAG for French, and which allows to extract information on a large scale.

It is a first step towards online disambiguation, similarly to what was done for English in (Srinivas & al 94), by allowing to refine a first-pass strategy during parsing (cf Kinyon 99a), and by coupling it with a parse-ranker for TAGs (cf Kinyon 99b,c)

Also, Lexik is being extended to serve as a front end to a function annotation tool, in order to create a large treebank for French (cf. Abeillé & al 00b).

It is also used as the front end of a rule-based supertagger for French, and to collect data in order to build a psycholinguistically relevant processing model for TAGs (cf Kinyon 99d,00)

References

Abeillé A., 1991. Une grammaire lexicalisée d'arbres adjoints pour le français. PhD Thesis, University Paris 7.

Abeillé A. Clément L., 1999. A reference tagged corpus for French. Proc. LINC99, EACL. Bergen.

Abeillé A., Candito M.H., Kinyon A. 2000a : Current status of FTAG. Proc. TAG+5. Paris.

Abeillé A., Clément L. Kinyon A., 2000b : Building a treebank for French. Proc. LREC'2000. Athens

Abeillé A., Candito M.H., Kinyon A. : FTAG : current status and parsing scheme. Proc. Vextal'99. Venice.

Barrier S. 1999. Classification et repérage des valences verbales en français : experiences avec FTAG. MS thesis. Univ. Paris 7.

Barrier N. 1999. Lexik : un outil de lexicalisation des TAGs. Application à la désambiguation syntaxique du français MS thesis. Univ. Paris 7.

M-H Candito, 1996. A principle-based hierarchical representation of LTAG, Proc. 15th COLING. Kopenhagen.

M.-H. Candito, 1999. Représentation modulaire et paramétrable de grammaires électroniques lexicalisées. Application au français et à l'italien. PhD dissertation. University Paris 7.

M-H. Candito, S. Kahane, 1998. Can the TAG derivation tree represent a semantic graph ? an answer in the light of MTT, Proceedings TAG+4 Workshop, Philadelphia.

Frank R. 1992 Syntactic Locality and Tree Adjoining Grammar : Grammatical Acquisition and Processing Perspectives. PhD dissertation. University of Pennsylvania.

Kinyon A. 1999a : Distinction entre Regard en avant et Première Passe pour l'analyse des LTAGs. Proc. Recital'99. Cargèse.

Kinyon A. 1999b : Parsing preferences and lexicalized Tree Adjoining Grammars : exploiting the derivation tree, Proc. ACL'99. Maryland.

Kinyon A. 1999c : Hiérarchisation d'analyses basée sur des informations dependancielles pour les LTAGs, Proc. TALN'99. Cargèse.

Kinyon A. 1999d : Some remarks about the psycholinguistic relevance of LTAGs. Proc. CLIN'99. Utrecht.

Kinyon A. 2000: Towards a psycholinguistically relevant processing model for LTAGs. Proc. Cogsci'2000. Philadelphia.

Kroch A., Joshi, A. 1985. The linguistic relevance of Tree Adjoining grammars, Technical Report, Univ. of Pennsylvania.

Sarkar A., Wintner S. 1999 : Typing as a means for validating feature structures. Proc. CLIN'99. Utrecht.

Srinivas B., Doran C., Kulick S. 1995 : Heuristics and Parse Ranking. Proc. IWPT'95. Prag. Czech Republic.

Xtag group 1995 A LTAG for English. Technical Report IRCS 95-03. University of Pennsylvania.