LTAG Workbench: A General Framework for LTAG

Patrice Lopez

DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrcken, Germany
lopez@dfki.de

Abstract

This paper presents the LTAG Workbench, a set of graphical tools and parsers freely available for LTAG. The system can be view as a modern alternative to the XTAG system. We present first the outlines of the workbench including different graphical editors and two chart parsers. The encoding of resources and results is based on an XML application called TagML. We present then future works dedicated to speed efficiency: Optimization based on sharing techniques and preprocessing of features. The whole system has been developed in Java which allows a strong portability and interesting reusability properties.

1. Introduction

The success of a linguistic formalism can largely depend on the availability of dedicated tools. They are needed first for maintaining the consistency of a grammar and for checking its correctness, but also for proving the adequacy of a formalism for computational applications. Such tools raise several engineering problems that should not be neglected as portability, reusability, user-friendly graphical interface, easy installation procedure and recycling of existing grammars, see for instance (Erbach & Uszkoreit, 1990) for a overview of these problems. Focusing on these features, we present a set of freely available tools dedicated to the LTAG formalism (Joshi *et al.*, 1975) which aims to be an alternative to the XTAG system (XTAG research group, 1998). The *LTAG workbench* is still an on-going work and we hope that it will appear enough promising to give rise to interests and possible contributions from the LTAG community.

We present first the outlines of the current workbench including different graphical editors and two chart parsers. We introduce then our solution for resource management which is based on a XML application called TagML. The section 4 is dedicated to future optimizations for speed efficiency that emphasize precompilation techniques and sharing of computation on the basis of grammar redundancies.

2. The LTAG Workbench

2.1. Editors

The workbench proposes general editors for the set of elementary trees schema and for the morphologic and syntactic lexicon. The graphical editors are based on a general tree editor developed at Thomson-CSF (France). It includes a *lexicalizer* function, similar to the one of the XTAG system, that allows to visualize an instancied elementary tree given a schema and lexical entry. These editors covers the functionality of the XTAG system and include browsers for lexicons.

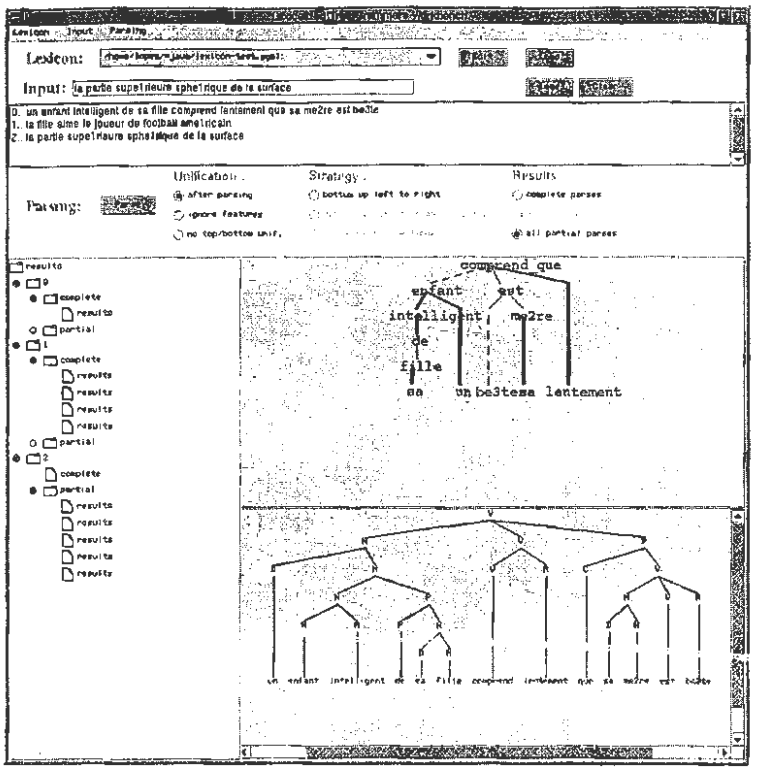

Figure 1: Screen shot of the LTAG parsing workbench.

2.2. Parsers

The workbench includes currently two parsers in a parsing test workbench (see screen shot on figure 1):

- A bottom-up *connection driven* parser which can deliver extended partial results compared with other classical bottom-up algorithms and without time penalty (Lopez, 2000).
- An implementation of the top-down Earley-like parser proposed in (Schabes, 1994).

