Lexicalized Grammar Specialization for Restricted Applicative Languages

Patrice Lopez, Christine Fay-Varnier, Azim Roussanaly

Lexicalized Grammar Specialization for Restricted Applicative Languages

Patrice Lopez, Christine Fay-Varnier, Azim Roussanaly

LORIA, INRIA Lorraine and Universities of Nancy
BP 239, 54506 Vandœuvre-lès-Nancy, France {lopez, fay, azim}@loria.fr

Abstract

In the context of spoken interfaces, we present a practical methodology and an implemented workbench called EGAL (Lexicalized Tree Grammar Extraction) dedicated to design and test restricted languages used in specific task-oriented applications. A complementary methodology is proposed to process the extraction of these applicative languages from a general LTAG grammar and a training corpus. Additional results allow us to estimate the representativeness of the training corpus. An application of the system is presented for the tuning of a LTAG grammar dedicated to a spoken interface on the basis of a Wizard of Oz corpus.

1. Introduction

1.1. Motivations

In the case of a spoken dialogue system, the quality of the human computer interaction largely depends on the ability of the computer to understand spontaneous utterances normaly used by humans. The practical development of a spoken interface for a restricted domains implies that we perform the tunning of existing lexicon and grammar to a particular application. This paper proposes a methodology and an implemented workbench called EGAL (Lexicalized Tree Grammar Extraction) dedicated to design and test restricted natural languages used in specific task-oriented applications. This workbench is a sub-component of a general platform for designing spoken language systems and addresses software designers who are non-experts in natural language processing.

Specializing a grammar for restricted domains supposes at least the two following tasks:

- Cutting down the existing lexicon and grammar.
- Adding new words and new syntactic constructions.

In recent years, the development of large covering lexicalized grammars could be observed. Complementary, studies about the use of this kind of formalism for parsing spoken language have been performed. To address spoken disfluencies and robustness constraints in the context of human computer interaction, additional mechanisms have been proposed which often depend on the application domain. At the lexical and syntactic level, the following adaptations are required:

- Model spoken phenomena that could be considered agrammatical or rare in written language but frequent in spontaneous speech such as ellipsis or interpolated clauses (Price et al., 1989).
- Use robust parsing techniques to take into account the variability of the input.
- Specialize a lexicon and a grammar dedicated to text to a specific kind of dialogue and a specialized domain.

This paper addresses the last point. The specialization of a general hand written grammar to a specific domain is not a trivial task. Probabilistic methods and grammar inference as (Bod, 1995) can be seen as an alternative to this problem. Still a linguistically motivated hand written grammar provides a precise understanding of the occuring phenomena and reusability. In particular, this kind of grammar allows us to take into account the important ambiguity of the syntactic level. This ambiguity is one of the main differences between natural language that we want to process and regular languages which are just an approximation of natural language. Moreover, probabilist methods need very large annotated training corpora. Their development can require the same amount of effort as the writing of a wide-covering grammar.

We present in this paper a methodology and an implemented system called EGAL (Lexicalized Tree Grammar Extraction), able to perform an assisted specialization of a general grammar in order to obtain an applicative sublanguage from a corpus. When the specialized grammar has been obtained, a parsing module allows the evaluation of the grammar on a test corpus and the choice between various parsing algorithms and strategies. The partial and complete derivations can be visualized and compared following different criteria. The methodology also allows us to obtain information about the representativeness of the initial training corpus. Finally, the lexicalized grammar and the parser can be integrated in concrete HCI systems.

The proposed workbench can be applied to various domains. Our main goal is to design generic and portable spoken systems that can process spontaneous language. To illustrate our methodology and system, we have chosen a target application and collected an experimental Wizard of Oz corpus from which we have extracted a lexicon and a specialized grammar. We have finally evaluated the representativeness of the resulting grammar.

1.2. Lexicalized Tree Adjoining Grammars

The lexicalization of a syntactic formalism consists of the association of a set of appropriate syntactic contexts to each entry of the lexicon. Lexicon and grammar are merged in a single entity called *syntactic lexicon*. Lexicalization provides at least two main advantages: First the ability to describe syntactically each specific lexical entry allows us to choose the required complexity of the syntactic structures with flexibility. Even for restricted domains, too much generalization in syntactic descriptions generally results in unexpected border effects. Secondly the lexicalization allows parsing heuristics since a lot of syntactic ambiguity problems become lexical ambiguities which are easier to process (Abeillé, 1991).

