FTAG : developping and maintaining a wide-coverage grammar for French

Anne Abeillé, Marie-Hélène Candito, Alexandra Kinyon
TALANA, University Paris 7
2 Pl Jussieu, case 7003
75005 Paris France
{abeille,candito,kinyon}@linguist.jussieu.fr

Introduction

We describe the current status and organization of a French Lexicalized Tree Adjoining Grammar (FTAG), developed over the last 10 years at Paris 7 (Abeillé 91, Abeillé & al. 99). The new version of the grammar is generated semi-automatically, independently of any corpus or application domain. It is intended to model speakers' competence, and can be used both for parsing and generation. As far as parsing is concerned, we describe a general processing module which can rank the different parses produced based on linguistic information present in LTAG. We also present Lexik : a maintenance tool for wide coverage TAGs basec on the Xtag format (cf Barrier & al 00) and compare it with the work of (Sarkar & Wintner 99). We assume the reader is familiar with LTAGs and refer to (Joshi 87) for an introduction.

1. General linguistic choices

FTAG is based on two linguistic approaches: that of LADL's lexicon grammar (for most lexical encoding) (cf Gross 75, Leclere 89) and that of HPSG (for most choices of syntactic structures). Most of our linguistic analyses follow those of (Abeillé 1991)

1.1 A minimal tagset

We depart from traditional part of speech wherever the modern linguistic analyses have better to propose, especially in the generative tradition (Clitics for weak pronouns, and Complementizers). We collapse proper names, common nouns and pronouns into one category N, with features. We do not have a tag for subordinating conjunctions which are either Prepositions or Complementizers. Sentential structures are 'flat' (no internal VP). We thus use the following tagset.

- Lexical categories: D (determiners), N (nouns, names, pronouns), V (verb), Cl (clitic pronoun), Prep (preposition), A (adjective), Adv (adverb), Conj (Coordinating conjunction), C (complementizer, subordinating conjunction),
- Non lexical categories: SP (prepositional phrase), S (sentence). A and N are also used for nominal or adjectival phrases.

1.2 A rich set of grammatical functions

We added grammatical functions (as features) in order to express 'deep' subcategorization (as it is used in the partial description of trees in the metagrammar, see below), but also in order to constrain the distribution of pronominal NPs or quantifiers:

** Moi viendrai * Paul voit tous (*Me will come / *P. sees all)*

Paul pense à moi / à tous. (P. thinks about me / about all)

Strong pronoun such as *moi (me),* or the quantifier *tous (all)* cannot be surface subjects or direct objects, but can be prepositional complements. The functions used in the FTAG for verbs are the following:

- subject
- object

- dative-object
- oblic-object
- genitive-objet
- locative
- source-locative
- manner
- goal-infinitive
- perception-infinitive
- interrogative clause
- predicative complement
- agent-object
- modifier.

1.3. A parsimonious use of features

Most of the syntactic properties handled by feature structures in unification based linguistic theories (LFG or HPSG) are directly captured by the topology of the elementary trees in FTAG. No use has to be made of valence or slash features to ensure subcategorization requirements or filler-gap relations. No feature passing principles, besides unification, are needed either.

We only rely on atomic valued features (which guarantees against any cyclic structure). We distinguish between:

- **Morphological features**, which are used in the morphological lexicon, in the syntactic lexicon when an argument is constrained for them (eg *trouver-find* has only indicative sentential complement) and for agreement in the elementary tree sketches,
- **Syntactic features**, used in the syntactic lexicon (for a verb to disallow passive for example) and in the tree sketches (to distinguish between trees in the same family or to further constrain tree combinations),
- **Semantic features** : large-grained classifications used for arguments (human, locative etc) which should be further refined.

We are currently using approximately 40 features:

morphological features: <det>, <card>, <case>, <el>, <mode>, <num>, <ord>, <pers>, <P-num>, <P-pers>, <tense>.

syntactic features: <ant>, <ant-s>, < ant-v>, < aux>, <cq>, <det>, <extrap>, <gen>, <inv>, <modif>, <neg>, <nom>, <passive>, <part-num>, < part-gen>, < pred>, <princ>, <pro>, <quant>, <san1>, <san2>, <suj-gen>, < suj-pers>, < suj-num>, < sym>, <tense>, <wh>.
semantic features: <conc>, <degre>, <hum>, <loc>, <man>.

