ISCA Archive
http://www.isca-speech.org/archive

ISCA & IEEE Workshop on Spontaneous
Speech Processing and Recognition
Tokyo Institute of Technology, Tokyo, Japan
April 13-16, 2003

New Perspectives of Linguistic Study

How can linguistic theories help corpus-based techniques in NLP or vice versa?

Jun-ichi Tsujii
Deprtment of Computer Science
Faculty of Information Science and Technology
University of Tokyo
and
CREST, JST
(Japan Science and Technology Corporation)
tsujii@is.s.u-tokyo,ac.jp

Abstract

Corpus-based techniques and symbolic, linguistics-based NLP techniques have to be unified for further development of the field. The integration of these two streams of research will open up new perspectives of linguistic study that will unify rationalists' theories with empirical study of languages. Several research directions are discussed

1 Introduction

In the last decade, we have witnessed development of corpus-based techniques in NLP, which overshadows formal linguistic theories that had dominated the fields of computational linguistics and NLP in the preceding two decades. While the success of corpus-based techniques is significant, it seems that we have reached at a stage where new ideas are required for further development.

Corpus-based techniques here mean statistical modeling of language and NLP techniques derived mainly from theories in machine learning. Since we have applied almost all techniques in these fields from HMM to Decision Tree to Maximum Entropy to SVM, we may have to introduce new ingredients from linguistic theories of the previous decades in order to further advance corpus-based techniques by enriching the linguistic aspects of language models.

On the other hand, rationalistic linguistic theories will certainly benefit from the achievements of corpus-based techniques in the last decade.

In this paper, I would like to discuss several possible research directions that will lead to unification of the two streams in linguistic study and NLP, based on what our research group of the University of Tokyo is doing.

2 Statistical Models and Linguistic Theories

Statistical language models are based on *observables*. We start with a sequence of words that can be observed in order to construct a language model. A language model in speech recognition should be able to predict the next word, which is also observable, by observing the preceding sequence of words.

HMM in speech recognition, for example, has a hidden part, *hidden* in the sense that it is not observable. While the hidden model in HMM is simple like an FSA (Finite State Automata), HMM can predict next words with probability, which helps a recognizer recognize actual words in speech (Figure 1). The non-observable part of a model is important only so far as it contributes to prediction of following words.

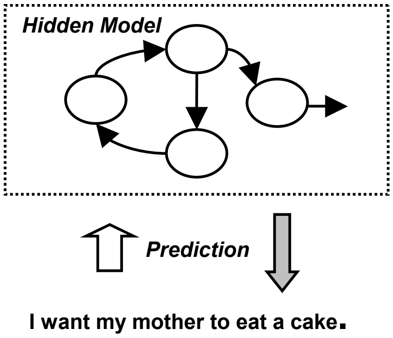

Fig.1 Language model in Speech Recognition

While HMM assumes a simple framework as the *base* linguistic model where a sequence of words is to be generated by successive state transitions, more sophisticated linguistic models like CFG fail to improve the predictive power of a language model.

Subtle constraints on language expressed by CFG do not contribute to probabilistic prediction of next words, even though they may reduce the size of a set of grammatical sequences of words in language over which probabilities are estimated

However, unlike speech recognition, techniques of NLP have to be more concerned with the hidden part of a language model. For example, in NLP, we are interested in assigning syntactic structures to given sentences, and a syntactic structure is representation of the hidden process (derivation history, i.e. how the surface word sequence is generated).

Furthermore, unlike word recognition in speech, syntactic structure assignment itself is not the ultimate goal in NLP. We are interested in syntactic structure because we assume that it has direct relationship with meaning conveyed by a sentence.

In general, syntactic structure cannot be arbitrarily defined. It has to be defined in such a way that the systematic relationship between a surface sentence and the meaning is revealed. It is the role of rationalists' theories to establish frameworks for relating these two *non-observable* structures systematically. Thus, the linguistic part of a language model inherently plays a crucial role in modeling of language in NLP