The choice of the formalism is essential for the representation and the understanding of linguistic phenomena. It is also important to consider its applicability for NLP applications. The Lexicalized Tree Adjoining Grammars (LTAG) (Joshi and Schabes, 1992) is interesting for parsing and generation thanks to the lexicalization property and extended domain of locality. Linguistic studies and large-covering grammar developments for example in English and French have shown the practical interest of these properties. Moreover probabilistic models based on LTAG as stochastic TAG (Srinivas, 1997) or supertagging (Srinivas, 1997), allow optimizations for the processing of lexical and syntactic ambiguities on the basis of preferential choices. These properties make the LTAG formalism interesting for spoken utterances understanding (Halber, 1998) and generation in spoken systems (Becker et al., 2000).

Still the lexicalization has some drawbacks, in particular the task of designing of the grammar. Still work in progress, the English grammar of the XTAG system (Doran et al., 1994) already took ten years of development, the French grammar (Abeillé et al., 1994) more than seven years. A large covering grammar can include several thousand of elementary tree patterns called *schemata* (Candito, 1999) and a syntactic database that gives for each lemma the set of corresponding trees or tree families. Considering a given application, the use of the whole general grammar would lead to a prohibitive number of hypotheses. Moreover our goal is to avoid the development of a new grammar for each new application.

Work on the use of LTAG for dialogue systems for both parsing and generation of a sublanguage has been done recently, but the tuning of a general grammar to a specific application and domain remains a problem for the practical application of such a lexicalized formalism. The extraction of sublanguage grammars for LTAG has been discussed in (Doran et al., July 1997). But the proposed solution was based on successive manual approximations by experts. No practical methodology was proposed. No significant features have been identified that could help to perform more efficiently this task or that could lead to a software engineering solution.

2. Collection methodology

2.1. Restricted language

A restricted language can be defined as a set of utterances linked by a restricted domain, used for a particular function and generated by a specific grammar and vocabulary (Deville, 1989). Two factors limit the general language: The kind of discourse or dialogue which is realized and the application domain of the system. A restricted language is not only a subset of the whole language since an application can use technical terms which are only relevant for the domain. Moreover even in limited domains, the size of the vocabulary and the syntactic constructions change as the application evolves. Consequently a system has to propose a methodology to add new words and new syntactic contexts for the structures that would not be covered by a general grammar.

The practical advantages of the restricted language definition are a reduction of the combinatoric complexity of the processing and the ability to use a hand-written grammar (which is for example not realistic for dictation systems).

In the case of spoken dialogue systems, we claim that the systems should not understand words out of the corresponding restricted language because such words do not belong to the competence of the system. The lexicalized grammar defines here the norm of the applicative language, i.e. what is acceptable or not. Since domain restricted applications should not understand every user's request, they eventually have to lead additional dialogues with the user in the case of out of domain words.

2.2. Wizard of Oz experiments

The Wizard of Oz experiments are now widely used as a first step of the design of a spoken dialogue system. This experiment consists in the simulation of a spoken dialogue system in order to get a set of possible user interactions for a given application. The resulting corpus (which has a subjective representativeness) becomes a reference for the linguistic modeling. In other restricted domain applications, such as automatic thematic classification of e-mail in e-commerce, a similar step is necessary.

One of the main problems related to this kind of corpus is its representiveness for the application sublanguage we want to model. If the principle of restricted language is relevant, we can expect that by increasing the size of the training corpus, we will reach a size such that any addition will not result in a significant increase in the vocabulary or the size of the grammar.

Our approach consists first of obtaining a corpus which is classically divided in two parts. The first part is used to design the grammar of the restricted language (*training corpus*). The second one is dedicated to test (*test corpus*).

We have presented the different aspects which are essential for the kind of system we want to build: WoZ Experimental approach in order to obtain a corpus, specialization/designing of a lexicalized grammar dedicated to spoken language understanding, test of the resulting grammar and representativeness evaluation of the training corpus.

We have not found any existing workbench for lexicalized grammar which would combine all these aspects.