2 The organization of FTAG

2.1 Three sources of information for lexicalized elementary trees

FTAG is organized using the XTAG format (XTAG group 1995) : lexicalized elementary trees are compiled out of three sources of information:

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

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

A lexeme selects one or several families (corresponding to one or several initial subcategorization frames) and with the help of features selects exactly the relevant tree sketches : The features may rule out some tree sketches of the selected family, either because of morphological clash (eg. the passive trees are only selected by past participles) or because of `` idiosyncrasies '' (eg. the French transitive verb *peser to weigh* disallows passive).

2.2 The lexicons

They currently comprise the following items:

- **Morphological lexicons**: over 50 000 (inflected) forms: 45800 for verbs, 3500 for nouns and pronouns, 950 for adjectives and 50 for determiners.
- **Syntactic lexicons** : over 6000 (disambiguated) entries: 3700 for verbs, 500 for prepositions and adverbs, 800 for adjectives, 80 for determiners, 2000 for nouns, 350 for idioms

The lexical items chosen have been extracted as the most frequent ones from the frequency lists of (Julliand & al. 1970) and (Catach 1984), except for idioms where we rely on personal intuitions. They have been disambiguated (and separated into different syntactic entries) with standard dictionaries as well as LADL lexicon-grammar tables (Gross 1975). The morphological lexicons have been automatically generated, using PC-Kimmo adapted to French. Both lexicons are organized in lexical databases, and the features normalized with templates.

The morphological lexicon associates forms with lemmas. The syntactic lexicon associates lemmas with constructions (elementary trees or tree families with features) and performs some meaning disambiguation (based on different syntactic constructions, for example for the French verb *abattre - knock down, shoot down*) :

INDEX: abattre/1 (physical meaning)
ENTRY: abattre
POS: V
FAM: n0Vn1
FS:

INDEX: abattre/2 (psychological meaning, possible sentential subject)
ENTRY: abattre
POS: V
FAM: s0Vn1
FS: #N1_HUM+, #N0_HUM-

2.3. The metagrammar

Crossings of linguistic phenomena (transitivity alternation, word order variations etc) typically lead to several hundreds of tree sketches. To represent in a more compact and organized way the set of tree sketches for FTAG, we use an additional layer of linguistic description, called the metagrammar (MG) (Candito 96, 99). MG imposes a general organization for syntactic information and formalizes the well-formedness conditions on elementary trees. It provides a general overview of the grammar and makes it possible for a tool to automatically generate the desired tree sketches from the combination of smaller descriptions.

While trees specify for any pair of nodes either a precedence relation or a path of parent relations, these partial descriptions of trees are sets of constraints that may leave underspecified the relation existing between two nodes. The relation between two nodes may be further specified by adding constraints in sub-classes or in lateral classes in the inheritance network. Inheritance of partial descriptions is monotonic.

In MG, nodes of partial descriptions are augmented with specific feature structures, called meta-features, constraining for instance, the possible parts of speech of a node or the index in the case of argumental nodes.

Subcategorization (hereafter subcat) of predicates is expressed as a list of syntactic functions, and their possible categories. An initial subcat is distinguished, namely the one for the unmarked case, and is modifiable by redistribution of the functions associated with the arguments of the predicate.

Structures sharing the same initial subcat are grouped in a tree family. For verbal predicates, an elementary tree is partly represented with an ordered list of successive subcats, from the initial one to the final one. MG represents this repartition of information by imposing a three-dimension inheritance network:

- Dimension 1 for initial subcategorization
- Dimension 2 for redistributions of functions (ex for full passive or impersonal) and
- Dimension 3 for surface realizations of syntactic functions (ex for extracted arguments).

The 3 dimension hierarchy is partially handwritten, the elementary trees are automatically generated with a two-step process. First the compiler automatically creates additional classes of the inheritance network : the "crossing classes". Then each crossing class is translated into one or several tree sketches. Crossing classes are automatically built as follows:

- a crossing class inherits one subcat (dimension 1)
- then, the crossing class inherits one redistribution (dimension 2)
- then, the crossing class inherits realizations for all its arguments (dimension 3).

For a crossing class to be well-formed, all unifications involved during the inheritance process must succeed, either for feature structures or for partial descriptions.