Note that both of them are complete chart parsers, including extraction of results from the shared parse forest and two-step feature based processing. The bottom-up parser gives complete and partial parses considering several parsing heuristics with or without unification of the feature structures used in Feature Based LTAG. It is also possible to test and compare various parsing heuristics and strategies in term of speed efficiency.

2.3. Results

The system can deliver and edit different kinds of results: complete parses (derivation and derived trees) or partial parses, with complete unification, with only the first unification steps or without any unification. These different kind of results aims:

- To test a grammar by identifying the step involved in the failure of a parse during grammar debugging.

- To study out of grammar phenomena.

The workbench is implemented in Java for portability and reusability reasons. The Java sources, classes and documentation of the editors and parsing test workbench will be freely available by the end of May 2000. We present now another facet of the technical choices concerning the workbench: all the involved data are encoded with the highly portable formalism XML.

3. TagML

3.1. *Motivations*

A significant number of works are based on the TAG formalism. Still, for the moment, none has led to a common representation format of the grammars which would facilitate the exchange of TAG grammars and associated data or the development of normalised parsers and generic tools. A working group gathering people, mainly from TALaNa (University of Paris 7, France), ENST (Paris, France), INRIA (Rocquencourt, France), LORIA (Nancy, France) and DFKI (Saarbrücken, Germany) who are currently working on this formalism, made it necessary to define a shared and common representation with the aim of exchanging grammars and associated data, developing normalized parsers and specifying generic tools. Our proposal, TagML (Tree Adjoining Grammars Markup Language) is a general recommendation for the encoding and the exchange of the resources involved in LTAG. Anyone implementing a tool on the basis of this encoding can guarantee its interoperability with existing ones.

The XTAG system (XTAG research group, 1998), developed in the early nineties, offers the first workbench dedicated to LTAG grammar design and an Earley-like parser. However, this integrated parser provides only a binary answer (accepted or rejected sentence) hardly compatible with the test of a large grammar. Partial results and diagnostics about errors are necessary to test a grammar and to identify the step involved in the failure of a parse during grammar debugging. Thus, designing a new parser is justified but integrating new components to the XTAG system is technically very difficult for someone that has not been involved in the initial development of the system. More generally, this system has not been developed technically to be distributed since it is based on proper and non specified formats. It requires a narrowly-specialised skill for its installation, its usage and its maintenance. TagML can be viewed as a standardization and an extension of the XTAG formats and more generally as an answer to these technical problems. We present in the following sections the broad outlines of TagML, for more details see (Bonhomme & Lopez, 2000).

3.2. *Principles*

The definition of a generic tool for parsing and managing LTAG grammars supposes a common language specification, shared by the concerned community. The first step toward more generic and flexible tools undergoes the definition of an appropriate encoding for the management of large-size linguistic resources. This encoding should be able to structure possibly heterogeneous data and to give the possibility to represent the inevitable redundancies between lexical data. Given these expectations, we decided to define TagML as an application of the XML recommendation.

A LTAG grammar is defined by a morphological lexicon, a syntactic lexicon and a set of elementary tree schemas. The schema are ordered in tree families in order to capture the general aspects of the lexicalization process. This lexicalization is obtained on the basis of information given in the syntactic lexicon. For the moment, a complete Document Type Definition (DTD) has been proposed for the schema.

In an elementary tree schema, we can distinguish:

- The structural part, i.e. a partial phrase structure or a partial parsing tree.
- The set of feature equations constraining top and bottom feature structures.

We keep from (Issac, 1998) most of the elements involved in the encoding of schema structures:

- $< t >$: elementary tree, document that we specify in this part.
- $< n >$: general node, the attribute *cat* gives the category of this node and the attribute *type* distinguishes foot node, substitution node and anchor.
- $< fs >$: feature structure, of type *bottom* or *top*
- $< f >$: typed feature (attribute-value) similarly to the TEI. For typed feature equation and their re-usability, we introduce the element *linkGrp* as specified in the TEI to group internal or external links (element *link*) (Sperberg-McQueen & Burnard, 1994).

3.3. Structural component of schema

Similarly to (Issac, 1998) proposal, we represent straightforwardly the tree structure of a schema by an isomorphy with the XML tree structure (see figure 2).