3. Presentation of the workbench

The general organization of a lexicalized tree grammar dedicated to parsing relies on three main knowledge sources:

- A morpho-syntactic database which associates an inflected form, a syntactic category and a set of morphological features.
- A syntactic database which associates a given lemma to a set of elementary trees representing the valid syntactic context for this lemma.
- A set of schemata (Candito, 1999).

The grammar designing/tuning module of the system is based on these three kinds of databases (see figure 1).

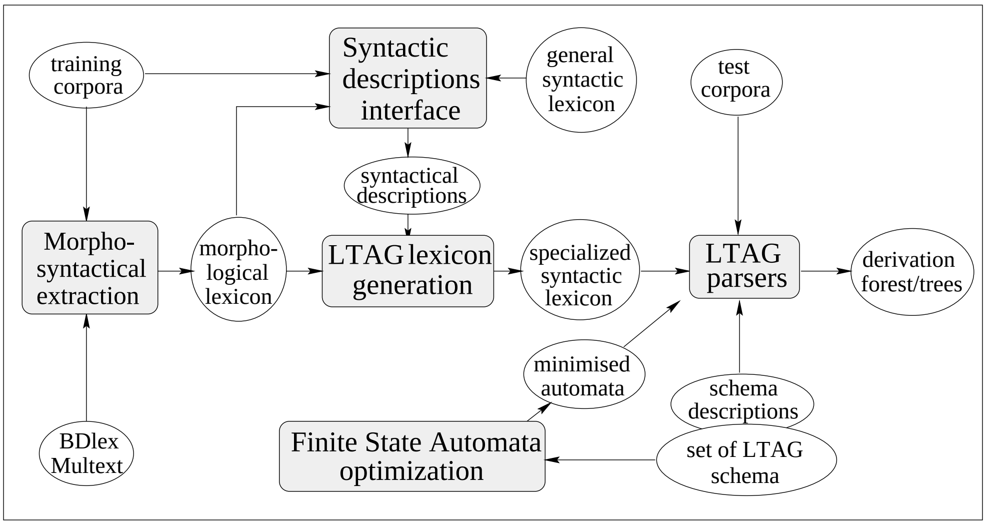


Figure 1: Overall presentation of the EGAL workbench.

3.1. Assisted generation of the lexicons

Morpho-syntactic extraction Given a training corpus, this step just corresponds to the exploitation of existing morpho-syntactic databases, Multext and BDlex (Ide and Véronis, 1994), by extracting the required information for all the words used in the corpus. This process has been implemented with an automaton-based compilation of the morpho-syntactic databases.

Set of schemata We assume that we already have a set of schemata (non-lexicalized elementary trees). For instance this schema can come from an existing hand-written grammar or from an automatic tree generation system as proposed by (Candito, 1999). A graphical editor allows the design of new schemata or the modification of existing ones.

Syntactic descriptions The goal of this module is to identify the syntactic properties associated with a lemma in order to select its correct syntactic structures. This identification is not an automatic process since resources able to enumerate all the possible predicative structures for a given lemma are not available. This result is obtained on the basis of a graphical interface dedicated to non-grammarian users.

The main idea is to associate a term of syntactic features to characterize (i) the various possible syntactic contexts covered by the general grammar (i.e. the various LTAG schemata), (ii) each lemma of a given corpus on the basis of a linguistic test suite illustrated by examples. The unification of these two structures characterizes then the precise subset of the acceptable syntactic constructions for each lemma.

The definition of our syntactic feature set is based on linguistic studies of French (mainly (Abeillé, 1991)). The current system uses nineteen syntactic features for the characterization of a verbal context (for example arity, passive, subject-verb inversion, support verb, equi-verb, reflexive, auxiliary,...) and a frame of possible prepositions. An alternative would be to use the syntactic features corresponding to the metagrammar described in (Candito, 1999) and the corresponding grammar generation system: In this case the description term corresponding to the schema that would be obtained automatically with the generation of the schemata.

For each syntactic feature we create a linguistic test composed by a question labeling the set of possible values and a set of examples. The tests are stored in a declarative way in a XML document. This XML document is then used by a generic test interface that allows a user to fill the frame for each lemma in a friendly way. The result of these tests consists of a feature term which is the syntactic description of the lemma.

For example the two following questions begin the French linguistic tests for verbs:

- Which auxiliary is used with the verb? (one between *être* and *avoir*)
- Can the verb be used in an intransitive/transitive/ditransitive context?