3. Elementary trees in FTAG

3.1 Linguistic principles for elementary trees

Within FTAG, elementary trees respect the following linguistic well-formedness principles: (Kroch & Joshi 85, Abeillé 91, Candito 99, Candito & Kahane 98)

- Strict Lexicalization : all elementary trees are anchored by at least one lexical element, the empty string cannot anchor a tree by itself,
- Semantic Consistency : no elementary tree is semantically void (this ensures the compositionality of the syntactic analysis),
- Semantic Minimality : no elementary tree correspond to more than one semantic unit (modulo lexicalism : lexical anchors are not broken down into morphemes).
- Predicate Argument Cooccurrence Principle (PACP): an elementary tree is the minimal syntactic structure that includes a leaf node for each realized semantic argument of the anchor(s).

Initial trees are used for arguments, verbs with non sentential arguments, auxiliary trees are used for modifiers, determiners, modals, auxiliaries and verbs with sentential complement. Some examples of elementary trees can be seen on figure 1

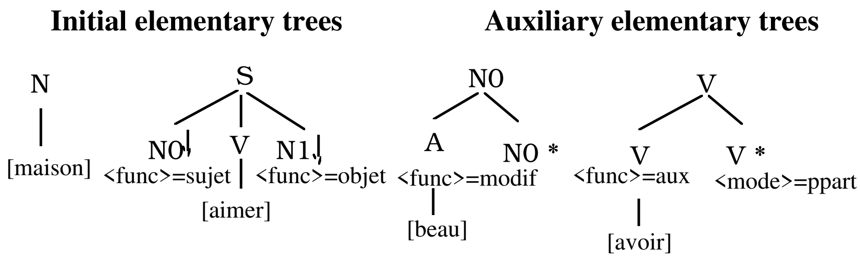

Figure 1. Elementary trees

We dispense with most empty categories, especially in the case of extraction. Semantically void (or non autonomous) elements, such as complementizers, argument marking prepositions or idiom chunks, are coanchors in the elementary tree of their governing predicates (figure 2)

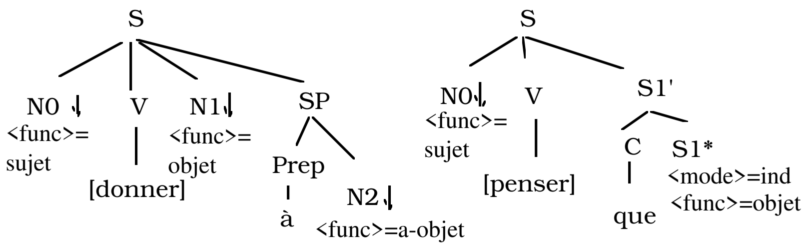

Figure 2. Elementary trees with functional co-anchors

3.2. The metagrammar for FTAG

FTAG currently comprises over 5000 elementary tree sketches (not counting trees for causative constructions) . Currently, all but 40 of them are compiled from the French metagrammar. The 40 remaining tree sketches are trees for determiners (plain and complex), nouns used as arguments, coordination conjunctions, clitics and "special" trees for deficient verbs such as raising verbs and auxiliaries.

The French MG comprises the description for the tree sketches anchored by full verbs, prepositions, adverbs, adjectives, and nouns (when used as modifiers). Within dimension 1, the French metagrammar comprises 54 initial subcat frames for verbs (which means there are 54 tree families for verbs in FTAG), 4 initial subcats for adjectives, 12 for adverbs, prepositions, subordinating conjunctions and nominal modifiers. In dimension 2, primarily relevant for verbs, we have defined as redistributions the following phenomena:

- Passive (with or without agent) : additional V-headed elementary trees with auxiliary
- Substitution of the auxiliary "être"
- Causative constructions
- Reflexive
- Impersonal constructions (active and passive)
- Middle "se" (*ces robes se lavent facilement/These dresses wash easily*)

In dimension 3 , we define as realizations the following phenomena:

- Infinitival and sentential arguments (treated as S-complements)
- Relatives (*qui, que, dont*, Prep *qui*, Prep *lequel*), indicative, subjunctive
- Interrogatives (direct, indirect, *est-ce-que*)
- Cleft sentences (*c'est que, c'est qui, c'est* Prep N *que*)
- Clitic pronouns
- Subject inversion (nominal or subject clitic)
- Unbounded dependencies (with island constraints)
- Participials (past or present part, NP modifiers), Past participle agreement.
- null realization (empty subjects for infintiives, participials or imperatives)
- factorization (for subjects in coordinated phrases)
- word order variation (among complements)

Work remains to be done of the syntax of quantifiers (often discontinuous in French), on negation (including negative concord), coordination and comparison (including superlatives).