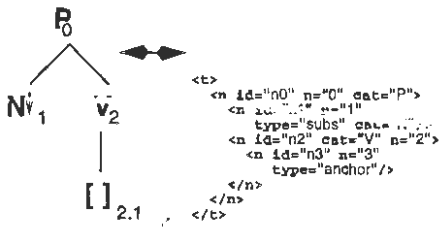

Figure 2: Isomorphy between the elementary tree schema and the XML tree structure

In practice in a broad-covering lexicalized grammar, the redundancy of common substructures is very important. For instance, the subtree dominated by a V category with a depth of 1 (the anchor and the pre-terminal category) is shared by most of the trees describing a verbal syntactical context (several hundred of trees for the English XTAG grammar, several thousand for the French LTAG grammar). This redundancy can be very useful to encode for linguistic or efficiency issues. In order to represent these redundancies, we propose to use the XML Link mecanism (DeRose *et al.*, 1999) and to identify systematically every nodes. We use the principle of virtual resources systematically to obtain only one representation of the different nodes within the whole grammar.

3.4. Feature equations

The TEI (Sperberg-McQueen & Burnard, 1994) proposes a recommendation for the encoding of feature structures that we propose to integrate to TagML. This standardization allows to type the features and to represent explicitly feature percolation. Note that the features used in the LTAG formalism have atomic values thanks to the extended domain of locality principle. The feature equations of an elementary tree schema can be view as a global term for a complete elementary tree, or as several terms distributed in the various nodes of an elementary tree sharing common variables. We propose to link directly the shared features in order to avoid the necessity to manage shared labels during the parsing of the features structures. These links are specified in *linkGrp*.

We have the possibility to give a type to a *linkGrp*, i.e. for a feature equation, for instance *subject-verb agreement*, then by identifying this *linkGrp* to share the corresponding feature equation to several elementary tree schemas. If we still consider the example of *subject-verb agreement* feature equation, the corresponding *linkGrp* will be shared by all elementary tree schemas that include this kind of agreement. The nodes corresponding to the features linked by percolation can be identified hy a special attribute which gives the function of each terminal node. The access to these specific nodes are obtained with the selection language proposed both for XSL Tranformation Language (Clark, 1999) and for the XML pointers called XML Paths (Clark & DeRose, 1999).
As we can see in figure 3, the percolated feature is linked to the *linkGrp* corresponding to the feature equation, so it is straightforward to access with this link all the other features which share the same value, without dealing with any labels and tables of labels.

```
<n cat="P" id="n0">
  <fs type="top" id="fs0">
    <f name="num" id="f0">
      <link xlink:type="simple"
            xlink:href="doc#id(I0)"/>
    </f>
    <f name="det" id="f1"><minus/></f>
  </fs>
  <fs type="bottom" id="fs1">
    /* ... */
  </fs>
</n>

/* External document */

<linkGrp type="accord">
  <link targets="
    id(n0)/fs[1][@type,top]/f[1][@name,num]
    id(n2)/fs[1][@type,bottom]/f[1][@name,num]"
        id="I0"/>
</linkGrp>
/* ... */
```

Figure 3: Shared features and factorisation of common feature equation

3.5. Tree family

In order to manage efficiently a set of elementary trees that could be quite large, TagML provides a mechanism allowing to gather elementary trees sharing the same sub-categorisation frame. A tree family is described (indicated by the tag $< tfamily >$) by defining a set of links to a subset of elementary tree schemas. The figure 4 presents an example of tree family definition (in this example *I1_VTA_0* and *I2_VTD_1B* refers to two elementary tree schemas for transitive verbs and *I2_adjectif6* and *I1_adjectif1* to two elementary tree schemas for adjective).
The encoding of the syntactic lexicon which is much more complex will be the subject of further research. The current system works with a very basic XML encoding of lexicon closed to the XTAG system flat representation.

3.6. Existing tools

Our implementations are based on the Silfide XML toolkit[1]. The following tools are currently available:

- A XSL style sheet allowing the automatic generation of Latex documentation from the TagML data.
- A conversion tool for the XTAG format.