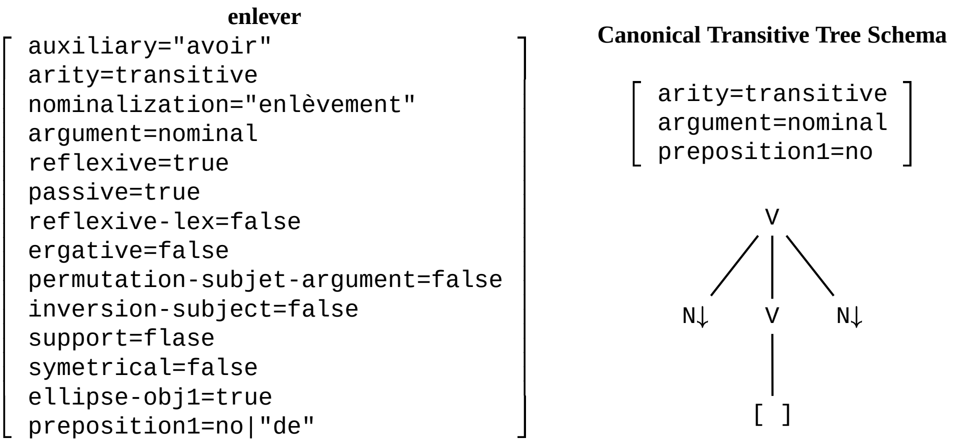
The tests continue until the complete frame of syntactic features and the preposition frame are specified.

The unification of the terms associated to the different schemata and the term obtained for a given lemma gives the correspondence between an entry of the lexicon and the subset of schemata that can be anchored by this entry. For instance on figure 2, the tree schema can be used with the lemma *enlever* since the two syntactic descriptions can be unified. This lexicalization process is uniform with the lexicalization performed on the basis of morphological features (for instance infinite verbs only lexicalize infinitive contexts).

This module can be used in two different ways:

- Completion of the whole list of linguistic tests in order to characterize completely a lemma for all its possible uses.
- Characterization of the syntactic contexts observed in the training corpus.

For the proposed methodology, the second possibility must be chosen. The list of utterances (in the training corpus)



```
enlever
[ auxiliary="avoir"
  arity=transitive
  nominalization="enlèvement"
  argument=nominal
  reflexive=true
  passive=true
  reflexive-lex=false
  ergative=false
  permutation-subjet-argument=false
  inversion-subject=false
  support=flase
  symetrical=false
  ellipse-obj1=true
  preposition1=no|"de" ]

Canonical Transitive Tree Schema
[ arity=transitive
  argument=nominal
  preposition1=no ]
```

Figure 2: Two examples of syntactic descriptions: one for the French lemma *enlever*, one for a transitive tree schema.

which contain the lemma and the linguistic tests are proposed simultaneously to the user by the graphical description tool. In our methodology, contrary to the classical approach for cutting down the grammar, we specify each entry of the lexicon in terms of its category and also in terms of its correct syntactic contexts. The resulting grammar is really a *lexicalized* subgrammar.

We do not use the principle of tree family used by the XTAG system because of the small size of the lexicon and for reasons of computational efficiency. With tree families, the final selection of trees associated to an entry of the lexicon is obtained dynamically by unification at the time of instantiation. Here the correct trees are already predefined and listed in the syntactic lexicon.

A complementary tool for linguists allows the design of linguistic tests. We note that:

- The descriptions obtained by filling the features frame are independent from the lexicalized formalism. For instance, one could use HPSG lexical types.
- This module allows us to integrate easily new words to a system by characterizing the inflected forms which are not recognized during the morphological extraction. Moreover a very important point is that adding new words with this tool can be done by a non-linguist user if the linguistic tests are correctly written.

Automatic generation of the specialized LTAG syntactic lexicon This step produces the syntactic lexicon by exploiting information from the three databases described before. We add to each entry of the morphological lexicon the list of LTAG schemata which can be lexicalized. This list is obtain by

- The unification of the morphological features of the flexed form with the morphological features of the node to be anchored.
- The unification of the syntactic feature term that describes the corresponding lemma with all the syntactic feature terms of the schemata.

The links to schema are simply noted with external references using the XML links mechanisms. The final anchoring is classically done as a pre-parsing process.