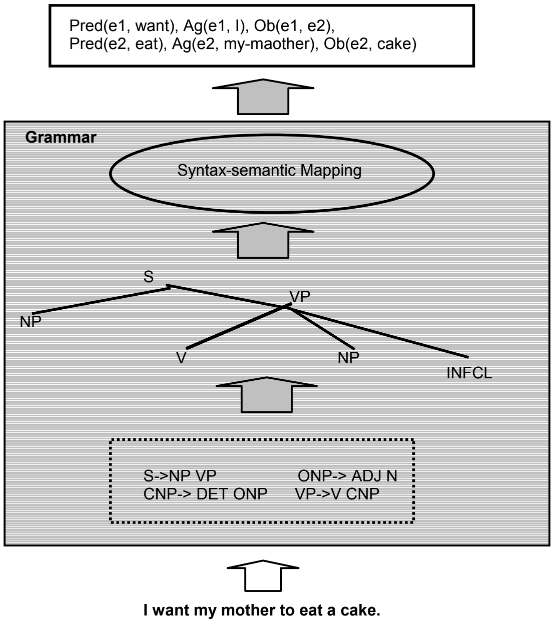
Pred(e1, want), Ag(e1, I), Ob(e1, e2),
Pred(e2, eat), Ag(e2, my-maother), Ob(e2, cake)
Grammar
Syntax-semantic Mapping
S
VP
NP
V
NP
INFCL
S->NP VP
ONP-> ADJ N
CNP-> DET ONP
VP->V CNP
I want my mother to eat a cake.

Fig. 2 Language Model in NLP

3 CFG, Lexicalized CFG and Unification-based Grammar Formalisms

There are two aspects of linguistic frameworks that affect language models. That is,

(1) Grammar formalism
(2) Specific grammar in the formalism

Even if we chose CFG as our formalism, there were many specific grammars (weakly equivalent grammars) to define the same language. A set of non-terminals can be different, and the same set of non-terminal symbols can be used to define different rewriting rules. Depending on specific grammars, the number of parameters to be estimated in statistical language models will change.

Furthermore, choice of formalisms has significant implications on a language model. We have witnessed a major shift from ordinary CFG to dependency grammar or lexicalized versions of CFG. They outperform ordinary CFG in the task of assigning phrase structures to sentences i.e. the task of structural tagging. This implies that influence from semantic structures or argument structures (the top part in Figure 2) cannot be ignored in structural tagging (the middle part in Figure 2).

While lexicalized CFG or Dependency grammar can capture the relationship between surface word sequences and argument structures in trivial cases, they fail to capture the relationship in cases such as those involving deletion, raising, long distance dependencies, etc.

In order to properly treat the sentences

(a) Who do you think John wants to eat?
(b) What do you think John wants to eat?

one has to consider their argument structures in which the relations among *who (what), eat,* and *John* are explicitly captured. Simple dependency grammar and lexicalized CFG fail to capture them.

While (a) and (b) are assigned the same phrase structure, they have different argument structures (or meaning structures). In order to judge which meaning structure is appropriate for (a) (or (b)), the relations among *eat, who (what)* and *John* have to be considered.[1]

[1] It is our contention that to capture these semantic relations is important not because they may contribute to better performance of structural taggers (we believe that it certainly does) but rather because it is the ultimate goal of a parser in most of NLP application.

An obvious extension of current language models such as lexicalized CFG is to use unification-based grammar formalisms. Since one of the major motivations in modern, formal linguistic frameworks such as HPSG, CCG, etc. is to capture not only grammaticality of word sequences but also the systematicity of mapping between word sequences and their meanings, these frameworks provide explicit mapping between surface sequences and their argument structures.

However, since unification-based grammar formalisms replace non-terminal symbols in CFG with feature bundles, it is not straightforward to define stochastic models on them. The flexibility and expressivity of unification grammar formalisms also make probability estimation more complicated and computationally expensive.

[Miyao 02] and [Johnson 02] discuss general frameworks of constructing stochastic models by ME for feature-unification grammars and show how to estimate parameters of such stochastic models efficiently. [Miyao 02] proposed a generalized version of inside-outside algorithm for unification-based grammar. While it remains to be seen whether such enhanced models can outperform lexicalized CFG, these are important steps to integrate statistical language models with rationalists' theories

4 Grammar Learners from Annotated Corpora

While we assume in Section 3 that the linguistic part of a language model is given in advance before estimating statistical parameters, there have been attempts in which the linguistic part of a model is also constructed from structurally annotated corpora. In these attempts, since the linguistic part of a model (actual rules) is to be constructed according to annotations, one can avoid arbitrariness of rationalists' theories, and grammar thus constructed is consistent with annotations attached to actual language uses.

Estimation of statistical parameters in Section 3 is to be performed for the grammar thus extracted from corpora (Figure 3 and 4).

