A Comparative Analysis of Extracted Grammars

A Comparative Analysis of Extracted Grammars

Alessandro Mazzei and **Vincenzo Lombardo** [1]

Abstract. The development of wide-coverage grammars is at the core of robust NLP systems. This paper addresses the problem of grammar extraction from treebanks with respect to the issue of broad coverage along three dimensions: the grammar formalism (context-free grammar, dependency grammar, lexicalized tree adjoining grammar), the domain of the annotated corpus (press reports, civil law) and the language of the corpus (English, Korean, Chinese, Italian). We have extracted three grammars from an annotated corpus of Italian and we have comparatively analyzed the coverage of a test set; then, working on two different domain subcorpora we have compared the cross-domain coverage of the extracted grammars; finally, we have compared the grammars for four different languages. The results are that there are relevant differences in coverage among formalisms and domains; a more limited difference appears in the cross-linguistic comparison.

1 Introduction

The development of wide-coverage linguistic resources is at the core of robust NLP systems. The case of grammars is particularly relevant because most NLP systems use some form of syntax as the primary linguistic knowledge. After the advent of annotated corpora, the most effective way to build wide-coverage grammars is to extract them from treebanks. This paper addresses the problem of grammar extraction with respect to the issue of broad coverage along three dimensions: the grammar formalism, the domain of the annotated corpus and the language of the corpus. The coverage task allows to concentrate on the grammar contribution to a corpus-based approach, since we can get rid of the influences of the stochastic model to the parsing performance.

The grammar formalism represents the way in which the grammatical knowledge is encoded. There exist dozens of formalisms in the theoretical and computational linguistics literature, some of which have been employed in the corpus-based approaches. The ubiquitous context-free phrase structure grammars form the baseline against which any other formalism has to be compared in performance. In this paper we have chosen two other formalisms: a dependency grammar written in the style of [6] and a lexicalized tree adjoining grammar (LTAG – [9]). Dependency grammars have been employed in a number of NLP applications because of the significance of the syntactic relations that they represent [11]. Also, the treebank we have used in our tests is annotated in a dependency format. So, it is easy to extract a grammar in a format which is similar to the annotation schema. Both dependency relations and phrase structure are represented in LTAG, in the derivation tree and the derived tree, respectively (see below). LTAG is a well known formalism that has been deeply investigated in mathematical and computational aspects, and has also received attention in corpus-based approaches. In the context of grammar extraction, LTAG has been the target formalism in an extraction and comparison experiment in a cross-linguistic setting, which involved an English, a Chinese and a Korean treebanks [14]. The results yielded there will be compared with our results on Italian (see below).

The domain axis gives an interesting perspective on the portability of the grammars extracted. Since the corpus annotation is a very laborious and time-consuming task, the applicability of a grammar extracted from some corpus to another corpus is a desirable property. There are a few experiments in the literature on tasks that involve a training corpus which is different from the test corpus. The difference in corpus can be considered in either the task of grammar coverage of the test set or the task of stochastic modeling of the test set [2]. A different corpus usually leads to a different distribution of sentences in various domains; the differences in parsing performances can be due to either undergeneration of the grammar or inadequacy of the stochastic model. A few works have appeared in the context of statistical parsing[2] Sekine [12] has conducted several parsing trials on the Brown corpus by considering nine text categories classified under the fiction/non fiction dichotomy. The results were that parsing achieves better performance (of up to 5% in precision and recall) when trained on the same or similar domain of the test sentences, since the domain affects the syntactic structure distribution because of the presence of idiosyncratic structures. Similar results were found by Gildea [7] in passing from the Wall Street Journal to the Brown corpus. In the task of grammar coverage, Black et al. [1] have found a 4% of failures when applying a grammar manually developed for a restricted domain (computer manuals) using a "treebanking" method to previously unseen text. In this paper we compare the grammar coverage in two different domains, one of press reports from several Italian newspapers and one of the civil law.

Finally, the last comparing dimension is the language. The possibility of porting a grammar from one language to another is relevant on both theoretical (a sort of quantitative testing of the Universal Grammar Hypothesis) and engineering (extracted grammars as resources for a number of NLP tasks) grounds [14]. In order to compare the grammars extracted from the treebanks of different languages we must at least share the formalism in which the grammar rules are encoded. Then, we must abstract from the mere grammar rules in order to find the sources of variation in terms of more general linguistic parameters, like word order, pro-drop, tag sets, syntactic relations. In this paper, we extend the results of [14] to Italian. We employ the extracted LTAG grammar and we compare the templates of the grammar for four languages (English, Korean, Chinese, Italian).