3.2. Parsing test workbench

After the generation of a grammar for an applicative sublanguage given a training corpus, this module aims to test the results on a second test corpus. It allows us:

- To visualize the parsing results (both partial and complete ones).
- To check the generated grammar and possibly change manually some data in the syntactic lexicon or the set of schemata.
- To test and to compare various parsing heuristics and strategies.
- To study out of grammar phenomena.

This workbench implements two chart parsing algorithms and several parsing heuristics:

- A bottom-up connection driven algorithm that delivers extended partial results (Lopez, 23 25 February 2000).
- An implementation of the top-down Earley-like algorithm of (Schabes, 1994).

The bottom-up parser gives complete and partial parses with or without unification of features structures used in Feature Based LTAG. These different kinds of results aim to test the grammar by identifying the step involved in the failure of the parsing.

3.3. Technical choices

The implementation have been made in Java for portability reasons. All the involved data are encoded in the highly portable formalism XML. A specific application of XML dedicated to resources used with LTAG has been developed called TagML (Tree adjoining grammar Markup Language) (Lopez and Roussel, 2000). TagML allows an efficient representation of these data in term of redundancy. For instance it is possible to encode only one time substructures that are redundant in several schemata. Similarly it is also possible to share feature equations occuring in several schemata. All these redundancies imply redundant computation that could be avoided. This standard representation allows easy resource exchanges with our research partners and allows the sharing and the comparison of tools. The DTD allows us to check the consistency of the whole grammar. Every parser that respects this encoding norm can be integrated to the parsing workbench very easily.

The Java sources, classes and documentation of the parsing test workbench, including editors, are freely available on request. The other modules should also be packaged and available at the time of the conference.

4. Grammar of the GOCAD corpus

4.1. A target application: GOCAD

The GOCAD application aims to model geological surfaces. The protocol and the Wizard of Oz experiment used with this application are presented in (Chapelier et al., 1995). This experiment allowed us to obtain a corpus which has been encoded following the TEI specifications[1]. This corpus of transcribed French spoken utterances is presented in Table 1.

4.2. LTAG for the applicative restricted language

The corpus has been divided in a training corpus (80% of the utterances) and a test corpus (20%). The size of the LTAG grammar obtained with the EGAL system is presented Table 2. The total number of links to schema is a good metric for the whole size of the syntactic lexicon.

Given this specialized lexicalized grammar, the average time for parsing is 167 ms per utterance with an average lenth of utterances of 6.42 words per utterance on Sun Ultra 1. It is difficult to compare with results obtained with the complete French LTAG grammar because first the covering of this complete grammar is really limited for this corpus (124 unknown words). Moreover, for technical reasons, this grammar has been designed for the XTAG system which is very difficult to install (SunOS 4 only for instance) and use. For indication, the parsing of sentences of 10 to 15 words can take more than ten minutes.

4.3. Representativeness of the training corpus

The morphological extraction phase and the generation of the syntactic lexicon for GOCAD are fast (less than one second for the first one, less than ten seconds for the second on an average workstation). Consequently it is possible to realize systematic tests to study the evolution of the generated data. The method consists of first randomly selecting utterances from the whole corpus and then generating the corresponding LTAG grammar. This allows us to study the evolution of the size of the grammar given the number of links to a schema in function of the number of utterances taken into account. A decrease of the slope of the curve indicates an improvement of the coverage. A horizontal asymptote would mean that the coverage of the grammar is perfect for the target sublanguage. The Figure 3 gives the evolution observed for the GOCAD corpus: The number of new structures obtained by considering the last two hundred utterances is very low and we can conclude that the final generated grammar is a good approximation of the GOCAD sublanguage.

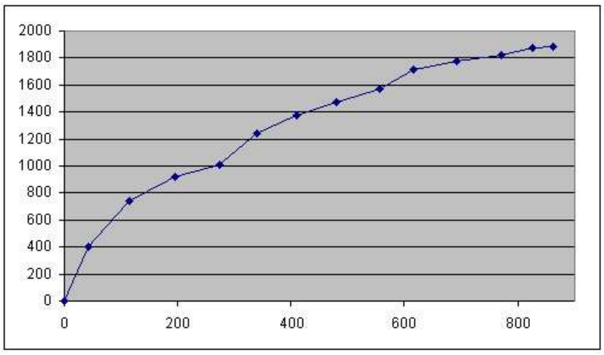


Figure 3: Evolution of the size of the generated LTAG grammar (number of links to schema) as a function of the size of the training corpus (number of utterances)

Such a result can be very useful to estimate the size of the corpus needed to reach a satisfactory covering rate. Covering 100% of the utterances is not our objective since in our approach only utterances corresponding to the competence of the spoken system need to be understood.