Though this approach to construction of the linguistic part of a language model has definitely empiricists' flavor, it also uses native speakers' intuition manifested in terms of annotation. If annotation includes a set of phrase markers such as noun phrase, verb phrase, relative clause, etc., it already commits to a rationalistic linguistic theory.

The WSJ in LDC claims that their structural annotation uses only the minimum denominators of various linguistic theories and thus does not commit itself to any specific linguistic theories. However, while it may not commit to any specific rationalists' theory and thus reflects theory-independent native intuition, it still relies on human intuition to reveal *non-observables* such as constituent structures, phrase markers to be attached to constituents, etc.

More importantly, the fact that different grammars (CFG, lexicalized CFG, etc.) can be extracted from the same annotated corpora means that we have to consider the roles that linguistic formalisms play in such attempts (Figure 3(a)(b)).

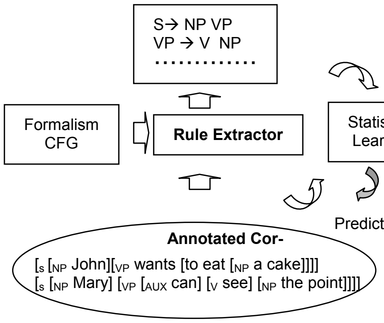
S→ NP VP
VP → V NP
....
Formalism CFG
Rule Extractor
Statistical Learner
Prediction
Annotated Cor-
[$_S$ [$_{NP}$ John][$_{VP}$ wants [to eat [$_{NP}$ a cake]]]]
[$_S$ [$_{NP}$ Mary] [$_{VP}$ [$_{AUX}$ can] [$_V$ see] [$_{NP}$ the point]]]]

Fig. 3(a) Rule Extractor based on CFG

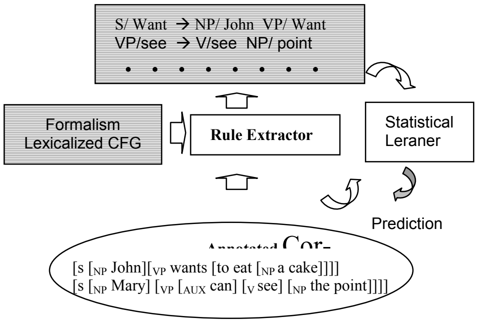
S/ Want → NP/ John VP/ Want
VP/see → V/see NP/ point
• • • • • • • •
Formalism Lexicalized CFG
Rule Extractor
Statistical Leraner
Prediction
Annotated Cor-
[s [$_{NP}$ John][$_{VP}$ wants [to eat [$_{NP}$ a cake]]]]
[s [$_{NP}$ Mary] [$_{VP}$ [$_{AUX}$ can] [$_V$ see] [$_{NP}$ the point]]]]

Fig. 3(b) Rule Extractor based on Lexicalized CFG

As we mentioned in Section 3, by replacing CFG with lexicalized CFG as the *base* formalism, we can get better performance of structural taggers even if we use the same learner for stochastic modeling.

Instead of constructing an undisciplined grammar in CFG or lexicalized CFG, there have been recently a few attempts of constructing grammar in a framework of more linguistically sound grammar formalisms such as LTAG, CCG, etc.[Chiang 00][Xia 01][Hara 02][Clark 02].

These attempts have tried to extract lexicalized information such as elementary trees anchored by specific lexical items or subcategorizations of specific lexical items. Since linguistic theories such as LTAG, HPSG, CCG, etc. have their own built-in ways of interpreting these pieces of information attached to lexical items, these attempts distinguish grammatical knowledge that has to be acquired from corpora from those that are inherently built-in in the frameworks.

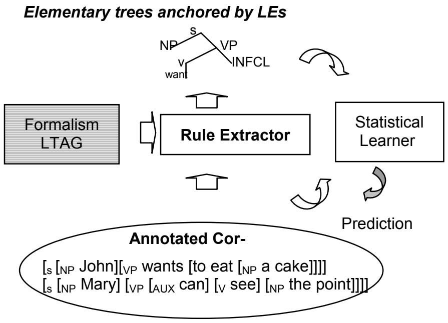
Elementary trees anchored by LEs
S
NP
VP
V
INFCL
want
Formalism
LTAG
Rule Extractor
Statistical
Learner
Prediction
Annotated Cor-
[S [NP John][VP wants [to eat [NP a cake]]]]
[S [NP Mary] [VP [AUX can] [V see] [NP the point]]]]

Fig4. Grammar Learner based on LTAG

A linguistic theory is usually more than a simple monolithic descriptive framework like CFG. It usually is a richer system that consists of several descriptive components. In LTAG, a set of elementary trees anchored by the same lexical item constitute a family and a group of lexical items belong to the same family. To recognize existence of such families is an important step to capturing regularities in lexicon. [Hara 02] implemented a system that discovers such families of words that go through the same paradigms of structural changes.