After a brief description of the treebank we use in our experiments, we introduce the three grammar formalisms together with the

[1] Dipartimento di Informatica, Università di Torino, Torino, Italy, email:{mazzei,vincenzo}@di.unito.it

[2] It has to be noticed, in fact, that most parsing approaches are trained and tested on the same corpus domain, e.g. the Wall Street Journal ([5], [4]).

extraction methods, and we analyze the coverage issue for all the formalisms. Then, we address the corpus domains, and finally the cross-language comparison.

2 The Turin University Treebank

The Turin University Treebank (TUT) is an on-going project of the University of Turin on the construction of a dependency style treebank for Italian [3]: each sentence is semi-automatically annotated with dependency relations that form a tree, includes traces for achieving a greater clarity in the representation of predicate-argument structures, and relations are of morphological, syntactic and semantic types. Its current size is 1500 annotated sentences (=33,868 words), although in this work we report data on 1235 sentences (=32,221 words). The original corpus is very varied, and contains texts from two major domains, a collection from the civil law (500 sentences) and a collection of press reports on the theme of Albanians in Italy (600 sentences). The rest (400 sentences) is a miscellanous of newspaper articles, novels, press agency news. In figure 1 is a tree from the TUT corpus. The nodes of the tree are the words of the sentence and some trace (lexically empty) symbols; the edge are labelled with the functional relations between the words. The sentence is: "In quegli istituti finanziari forse sono andati a finire quei soldi, ha aggiunto." ("In those agencies financial maybe have gone to finish those money, has added." word-by-word translation, "Maybe that money has gone to those financial agencies, he added."). It is a pro-drop sentence (see the empty node "&"). The main verb is "aggiunto" (added), with one object clause rooted by "andati" (gone). Also the embedded has a verbal modifier "a finire" (to finish) with an equi control on the subject ("quei soldi" – that money). "Arg" is a generic label for an argument, and "rmod" is a restrictive modifier. In this tree we can find three predicate-argument structures: one transitive structure (subj,obj) rooted by "aggiunto", one (subj,loc) rooted by "andati", one (subj) rooted by "finire". Although the treebank has a small size

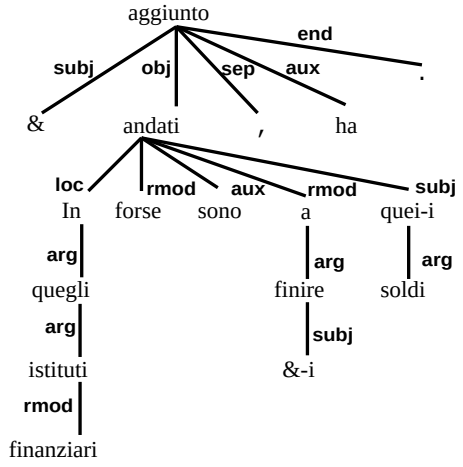

Figure 1. A sample tree from the Turin University Treebank.

with respect to the largest treebanks (e.g. the Penn Treebank or the Prague Dependency Treebank), previous literature has shown that a reduced size is sufficient to yield performances comparable with relevant results in coverage [14] or, in some cases, parsing (see e.g. [12]).

3 The grammar formalism

In this section we introduce the three formalisms that we have used to extract the grammar from the treebank. We describe here the basic issues of each formalism and the algorithms used to extract the corresponding grammar from the TUT corpus. Then we see what is the coverage of the grammar in the corpus.

3.1 Dependency grammar

Since the TUT corpus relies on a dependency-based annotation, the extraction of a dependency grammar is immediate, especially in a simple formalism like the context-free weakly equivalent Gaifman's system [6].
Dependency theories have a long tradition especially in the encoding of free word order languages [8] [11] [13], and have been used extensively in applicative tasks in various forms. The choice of a dependency format for an Italian treebank depends on its property of partial configurationality, that assigns a percentage of more than 25% to orders of verbal arguments other than Subject-Verb-Complement. The basic tenet of dependency formalisms is that the syntactic structure is given by relations over the words in the sentence; these relations can be labelled (as in the TUT corpus) or unlabelled (as in the Gaifman's system). The rules of Gaifman's system are in the format