[1] http://www.loria.fr/projets/XSilfide/EN/sxp/

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE tag SYSTEM "tagml.dtd">
<tag xmlns:xlink="http://www.w3.org/XML/XLink/0.9">
  <desc>Our tree families</desc>
  <tfamily name="transitive verb">
    <desc>Tree family for transitive verbs</desc>
    <t xlink:type="simple"
       xlink:href="l1_VTA_0.xml"
       xlink:show="replace"
       xlink:actuate="auto"/>
    /* ... */
    <t xlink:type="simple"
       xlink:href="l2_VTD_1B.xml"
       xlink:show="replace"
       xlink:actuate="auto"/>
  </tfamily>

  <tfamily name="adjective">
    <desc>Tree family for adjectives</desc>
    <t xlink:type="simple"
       xlink:href="A1_adjectif1.xml"
       xlink:show="replace"
       xlink:actuate="auto"/>
    /* ... */
    <t xlink:type="simple"
       xlink:href="l2_adjectif6.xml"
       xlink:show="replace"
       xlink:actuate="auto"/>
  </tfamily>

  /* ... */
</tag>
```

Figure 4: Sample of a TagML document and two tree families

Every parser which respects the XML encoding of LTAG resources and a specific Java API can be directly integrated to the LTAG Workbench.
We plan to improve the conversion tool by performing a grammar simplification and compaction at various levels. We will see in the next section that our main goal here is to exploit redundancies of data to to reduce the processing cost.

4. Sharing computation and feature processing optimization

Consequence of the important size of existing large-coverage Lexicalized TAG grammars, the current parsers suffer from a lack of speed performance. Speed is an important factor for real-world application but also because the tools are constantly used during grammar development. We argue the improvement of LTAG parsers and tools depends on how the huge amount of data consequence of the lexicalization can be put into factor in order to share the computation. The initial idea is structure sharing by the way of Finite State Techniques for the elementary tree skeletons (Evans & Weir, 1997). Still we will see that similar sharing for feature equations and derivation extraction is also possible.

4.1. Structural Sharing

Lexicalization raises the problem of multiplication of the same substructures which can be serious. In Context Free Grammars the same rule can be used for all possible parsing trees which contain the corresponding substructure, but in Lexicalized Tree Grammars these substructures are duplicated. Considering classical linguistics choices for LTAG grammar design, polystructures (example : *to speak to..., speak about..., to speak to ... about...*) are very common, the corresponding elementary trees must share common substructures and therefore do not cost as much as an independant elementary tree for each.
(Evans & Weir, 1997) shares different substructures of elementary trees using Finite State Automata (FSA) and classical minimizing techniques. As presented in (Evans & Weir, 1997),

the authors use automata corresponding to one particular traversal of the trees. We use similar techniques to share here linearized structures between different elementary trees and obtain automata similar to the ones presented in (Roche, 1996). The main difference with (Evans & Weir, 1997) is that, since it represents elementary trees for any kind of tree walk, this FSA does not impose a specific strategy during the parsing.

When FSA are shared in a single one, each state contains identifiers of the elementary trees which pass through it and each item the list of elementary tree identifiers valid for the item's positions. To test conditions of a rule we must consider every possible transitions paths according to the shared FSA. The resulting item of a rule is valid for a subset of identifiers of the elementary trees passing through the both position states. The "uncompaction" can be done when we enumerate the derivations.

4.2. Preprocessing of center features

In Feature-Based LTAG, two sets of features, top and bottom, are associated to each nodes. This separation is necessary because of potential adjunctions which can change the value of a possible feature at a given node. Categories used as node labels are never changed after an adjunction which only capture recursive structures (i.e. root and foot node of auxiliary trees must have the same label). We say that the category value at a given node is monotonic according to the adjunction operation. We propose to define an additional set of features, called *center features*, in order to gather features which are also monotonic according to the adjunction operation. The main interest of this new set of features is computational efficiency by the improvement of the predictive power of the grammar. The set of possible trees for an attachment at a given node N is not only trees with a matching category but also trees that present unifiable center features.