[Oouchida 03] has gone further by constructing a system that examines structural changes across these families and recognizes the same types of structural changes in different families. That is, his system can recognize structural changes caused by *topicalization*, *passive,* etc. as such, even though different families of verbs have seemingly different sets of elementary trees anchored by them. His system has recognized 132 classes of structural changes, while human linguists at U-Penn recognize 78 of such types.

This line of research shows that we can envisage human (linguist) and machine cooperation that leads to comprehensive description of language by using the same linguistic framework. The descriptions thus produced are transparent in the sense that human linguists can interpret them in terms of his theory.

Note also that the paradigm of structural taggers in Figure 3 and 4 assumes that the performance of a structural tagger is measured in relation with annotations given to test corpora, which are *observables.* This makes NLP research on a par with that of speech recognition in terms of objective and quantitative measures of evaluation.

However, the ultimate objective of NLP research is to relate surface word sequences with their meanings such as in Figure 2. It is, therefore, indispensable in the end to construct and use grammar in full-fledged formalisms such as LTAG, HPSG, etc. that have provisions of linking surface word sequences with their meanings.

5 Grammar Development

It is not easy to develop a comprehensive description of a specific language or grammar with wide coverage. Development of grammar with wide coverage requires both good insights into language and careful observation of actual language usages.

It is natural to expect a computer system to help a linguist to develop grammar with descriptive adequacy. We have actually seen quite a few grammar development workbenches that intended to provide useful facilities for “debugging” grammar [Goetz 97] [Miyata 99] [Schmidt 96]. However, to debug grammar is very different in nature from debugging programs, while most of grammar development workbenches see the two activities as essentially the same.

In case of program debugging, a programmer knows in advance what the program has to do and how it is supposed to do it. In other words, s/he has rational grasp of the task and how to perform it.

On the other hand, since linguistic phenomena are so diverse and vast, a linguist does not have such rational grasp. Rather, s/he has a set of hypotheses on how language works or what the constraints that the language must conform to are, and want to examine whether these hypotheses are correct or not.

Theoretical linguists use intuition or introspection of informants to check the validity of their hypotheses. A linguist in the computational paradigm first formulates her/his hypotheses in a formal framework. Formal description is indispensable since language is such a complex system in which many factors, some of which are structural, interact and it is beyond naïve descriptive devices such as natural language, simple CFG, etc.

Once hypotheses are formulated in terms of a mathematical framework and presented as a whole system, we can see the whole implications of the hypotheses, i.e. how they interact and whether they are consistent, as a whole, with actual language usages.

If a program went wrong, a programmer had to find where the program deviates from her/his rational understanding of a problem. A linguist, on the other hand, has to revise his rational grasp or hypotheses of language more often than debugging their representations in a formal system. The process is more like hypothesis-observation/experiment-revision cycle in other scientific endeavors than mere debugging of formulation of theories and more empirical in nature.

[Yakushiji 2002] tries to capture this aspect of grammar development. In her paradigm (See Figure 5), annotated texts replace linguistic intuition of native speakers in theoretical linguistics. Instead of enumerating sentences or non-sentences on border by intuition, native speakers annotate texts that exist in the real world.

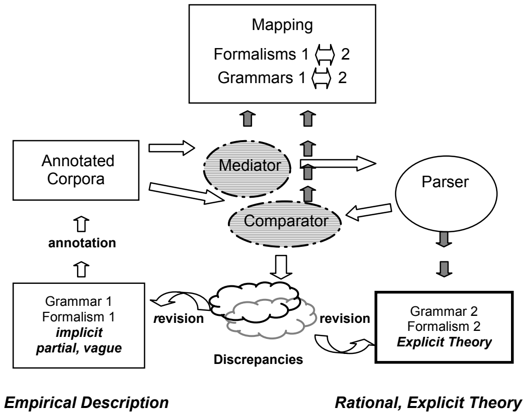

Fig 5 Grammar Development Workbench

Annotation can be performed in terms of, either a naïve concept of language that every native speaker has (like conception of constituent structures of sentences, co-references, etc.) or a theoretical framework, which can be different from the framework in which the linguist is going to develop her/his grammar.