4. Evaluating FTAG

Evaluating a wide coverage grammar is a difficult task, especially in the absence of reference tree banks for French. We performed a quantitative evaluation using the French test suite developed in the TSNLP project (Estival & Lehmann 1996). Further evaluation will be done on newspaper corpora.

4.1. Evaluation using TSNLP

We have performed an external evaluation using the TSNLP multilingual data base, which aims at covering the major syntactic phenomena for each language, using a minimal vocabulary (a few hundred words). We have extracted all the French items of the TSNLP data base, classified by grammatical status (we only took 0 and 1), by length and by phenomenon (according to TSNLP original classification). For all grammatical items, the results with the 1998 version of our grammar are as follows:

- over 80 % of the grammatical parsed, with an average of 2.9 parses per sentences
- over 82% of the non grammatical sentences have been correctly rejected.

There were no unknown words. The main failure cases are the following:

- missing lexical coding (transitive verb without object, transitive use of intransitives),
- missing elementary tree (causative trees, postverbal clitics with imperatives),
- feature unification clash (agreement with politeness forms: *vous êtes belle (You are beautiful)*, or with coordination : *deux bandes bleue et jaune- Two stripes blue and yellow*),
- missing phenomenon (tough construction, gapping...).

Cases of overanalysis either come from a disputable TSNLP encoding (for example for *concordance des temps*), or from the incompleteness of our representation (for example we overgenerate when dealing with coordination or negatio).

We can compare these results with a previous evaluation done in 1996 (Abeillé & al. 96), with a smaller coverage grammar (comprising about 830 elementary tree sketches), using the same lexicon. The results were 65% of the grammatical sentences (excluding coordination) parsed. The ambiguity rate was 1,5 parses per parsed sentence in 1996.

4.2. Comparison with other syntactic ressources

The lexicon-grammar developed at LADL for more than 20 years is an unrivaled source of knowledge reusable in the sense that it is not designed for any program and not even dependent on any special formalism. However, it cannot be directly used to analyze (or generate) a text since it only lists some basic constructions (with their lexical head). It does not code the crossing of constructions nor the productive phenomena which are not clearly lexically sensitive (such as causative, quantifier floating or argument extraction for simple verbs).

Thus, even though it is crucial to know that transitive *voler (to steal)* must be distinguished from intransitive *voler (to fly)*, more general grammatical rules are needed to know that it is the transitive *voler* which is instanciated in examples (1)-(2) without a postverbal np object, or that it is the intransitive *voler* which is instanciated in examples (3)-(4) (even though there is a postverbal np):

(1) Ils veulent tout voler *(They want to everything steal)*

(2) les bijoux qu'ils ont finalement avoué avoir volé... *(The jewelry they have finally admitted stealing)*

(3) Lufthansa fait voler ses avions 5 jours sur 7 *(L. makes its planes fly 5 days out of 7)*

(4) A une altitude à laquelle ne vole normalement aucun avion ... *(At an altitude at which flies normally no plane ...)*

M. Salkoff (1973, 1979) string grammar has listed numerous grammatical strings representative of French syntax but has never been associated with a sizable lexicon and cannot be reused independently of the parsing scheme it was made for. The GB grammar developed for French at LATL (Wehrli 1997), is more modular and associated with a sizable dictionnary. But it is not clearly separated from the program that uses it (extraction or passive phenomena are not handled as grammatical data but as types of action -attachment, trace creation... -that the program does at a certain stage in the parsing scheme) and thus cannot be reused as such for other applications.

5. Ranking parses

Our grammar is being used for text generation (Danlos 1998) and for parsing. The output of a TAG parser can be viewed as a derived tree (encoding phrase structure) or as a derivation tree (close to a dependency structure). Since it is both more compact and more informative (than the derived tree), we choose the derivation tree for the postprocessing ranking module that we developed (contrary to (Srinivas & al. 95)).