$$X \quad \rightarrow \quad Y_1 Y_2 ... Y_{i-1} * Y_{i+1} ... Y_n$$

where X and Y_j are POS tag categories, X is the head of the rule and Y_j are the dependents, and $*$ is the position of the head in the linear order of the dependents. Notice that such rules are actually template rules, where words are replaced by POS tags. This is not a limitation, since in all the applications the extracted grammars are usually augmented with respect to the lexicon and stochastic models include smoothing techniques to recover situations due to the sparseness of lexical data distributions. Also the comparisons in [14] are based on templates. The extraction algorithm for Gaifman dependency grammars is a recursive procedure descending the dependency tree. Starting from the root node, for each node of the tree we add a generative rule that has the POS tag of the node on the left hand side of the rule (the head). Following the linear order of the sentence, on the right side of the rule there are the POS tags of the children, by indicating the position of the lexical realization of the head with an asterisk. Here are two sample rules extracted from the tree in figure 1.

V $\rightarrow$ PRO V PUN $*$ V PUN
PRO $\rightarrow$ $*$

The first is the top rule of the dependency tree, that is clearer when we add the dependency relation labels. In fact, the variant of the algorithm allows us to include the relation that labels the edge from a node to its father. In this version of the dependency grammar (DG-r) the two sample rules become:

V() $\rightarrow$ PRO(subj) V(obj) PUN(sep) $*$ V(aux) PUN(end)
PRO(subj) $\rightarrow$ $*$

The second is the realization of the empty pro-drop subject.

3.2 Lexicalized Tree Adjoining Grammar

Lexicalized Tree Adjoining Grammars (LTAG [9]) is a well known formalism, with a vast literature on its mathematical properties, lin-

guistic relevance, wide coverage grammar development, statistical parsing. A LTAG consists of elementary trees (instead of rules) that are combined through substitution and adjunction to form syntactic trees. Elementary trees can be initial (argumental) trees or auxiliary (modifier) trees. Each elementary tree is associated with a lexical anchor. In initial trees the lexical anchor determines the predicate-argument structure, where arguments are specified by substitution nodes (this property is named *extended domain of locality*). Auxiliary trees factorize the recursive structures by including the minimal recursion between the *root* and the *foot* nodes. The final syntactic structure for a sentence is a *derived tree* and the history of combination of the elementary trees is in the *derivation tree*. As in the dependency grammar above we deal with template elementary trees, where the lexical anchor has been removed. In figure 2 there is a derived tree for the sentence in figure 1.

```
(S (S (NP (PRO &))
      (VP (S (S (NP &-QUELLO-1)
                (VP (PP (P IN)
                        (NP (ADJP (ADJ QUELLO))
                            (N1 (N1 (N ISTITUTO))
                                (ADJP (ADJ FINANZIARIO)))))
                    (VP (ADVP (ADV FORSE))
                        (VP (V ESSERE)
                            (VP (VP (V ANDARE))
                                (PP (P A)
                                    (S (NP &-QUELLO-1)
                                       (VP
                                         (V FINIRE)))))))))
             (NP (ADJP (ADJ QUELLO-1))
                 (N1 (N SOLDO))))
          (VP (PUN #\,)
              (VP (V AVERE)
                  (VP (V AGGIUNGERE))))))
   (PUN #\.))
```

Figure 2. Constituency (derived) tree for the TUT tree in fi gure 1.

In order to extract the LTAG grammar, we have converted the TUT treebank dependency format to a constituency format, and then we have adapted the algorithm in [14], that was applied to the Penn treebank. The algorithm that converts the dependency annotation in constituency annotation has two stages. In the first stage it builds a binary constituency tree: starting from the top node of the dependency tree, it incrementally creates a constituency tree by introducing new unlabelled nodes in a right-branching structure. In the second stage, each unlabelled node of the constituency tree is labelled on the basis of its daughters' labels: starting from the frontier nodes the algorithm climbs bottom-up the constituent tree, and assigns a label to an unlabelled node using some heuristic rules about the labels of its daughters (the most relevant rule concerns the label of the head daughters).

The extraction algorithm is more complex, because of the rich (morphological and semantic) annotation in the TUT format. The conversion from dependency to constituency trees exploits the dependency relation that originated a constituency link. In the extraction procedure we can use these relations to recognize if a daughter node is the root of an argument subtree, the root of a modifier subtree, or the head daughter. We use a recursive "cut" procedure to extract the elementary trees that can generate that sentence. First we identify the elementary tree anchored by the head of the root node; then we call the procedure on the nodes that are the maximal projection of the heads of the arguments, obtaining other initial trees; finally, we call the procedure on the nodes that are the maximal projections of the heads of the modifier, obtaining auxiliary trees. In figure 3 there are three templates of elementary trees, two auxiliary trees (for verb phrases and nbar respectively) and one initial tree (for the predicate-argument structure that includes a pro-drop subject and an object subordinate clause). The details of this algorithm are in [10].