Annotation such as that of WSJ, which is claimed to include only the minimum denominator of various theories and thus be theory neutral, is between these two extremes.

The system she developed has a component in which one can define mapping between the two frameworks, one for annotation and the other for grammar development. By using the mapping, the system can indicate discrepancies between annotations given by human and the descriptions given by the grammar.

A parser is used to assign descriptions to sentences, which in turn are compared with the annotations. The linguist examines the discrepancies and, if necessary, revises her/his theory accordingly. If we could replace the linguistic with a program, the program is one that automatically learns grammar from annotated corpora.

From the very incipient stage of computational linguistics, computer programs have been expected to be used for checking descriptive adequacy of grammar. In the paradigm proposed [Yakushiji 2003], we see natural integration of theoretical linguistics and corpus-based linguistics.

6 Various Types of Equivalence among Grammars

Formal linguists were interested in some forms of equivalence of grammar. The weak equivalence is not interesting since it is only concerned with a set of sentences to be generated by two grammars. The strong equivalence that require derivation processes to be the same is not interesting, either since only trivial grammars can be strongly equivalent.

The meaningful equivalence would be between these two extremes. [Yoshinaga 02] argued that we can weaken the strong equivalence by defining equivalence as “whenever a derivation process in one grammar can be transformed into a derivation process in another grammar and vice versa, two grammars is said to be equivalent”.

[Yoshinaga 2002] showed that the XTAG grammar developed by the group of University of Pennsylvania can be mechanically transformed into HPSG-like grammar, and thereby showed that the XTAG is equivalent to the grammar (we call it XHPSG) constructed from it but that looks like HPSG on surface.

This concept of equivalence based on formal representations of the two is fairy strong. All derivational histories in one grammar have their counter derivations in the other. The two grammars will assign exactly the same number of possible interpretations to any given fragment of linguistic expressions.

By using an annotated corpus, we may be able to define empirically another type of equivalence that is weaker than that defined by [Yoshinaga 02].

Let us consider the three grammars in Figure 3(a)(b) and Figure 4. These three grammars describe the same set of structurally annotated sentences in three different frameworks, i.e. CFG, lexicalized CFG and LTAG. While they may generate the same sentences in the corpus with the same structural descriptions (i.e. annotations attached to the sentences), the sets of sentences defined by them would be difference and even a sentence in the given corpus would be assigned different numbers of possible structures, one (the correct one) of which is shared by all the three grammars.

If we call such equivalence *external equivalence*, to see the relationships between that and Yoshinaga's equivalence will give us insights on relationships among various grammar formalisms and grasp what are the common conceptions shared by different formalisms, what the differences and their external consequences are.

Such, more fine-grained concepts of equivalence will be of much use, compared with the two extremes of equivalence

7 Parsing by Linguistically Sound Formalisms

One of the difficulties of using linguistically sound grammar formalisms for parsing is its inefficiency and combinatorial nature of ambiguities of natural language. Some techniques that have been investigated in these several years have improved the efficiency of parsing based on unification-based grammar formalisms significantly [Oepen 01].

However, the problem caused by the combinatorial nature of ambiguities still remains. This is particularly the case when we introduce the semantic layer of representation in our grammar, since the introduction multiplies semantic ambiguities with syntactic ones.

As we saw in Section 3, the same constituent structures often have more than one semantic representation, which results in combinatorial results of parsing. While we can reduce such explosion of ambiguity by using semantic constraints, semantic constraints are notoriously elusive. If we apply semantic constraints as actual constraints, a system becomes fragile since semantically ill-formed sentences often appear in real texts due to common usages of metaphors, metonyms, etc.

Here we encounter a perennial problem of NLP, i.e. language cannot be treated properly without semantics but introduction of semantics makes either a system very fragile or computationally intractable because of combinatorial explosions in parsing.

The relative success of lexicalized CFG over simple CFG shows that semantics will be indispensable in more sophisticated stochastic models for structural taggers, but this perennial difficulty involved in semantics has prevented us from using richer linguistic model.

However, as in Section 3, [Miyao 02] shows how to efficiently construct a stochastic model for unification-based grammar, and [Sakao 03] recently discusses how semantic representation in unification-based grammar can be incorporated in the stochastic models and how one can use it in controlling the process of parsing. The key of his method is how to deal with the non-local nature of semantic representation in a constructive computation of values of preference that are used to prune less plausible parsing paths.