5.1. General Disambiguation principles

Our parse ranker is based on lexical and syntactic preferences. Since we work on the derivation tree which exhibits the lexicalized trees used for parsing, it is easy to mix lexical and syntactic preferences. Our parse ranker thus uses 3 types of preferences : lexical preference (such as valence preference for verbs), grammatical preference (construction types) and general principles which are structure-based, domain, language and application independent (cf Kinyon 99a, b).

The lexical preferences code either a category preference or a valence principle. They have to be computed for each word, using annotated corpora. The grammatical preferences code a construction preference, for example active over passive or personal over impersonal. The general principles assume a universal preference for economy (e.g. adjunction is more costly than substitution) and therefore favor analysis that needs to perform the fewer operations. This allows to capture widely accepted preferences, difficult to formalize in terms of constituent trees : idioms are preferred over literal interpretations, arguments are preferred over modifiers.

These general principles are the following :

1-Prefer the derivation tree with the fewer number of elementary trees (=fewer nodes)
2-Prefer to attach initial trees low
3-Prefer the derivation tree with the fewer number of auxiliary trees

Principle 1 favors the idiomatic interpretation of a sentence over its literal interpretation, since the different idiom chunks belong to the same elementary tree. Principle 2 favors the low attachment of arguments, when several alternative attachments are possible . Principle 3 favors the derivation tree involving the fewer number of adjunctions (i.e. modifiers) and favors the attachment prepositional phrases as arguments rather than modifiers. In case of conflict, the priority is for lexical preferences, then grammatical preferences, then general principles[1].

5.2. Application to TSNLP

Before disambiguation, the parsed item from TSNLP had an average of 2.9 parses per item. No categorial ambiguity remained. Most feature ambiguities are handled via underspecification (eg *"les enfants-the children"* feminine ou masculine). The remaining (structural) ambiguities are the following (not all of these are spurious) :

- modifier adjoined to S ou V after an intransitive verb (l'ingénieur viendra volontiers),
- prepositional phrase analyzed as complement or modifier (L'ingénieur préfère le vin à l'eau; Il passe pour un spécialiste),
- passive with or without agent (the par-PP can be analyzed as an agent phrase or as a modifier)
- several adjunction sites in case of multiple modifiers.

After applying the general preference principles, we are left with only 2.17 derivations / sentence. It is intersesting to note that no conflicts were encountered between the 3 general principles. After applying in addition the language specific preferences, we are left with 1,5 derivation / sentence. It turns out that one of the main sources of spurious ambiguities lies in adverbial attachment. We are exploring how to add lexical preferences to deal with this case.

6. Maintenance of the Grammar

Thanks to the MetaGrammar, which allows to generate semi-automatically an LTAG, the number of

[1] It is interesting to note that contrary to general principles formulated on constituent trees, our principles formulated on dependancy structure yield no conflicts. A discussion on the cognitive relevance of these principles can be found in (Kinyon 99c, Kinyon 00c)

trees has augmented drastically: from 650 trees for the manually written grammar, we reached 5000 elementary trees in 2000. Maintaining and updating such a large database is difficult :

For example, one can generate a large grammar using the MG, 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).

Moreover, 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).

In addition, 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 ?

To remedy this problem, we have developed a tool which we call Lexik (cf Barrier & al. 00). The goal is twofold :

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

6.1. Lexik : a maintenance tool

We've seen in 2.1 that 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

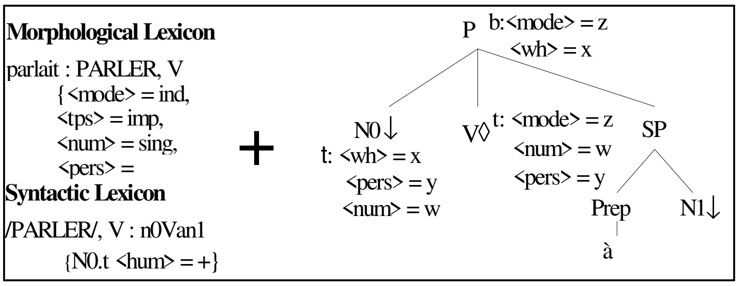

Figure 3 : 3 sources of Information

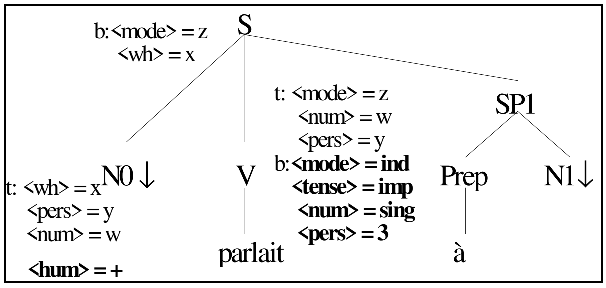

Figure 4 : Corresponding Elementary Tree

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 3) and outputs one hand fully lexicalized trees (figure 4) 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 5, a sample error file can be seen on figure 6.