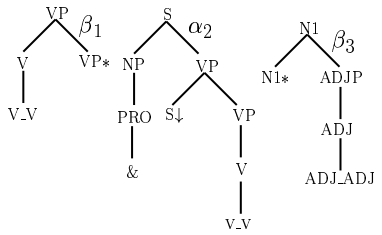

Figure 3. Templates of elementary trees extracted from the TUT corpus. The repeated category in a leaf correspond to the lexical anchor node.

3.3 Context-free grammar

Context-free grammars are the most widespread formalism in corpus-based approaches. This is because of its long tradition in the mainstream of computational linguistics, its desirable computational properties, and the number of corpora annotated in a phrase structure approach. The extraction of the context-free grammar relies on the first step in the extraction of the LTAG, when the dependency tree has been converted into a constituency tree (see below and figure 2). To extract the context free grammar from the constituency trees, we used a standard procedure (cf. [4]). Starting from the top node of the tree we define a context free rule that has the tag of the node on the left hand side of the rule, and the children of that node on the right hand side of the rule. Here are a few rules extracted from the tree in figure 2.

NP	→	ADJP N1
N1	→	N1 ADJP
NP	→	PRO
VP	→	S VP
S	→	S PUN

The first three rules represent the internal (recursive) structure of the NP, including one rule for the pro-drop subject (the third). The fourth rule represent a verb phrase that governs an embedded object clause that precedes the verb. The last is the top rule of the tree.

4 Coverage measures

In this section we report on the coverage of the extracted grammars. For each formalism, we have split the TUT corpus in two sets, and we have extracted the grammar from the first set (training set); then we have evaluated the coverage of the extracted grammar on the second set (test set). We have used the entire corpus of 1235 sentences ($32,221$ words). As mentioned above, in order to avoid the problem of the sparseness of data caused by the use of lexical items, we have limited our grammar rules to templates that only include the POS tag. The number of templates we have extracted were 995 (408 initial trees and 587 auxiliary trees) in the case of LTAG; 262 context-free rules, $2,222$ unlabelled dependency rules and $4,212$ labelled

dependency rules. In the case of LTAG, the limitation to the POS tag has produced a strong reduction of templates (from $8,423$ to 995); a very few variations occurred in the context-free and the dependency grammars. The very high number of dependency rules is due to the inclusion of both arguments and modifiers in a single rule: this has caused a great variation of patterns, with very sparse data.

Then we have performed several experiments to assess the coverage of the extracted grammars, by increasing the ratio between the test set and the training set. The goal is to see how the coverage of the extracted grammar improves with the size of the training corpus. Table 1 reports the results of three trials, where the size of the test set was 10% 20% and 50% of the size of the entire corpus; correspondingly, the learning set was 90%, 80% and 50%[3].

	10%	20%	50%
LTAG	91.3%	89.9%	85.8%
CFG	97.3%	96.8%	94.6%
DG	29.7%	27.1%	24.2%
DG-r	13.7%	12.9%	11.7%

Table 1. Coverage results on the complete corpus

As one can see from the data, the CFGs have the highest coverage. Afterwards, growing the size of the test set the coverage of the context free grammars changes slightly. This result suggests that the structural information contained in the whole corpus can be easily learned by CFGs, also using only a reduced subset of the corpus. The extracted LTAG is more dependent by the size of the learning set with respect to CFG. An analysis of the more frequent verbal elementary trees are transitive verb subcategorization frame, the same frame with a pro-drop subject, and the auxiliary tree for adjoining an auxiliary verb on the left, respectively, thus confirming the linguistically motivated analysis of the results. The two types of dependency grammars show a very low coverage performance. This is not surprising, since the Gaifman-style dependency grammars are strongly related to the number of arguments and modifiers of the head word, so that different rules are needed for analyzing two substructures which are the same except for the modifiers. This does not happen for CFG and LTAG, where modifiers are handled via the repeated application of recursive rules, a mechanism that can easily produce overgeneration (no constraints on the placement of modifiers). This result evidences that to use these type of dependency grammar, we need a strategy to avoid the sparseness of the data.