Since Sakao's model uses semantics as preferences based on a stochastic model, it is much more robust than those that use semantics as constraints. At the same time, it resolves the difficulties caused by combinatorial explosion by using stochastic semantic model to prune implausible parsing paths.

A preliminary experiment is being conducted with the ATIS corpus, and it shows that, by using semantics as preferences in Sakao's model, a parser for unification grammar produces all the correct interpretations (thus robust enough) and at the same time reduces possible interpretations significantly (60% of interpretations are disregarded as semantically less preferred).

8 Conclusion

Since Chomsky's manifesto of theoretical linguistics, there have not been much serious attempts in unifying empirical descriptive linguistics with rationalistic theoretical linguistics. Instead, researchers in the two camps seem to have antagonistic sentiment against each other.

However, because of huge amount of texts available in networks and WEB and because of significant progress in symbolic grammar formalisms together with techniques in statistical language modeling and machine learning, we now see the glimpse of hope for possible unification of the two streams of language study.

Language now becomes like real physical existence. Enormous amount of its actual uses are recorded and stored in computer. We can manipulate them and measure their properties by computer programs. On the other hand, due to progresses of grammar formalisms and parsing techniques, we can describe our rationalistic theories or hypotheses on language usages in these frameworks and try to describe actual language uses in terms of the theories, again by using computer programs. That is, we can check whether there are discrepancies between theories and actual language uses.

In this paradigm, the study of language become like other science to repeat the cycle of hypotheses-building, experiments, revision of hypotheses. There are significant differences between the initial manifestos of theoretical linguistics by Chomsky and the paradigm we envisage. For example, our paradigm seems to be more empirical than theoretical linguistics a la Chomsky in the sense that we use actual language uses as they are and there will not be strong distinction of *competence* and *performance*.

It is our contention in this paper that, due to progresses in computer science/technology, formal methods in linguistics, and statistical modeling and machine learning, we can envisage a more robust paradigm in language study, which has both empirical and rationalistic aspects of linguistics so far.

Acknowledgement

I am very grateful to the members, Ph.D and MSc candidates of the NLP research laboratory of the University of Tokyo. The lively discussions I have had with them are the basis of this paper.

References

[Chiang 00] D.Chiang, et.al.: Statistical Parsing with an Automatically-extracted Tree Adjoining Grammar, in Proc. of ACL00, 2000.

[Goetz 97] T.Goetz, et.al: The ConTroll system as large grammar development platform, in Proc. of Workshop on Computational Environments for Grammar Development and Linguistic Engineering, pp38-45, 1997.

[Hara 02] T.Hara: Clustering for Obtaining Syntactic Roles of Words from LTAG Grammar, BSc Dissertation, Dept. of CS, University of Tokyo, 2002.

[Johnson 02] M.Johnson, et.al.: Dynamic Programming for Parsing and Estimation of Stochastic Unification-based Grammar, in Proc. of ACL02, 2002.

[Miyao 02] Y.Miyao, et.al.: Maximum Entropy Estimation for Feature Forests, in Proc. of HLT02, 2002.

[Miyata 99] T.Miyata, et.al.: Implementation of GUI Debugger for Unification-based Grammar, in SIG Notes NL-129, pp87-94, 1999.

[Oepen 01] S.Oepen, et.al.: Collaborative Language Engineering: A Case Study in Efficient Grammar-based Processing, CSLI publication, Stanford Univ., 2001.

[Oouchida 03] K.Oouchida: Using Constructional Features for Classifying Lexical Entries in Lexicalized Grammar, BSc Dissertation, Dept. of CS, Univ. of Tokyo, 2003.

[Sakao 03] Y.Sakao: Constructive Calculation of Nonlocal Figures of Merits for Parsing, MSc dissertation, Dept. of CS, Univ. of Tokyo, 2003.

[Schmidt 96] P.Schmidt, et.al.: Lean Formalisms, Linguistic Theory and Applications, Grammar Development in ALEP, in Proc. of Coling 96, pp286-291, 1996.

[Xia 01] F.Xia: Automatic Grammar Generation from two Perspectives, Ph.D. Thesis, Univ. of Pennsylvania, 2001.

[Yakushiji 03] A.Yakushiji: A Workbench of Practical Grammar Development for Information Extraction, MSc Thesis, Dept. Of CS, The University of Tokyo, 2003.