The center features can be computed easily simply by identifying which features are never changed by any existing auxiliary trees. Unfortunately, considering the whole XTAG grammar for instance, we can always find an auxiliary tree modifying the value of a given feature. Still, it is possible to compute significant center features considering only the subset of the grammar which is valid after the lexicalization process.

Features as *aux* or *det* of the French LTAG grammar should still need a separation in top and bottom features, but many others, in particular morphological features as *num* or *gender*, will be in general monotonic for adjunction after the lexicalization process. With this simple preprocessing, we expect a significant speed-up factor during parsing.

4.3. Sharing of feature equations

Similarly to the problem of redundancy of common substructures between different elementary trees, the same *feature equations* (i.e. the same kind of percolation of feature values) are duplicated in many trees. For instance the subject-verb agreement could be shared between hundred of trees (Candito, 1996). Our idea is to associate a unique feature term to the set of derivations and to improve the sharing of the corresponding DAG. Given a feature equation, this improvement supposes to identify the common nodes which are linked by the feature percolation. This identification can be done not on the basis of similar Gorn Adress of nodes but by identifying the functions (subject, object1, object2, syntactic verbal head, ...) associated to each nodes. For instance the subject-verb agreement percolates feature values linked to the subject and the syntactic verbal head (main verb, modal or auxiliary). This feature equation could be evaluated only one time for all elementary trees (i) containing this feature equation and (ii) combined with the same elementary trees at nodes with the same functions.

5. Conclusion

We have presented a general framework dedicated to LTAG grammars with a special regard to portability and reusability. A lot of efforts are still necessary to achieve efficiency and practical real-world application but we have proposed some possible optimizations and, more generally, an ambitious basis which can be freely exploited and enriched.

Aknowledgement

We would like to thank all the participants of the TagML working group: Anne Abeillé, Nicolas and Sébastien Barrier, Marie-Hélene Candito, Lionel Clement, Alexandra Kinyon and Kim Gerdes (TALaNa), Patrice Bonhomme, Laurence Danlos and Laurent Romary (LORIA), Pierre Boullier, Eric de la Clergerie and Philippe Deschamp (INRIA Rocquencourt), Ariane Halber (ENST), Fabrice Issac (UT Compiegne), Yannick de Kerkadio (LIMSI), Rodrigo Reyes (LCR Thomson) and David Roussel (Aérospaciale).

References

BONHOMME P. & LOPEZ P. (2000). TagML: XML Encoding of Resources for Lexicalized Tree Adjoining Grammars. In *Second International Conference on Linguistic Ressources and Evaluation (LREC'2000)*, Athens.

CANDITO M.-H. (1996). A principle-based hierarchical representation of LTAGs. In *COLING'96*, Copenhagen, Denmark.

CLARK J. (1999). *XSL Transformations (XSLT) Version 1.0.* W3C, `http://www.w3.org/TR/-xslt`. W3C Recommendation 16 November 1999.

CLARK J. & DEROSE S. (1999). *XML path language (XPath) version 1.0. W3C,* `http://-www.w3.org/TR/xpath`, *8 October 1999.*

DEROSE S., ORCHARD D. & TRAFFORD B. *(1999).* XML linking language (XLink). W3C, `http://www.w3.org/TR/WD-xlink`, 26 July 1999.

ERBACH G. & USZKOREIT H. (1990). *Grammar Engineering: Problems and Prospects - Report on the Saarbrücken Grammar Engineering Workshop.* Technical report, CLAUS-Report Nr.1.

EVANS R. & WEIR D. (1997). Automaton-based Parsing for Lexicalized Grammars. In *Fifth International Workshop on Parsing Technologies*, Cambridge, Mass.

ISSAC F. (1998). A Standard Representation Framework for TAG. In *Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+4).*

JOSHI A., LEVI L. & TAKAHASHI M. (1975). Tree adjunct grammars. *Journal of the Computer and System Sciences.*

LOPEZ P. (2000). Extended Partial Parsing for Lexicalized Tree Grammars. In *International Workshop on Parsing Technology (IWPT2000)*, Trento, Italy.

ROCHE E. (1996). *Parsing with Finite-State Transducers.* Technical report, TR-96-30, Mitsubishi Electric Research Laboratories (MERL).

SCHABES Y. (1994). Left to Right Parsing of Lexicalized Tree Adjoining Grammars. *Computational Intelligence*, **10**, 506–524.

SPERBERG-MCQUEEN C. & BURNARD L. (1994). *TEI guidelines for electronic text encoding and interchange (P3). Chicago and Oxford,* `http://etext.virginia.edu/TEI.html`.

XTAG RESEARCH GROUP *(1998).* A Lexicalized Tree Adjoining Grammar for English. *Technical report, IRCS 98-03, University of Pennsylvania.*