5 Cross-domain comparison

In this section, we compare the grammars extracted from two different subcorpora, i.e. the civil law and the press reports collections. In this second set of experiments we test the dependency of the extracted grammars on the "type" (domain, genre) of the corpus. In the last years several works [12],[7] have considered the parsing performances of a probabilistic context free grammar extracted from a treebank. In particular [12] showed that if the test set is of the same genre of the training set, better parsing performances are obtained. The experiments reported in the following broaden this perspective by using several grammatical formalisms, and are based on the task of coverage, which is more independent of the particular language model. In particular we have considered two subsections of the TUT corpus, containing sentences of two well defined domains. The first domain includes 402 sentences (9232 words), and consists in a section of the Italian civil code (civil-code); the second domain (517 sentences, $13,092$ words) is a set of articles appeared on an Italian newspapers (newspaper-articles), concerning a particular topic. We first extract from each subcorpus the LTAG, CFG, and dependency grammars, and then we compare the coverage results, using different ratios of training set and test set. In the tables 2 and 3 we see the results of this experiment.

	10%	20%	50%
LTAG	88.3%	86.2%	74.3%
CFG	93.7%	94.3%	86.4%
DG	41.5%	39.3%	35.4%
DG-r	26.8%	22.7%	21.3%

Table 2. Coverage results on the civil law subcorpus.

	10%	20%	50%
LTAG	85%	83.3%	77.1%
CF	93.1%	93.7%	90.3%
DG	21.15%	19.8%	15.8%
DG-r	5.7%	5.8%	4.8%

Table 3. Coverage results on the newspaper-articles subcorpus.

The results on LTAG and CFG substantially replicate the data of the entire corpus (table 1). It also occurs an idiosyncratic phenomenon that using a 80% training set performances are better than 90% training set for CFG. However, the results for the dependency grammars show a great variance and are mostly better for the civil law subcorpus. Presumably, the much better results on the grammar extracted from the civil-code subcorpus reflect a greater regularity of the expressions used in the legal language.

In order to obtain deeper insights on the differnce of the subcorpora, we designed a second experiment, where the crossing coverage is computed. In other words we used one of the subcorpora as the learning set and the other as the test set (so, we use all the sentences in each subcorpus as the test set). The results are reported in Table 4. We can see that the grammars extracted from the civil law corpus perform better than the grammars extracted from the newspaper corpus. This is evident in the cases of the LTAG and CFG.

Learning Set	Test Set	LTAG	CF	DG	DGr
newspapers	civil-law	46%	65.4%	11%	3.7%
civil-law	newspapers	66%	80.7%	11%	2.1%

Table 4. Crossing coverage results for newspaper-articles and civil-code subcorpora.

The LTAG extracted from the civil-law counts 439 templates, while the LTAG extracted from the newspapers-articles has 567 templates. So, the latter grammar is larger than the first (also consider that the newspaper subcorpus is larger), but covers a minor portion

[3] These values are the means over 5 random trials.

of the civil code corpus. In the case of the context free grammars, we have roughly the same sizes (207 rules from the civil code against 216 rules from the newspapers corpora, respectively), and the coverage results are the same. The reduced number of rules is probably due to the greater factorization or domain of locality of CFG with respect to LTAG.

Since the coverage task is applied by crossing the domains of the grammar and the test set respectively, it occurs that it is the common set of rules between the two grammars extracted from the subcorpora that is responsible for the coverage. In the case of the LTAGs we have 245 common templates. Notice that this is slightly more than the half in the case of the civil law grammar, and much less than the half in the case of the newspaper grammar. This means that the language used in the newspaper corpus adheres better to this common set of rules than the language of the civil law corpus. So, the language in the newspaper corpus is mostly covered by a reduced number of rules but does not contain some rules that are necessary for parsing the civil code corpus. corpus has a major degree of regularity extraction of a more portable grammar.

6 Cross-linguistic comparison

In [14] english, korean and chinese are compared through some LTAGs automatically extracted from treebans. Using the treebanks developed at Penn University for those languages, several linguistic motivated comparing were performed. Since we have not access to the LTAGs extracted by Xia et al., we can only compare the size of our grammar with respect to the size reported in that work. As noted in [14], one has to normalize the size of the tag set between various treebanks to guarantee a more precise comparing. In fact, with much different tag sets we would detect different templates because some tags differ for minor features (e.g. different types of verb forms in the Penn treebank). The smaller tagset of the TUT with respect to the other treebanks depends by the originally dependency annotation schema. Some information that in the Penn treebank is usually annotated in the tag (e.g. the subject), was directly annotated with an arc in the dependency tree. We do not know the exact size of the tag set used by [14], and so we cannot perform an accurate analysis of the data, but we can note some divergences and some similarities. We can see in table 5 that the number of templates extracted from the Italian treebank is greater than the number of templates for Korean treebank (which is of a larger size), also when it has the original tag set. This difference can be partially explained with the type of the sentences belonging to Korean treebank. In fact, that corpus is a military guide annotated (http://www.cis.upenn.edu/ xtag/koreantag/), and contains many sentences about dialogues, with many one-word answer[4]. A similar argument can also explain the relative small difference in the number of templates for Italian and Chinese, since the latter corpus consisted of a set of news messages appeared on a Chinese web-wire [15]. We plan to verify these analyses in future works.