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

LEMME: abaisser
ENTRY: abaiss'e
TREES: n0Vn1as2-sa2 n0Vn1as2 R1n0Vn1as2-sa2 R1n0Vn1as2 C1n0Vn1as2-sa2 C1n0Vn1as2 n0Vn1as2-cl1-sa2 n0Vn1as2-cl1 W1n0Vn1as2-sa2 W1n0Vn1as2 n0Vn1as2-inf-sa2 n0Vn1as2-inf R1n0Vn1as2-inf-sa2 R1n0Vn1as2-inf C1n0Vn1as2-inf-sa2 C1n0Vn1as2-inf n0Vn1as2-inf-cl1-sa2 n0Vn1as2-inf-cl1 W1n0Vn1as2-inf-sa2 W1n0Vn1as2-inf n0Vn1as2-coord-sa2 n0Vn1as2-coord n0Vn1as2-coord-cl1-sa2 n0Vn1as2-coord-cl1 n0Vn1as2-im-sa2 n0Vn1as2-im n0Vn1as2-clinv-sa2 n0Vn1as2-clinv n0Vn1as2-clinv-cl1-sa2 n0Vn1as2-clinv-cl1 W1n0Vn1as2-clinv-sa2 W1n0Vn1as2-clinv n0Vn1as2-cl0-sa2 n0Vn1as2-cl0 R1n0Vn1as2-cl0-sa2 R1n0Vn1as2-cl0 C1n0Vn1as2-cl0-sa2

Figure 5 : Sample output

```
Opening syntax Files...
Opening verbes.txt... Done
Opening tree Files...
Opening lex.new... Done
Opening modif.new... Done
Opening Family n0Vn1as2... Done
#V_DAT- not found (from syntax file)
#V_DAT- not found (from syntax file)
reduire n'a pas d'entrée dans le dictionnaire morpho
Opening Family n0Van1-a... Done
#V_DATH- not found (from syntax file)
Family VAdpn not found... Skipping all entries
Family VAd not found... Skipping all entries
Opening Family cl0seVden1... Done
#V_REFL+ not found (from syntax file)
Opening Family cl0V-a... Done
#V_SING not found (from syntax file)
Opening Family a0Ad... Done
desespere n'a pas d'entrée dans le dictionnaire morpho
ferme-p n'a pas d'entrée dans le dictionnaire morpho
Family cl0seVdes1 not found... Skipping all entries
Family cl0seVs1subj not found... Skipping all entries
Family n0A_des1_ not found... Skipping all entries
Opening Family n0V_loc1__sbut2_-e... Done
Opening Family n0V-e... Done
```

Figure 6 : Error File

Lexik allows to detect 3 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, that can then be repaired (cf figure 6)

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 Doran & al 94, 00; Xtag group 95), while ours runs on FTAG for French . Since the 2 grammars resort to similar formats, it would be interesting to couple the 2 approaches.

6.2 Gathering information with Lexik

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

We took 1060 French verbs extracted from the 1 million word corpus LeMonde (cf Abeillé & Clément 99, Clément & Kinyon 00), and found out that verbs anchor on average 2.8 families / verb.

Only 7 of these verbs anchor 8 families or more[3] 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)*). Also, the two tree families n0Vn1 (transitive) and n0Vn1-à-n2 (ditransitive) are anchored by two thirds of lemmas

But staying at the level of families is not fine-grained enough : some verbs will anchor only some of the trees contained in a family[4]. Also, some trees in a given family will be favored in practice over other trees, depending on the preferences of realization of arguments for verbs. Thus, we wanted Lexik to go down to the level of trees.

We then find out that French verbs anchor on average 77,67 elementary trees (ranging from 1 to 800), whereas other parts of speech (i.e. nouns, adverbs, adjectives) only anchor between 1 & 2 trees.