5. Future direction

We plan to see how the workbench scales up to other corpora and applications different than spoken interfaces. Our second goal is to extend the specialization workbench to cover multilinguality. One difficulty that arises is that the syntactic features used for the description of tree schemata and lemmas can be different from one language to another. It would mean that only a subset of these features has a real multilingual validity and could be used for parallel specialization of multilingual syntactic ressources. Syntactic features depending on the language might be limited if we only restrict them to pairs of languages, i.e. not considering all the languages at the same time.

6. References

Anne Abeillé, Béatrice Daille, and A. Husson. 1994. FTAG : An implemented Tree Adjoining grammar for parsing French sentences. In *TAG+3*, Paris.

Anne Abeillé. 1991. *Une grammaire lexicalisée d'arbres adjoints pour le français*. Ph.D. thesis, Université Paris 7.

Tilman Becker, Anne Kilger, Patrice Lopez, and Peter Poller. 2000. Multilingual generation for translation in speech-to-speech dialogues and its realization in verbmobil. In *ECAI'2000, Berlin, Germany*.

Rens Bod. 1995. *Enriching Linguistics with Statistics : Performance Models of Natural Language*. Ph.D. thesis, University of Amsterdam.

Marie-Hélène Candito. 1999. *Structuration d'une grammaire LTAG : application au français et à l'italien*. Ph.D. thesis, University of Paris 7.

Laurent Chapelier, Christine Fay-Varnier, and Azim Roussanaly. 1995. Modelling an Intelligent Help System from

[1] This corpus is available on the Silfide server (`http://www.loria.fr/projets/Silfide/`)

Number of user utterances	number of words	average number of words/utterance.
862	5535	6,42

Table 1: GOCAD corpus

number of inflected forms	number of schemata	number of links to schema
526	71	1776

Table 2: Size of the LTAG grammar corresponding to the training GOCAD corpus.

a Wizard of Oz Experiment. In *ESCA Workshop on Spoken Dialogue Systems*, Vigso, Danemark.

Guy Deville. 1989. *Modelization of task-Oriented Utterances in a Man-Machine Dialogue System*. Ph.D. thesis, University of Antwerpen, Belgique.

Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. 1994. XTAG System - A Wide Coverage Grammar for English. In *COLING*, Kyoto, Japan.

C. Doran, B. Hockey, P. Hopely, J. Rosenzweig, A. Sarkar, F. Xia, A. Nasr, O. Rambow, and B. Srinivas. July 1997. Maintaining the forest and burning out the underbrush in XTAG. In *Workshop on Computational Environments for Practical Grammar Development (ENVGRAM '97)*, Madrid.

Ariane Halber. 1998. Grammatical factor and spoken sentence recognition. In *Workshop on Text, Speech and Dialog, Brno*.

Nancy Ide and Jean Véronis. 1994. Multext (multilingual tools and corpora). In *14th Conference on Computational Linguistics (COLING'94), Kyoto, Japan*.

Aravind K. Joshi and Yves Schabes. 1992. Tree Adjoining Grammars and lexicalized grammars. In Maurice Nivat and Andreas Podelski, editors, *Tree automata and languages*. Elsevier Science.

Patrice Lopez and David Roussel. 2000. Predicative LTAG grammars for Term Analysis. In *TAG+5*, Paris, France.

Patrice Lopez. 23-25 February, 2000. Extended Partial Parsing for Lexicalized Tree Grammars. In *International Workshop on Parsing Technology, IWPT 2000*, Trento, Italy.

Patti Price, Robert Moore, Hy Murveit, Fernando Pereira, Jared Bernstein, and Mary Dalrymple. 1989. The integration of speech and natural language in interactive spoken language systems. In *Proceeding of Eurospeech*, Paris, France.

Yves Schabes. 1994. Left to Right Parsing of Lexicalized Tree Adjoining Grammars. *Computational Intelligence*, 10:506–524.

Bangalore Srinivas. 1997. *Complexity of lexical descriptions and its relevance to partial parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.