7 Conclusion

This paper has presented a comparison over grammars extracted from an annotated corpus. We have extracted three different types of grammars, a context free grammar, a dependency grammar and a lexicalized tree adjoining grammar. The grammars were tested against a broad coverage test, also by working on different domain subcorpora and cross languages. The results were significant for the context-free and the LTAG grammars, while poor performances were achieved with the dependency grammars. We have also seen that some domains are better than others in allowing the extraction of grammars that are more portable. This has been the case of the civil law subcorpus, that performed better than the newspaper subcorpus. These results can be considered a first attempt in characterizing the usefulness of corpora in producing generalizable results: we think of applying these techniques to the Brown corpus that already provides a separation of genres, in order to test portability on a large scale.

[4] This is the reason for the very low average sentence length.

Language	Corpus size	Average sentence length	Tagset size	Number of templates	Number of context free rules
English	1,174K	23.85	94	6926	1524
English	1,174K	23.85	?	3139	754
Chinese	100K	23.81	92	1140	515
Chinese	100K	23.81	?	547	290
Korean	54K	10.71	61	632	152
Korean	54K	10.71	?	256	102
Italian	42K	25.77	27	995	262

Table 5. Cross-linguistic comparison of LTAG data.

ACKNOWLEDGEMENTS

We would like to thank Raffaella Ventaglio, who implemented the basic version of the LTAG extraction algorithm, Leonardo Lesmo and Cristina Bosco for many interesting comments.

REFERENCES

[1] E. Black, R. Garside, and G. Leech, *Statistically-driven computer grammars of English: The IBM/Lancaster approach*, Rodopi, Amsterdam, The Netherlands., 1993.

[2] E. Black, J. Lafferty, and S. Roukos, 'Development and evaluation of a broad-coverage probabilistic grammar of english-language computer manuals', in *Proc. of ACL*, pp. 185–192, (1992).

[3] C. Bosco, *A Grammatical Relation System for Treebank Annotation*, Ph.D. dissertation, Dipartimento di Informatica, Università di Torino, 2004.

[4] E. Charniak, 'Statistical parsing with a context-free grammar and word statistics', in *Proc. of AAAI97*, pp. 598–603, (1997).

[5] M. Collins, 'A new statistical parser based on bigram lexical dependencies', in *Proc. of 34th ACL*, pp. 184–191, (1996).

[6] H. Gaifman, 'Dependency systems and phrase-structure systems', *Information and Control*, **8**(1), 304–337, (1965).

[7] D. Gildea, 'Corpus variation and parser performance', in *Proc. of the EMNLP 01*, pp. 167–202, (2001).

[8] R. Hudson, *English Word Grammar*, B. Blackwell, Oxford, UK, 1990.

[9] A. Joshi and Y. Schabes, 'Tree-adjoining grammars', in *Handbook of Formal Languages*, eds., G. Rozenberg and A. Salomaa, Springer, (1997).

[10] A. Mazzei and V. Lombardo, 'Building a large grammar for italian', in *Proc. of LREC04*, (2004).

[11] I. Mel'cuk, *Dependency syntax : theory and practice*, State University Press of New York, 1987.

[12] S. Sekine, 'The domain dependence of parsing', in *Proc. of ANLP 97*, pp. 185–192, (1997).

[13] P. Sgall, E. Haijcova, and J. Panevova, *The meaning of the sentence in its pragmatic aspects*, Reidel Publishing Company, 1986.

[14] F. Xia, C. Han, M. Palmer, and A. Joshi, 'Automatically extracting and comparing lexicalized grammars for different languages', in *Proc. of IJCAI 01*, pp. 1321–1330, (2001).

[15] N. Xue, F. Chiou, and M. Palmer, 'Building a large-scale annotated chinese corpus', in *Proc. of COLING 02*, (2002).