Conclusion

We have presented the state of FTAG, a wide coverage grammar for French based on the LTAG formalisms.
Current and Future work includes :
- Improving a tool which allows to transform a FTAG derivation tree into a dependency tree (Clément 00)
- Implementing a hypertagger (cf Kinyon 00a, b) : a tool which allows to use the MG hierarchy in order to pack several supertags into one unique and linguistically motivated structure, building up on the work of (Srinivas 97)

References

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

Abeillé A., Candito M.H., Kinyon A. 1999. FTAG : current status & parsing scheme. Proceeding Vextal'99. Venice.

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

Abeillé A., Daille B., Robichaud B. 1996, FTAG : un analyseur de phrases françaises, Proceedings ILN, Nantes.

Barrier N. Barrier S. Kinyon A. 2000. Lexik : a maintenance tool for FTAG. Proceedings TAG+5 Paris

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

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

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

Catach, N 1984. Les listes orthographiques de base du français, Paris, Nathan.

Clément L 2000. Un outil de creation d'arbres de dependances. Proceedings TAG+5. Paris

Clément L. Kinyon A. 2000. Chunking, marking and searching a morpho-syntactically annotated corpus for French. Proc. ACIDCA'2000. Monastir.

Danlos L., 1998. GTAG : un formalisme lexicalisé pour la génération inspiré de TAG, TAL 39:2.

Doran C., Egedi D., Hockey, B., Srinivas, B., Zaidel M. 1994. The XTAG system : a wide coverage grammar for English, Proceedings 15th COLING, Kyoto, pp. 922-928.

Doran C. et al 2000, Evolution of the XTAG system, in Abeillé, Rambow (eds) Tree Adjoining grammars, CSLI.

Estival D. Lehmann S, 1997. TSNLP: des jeux de phrases test pour le TALN, TAL, 38:1, pp. 155-172.

Gross M., 1975. Méthodes en syntaxe, Hermann, Paris.

Joshi A. : 1987. *An introduction to Tree Adjoining Grammars.* In Mathematics of Language. A. Manaster-Ramer (eds). John Benjamins Publishing Company. Amsterdam.Philadelphia. pp. 87-114.

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

[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.

Julliand A., Brodin D., Davidovitch C., 1970. Frequency dictionnary of French words: 5000 words, Mouton, La Haye.
Kinyon A. 1999a. Parsing Preferences with Lexicalized Tree Adjoining Grammars : exploiting the derivation tree. Proceedings ACL'99. College Park.
Kinyon A. 1999b. Hiérarchisation d'analyses basée sur des informations dependancielles dans le cadre des LTAGs. Proceedings TALN'99. Cargèse.
Kinyon A. 1999c. Some remarks on the psycholinguistic relevance of LTAG. Proceedings CLIN'99. Utrecht.
Kinyon A. 2000a. Hypertags. Proceedings COLING'2000. Sarrebrucken.
Kinyon A. 2000b. Even better than Supertags : Introducing Hypertags ! Proceedings TAG+5. Paris
Kinyon A. 2000c. Are structural principles useful for automatic disambiguation ? Proceedings Cogsci' 2000. Philadelphia.
Kroch A., Joshi A., 1985. The linguistic relevance of Tree Adjoining grammars, Rapport Technique, University of Pennsylvania, Philadelphia.
Leclère C. 1989, Les tables de verbes du LADL, Langue française.
Salkoff M., 1973. Une grammaire en chaîne du français, analyse distributionnelle, Paris: Dunod.
Salkoff M., 1979. Analyse syntaxique du français, grammaire en chaîne J. Benjamin, Amsterdam.
Srinivas B., Doran C., Kulick S. 1995 : Heuristics and Parse Ranking. 4 th workshop on Parsing Technologies. Prag. Czech Republic.
Srinivas B., 1997. Complexity of lexical descriptions and its relevance for partial parsing, PhD thesis, Univ. of Pennsylvania.
Sarkar A., Wintner S. 1999 : Typing as a means for validating feature structures. Proceedings CLIN'99. Utrecht.
Wehrli E., 1997. L'analyse syntaxique des langues naturelles : techniques et méthodes, Paris: Masson.
Xtag group (1995) *A LTAG for English. Technical Report IRCS 95-03.* University of Pennsylvania.