Induction of Treebank-Aligned Lexical Resources

Tejaswini Deoskar
Dept. of Linguistics
Cornell University
td72@cornell.edu

Mats Rooth
Dept. of Linguistics and CIS
Cornell University
mr249@cornell.edu

Introduction

By 'treebank-aligned lexical resources' we mean ones where there is a systematic correspondence between the lexical resource and treebank syntactic resources. For instance, the lexicon resource contains features representing the subcategorization frames of verbs, which correspond to structural configurations that the verb occurs in, in a treebank. Given such an alignment, a treebank can be compiled into a lexicon by collecting the combinations of lexical entries and their local features which are found in the treebank. This paper focuses on the problem of creating and evaluating lexicons for PCFGs which encode lexical features such as verbal valence (subcategorization) in pre-terminal symbols [1]. We use a method based on constraint solving (similar to the LFG framework described by O'Donovan et al. (2005)), to add feature annotations to the Penn Treebank of English (Marcus et al., 1993). Features are then incorporated in the symbols of a context free grammar and frequencies are collected, resulting in a probabilistic grammar and a probabilistic lexicon which encodes lexical features.

Previous research has argued that because of sparseness of lexical distributions, computational lexicons derived from corpora should be based on very large corpus samples, much larger than the roughly 50,000-sentence Penn Treebank (Briscoe & Carroll, 1997). Beil et al. (1999); im Walde (2002) demonstrated that PCFG grammars and lexicons with incorporated valence features could be improved by iterative EM estimation; however their grammar was not a treebank grammar, and therefore could not be evaluated using standardized evaluation criteria. Our treebank-aligned grammar and lexicon allows us to evaluate lexical learning using a held-out portion of the treebank for testing. On the task of identifying the valence of token occurrences of novel verbs, we get a 24%reduction in errors rate following a standard inside-outside estimation procedure (Lari & Young, 1990). A modified inside-outside procedure which re-estimated lexical parameters while retaining syntactic parameters in the PCFG gives a reduction in error rate of better than 41%.

In the sections to follow, we first describe our methodology for augmenting treebanks, and then the procedure to re-estimate the treebank-aligned lexicon using unannotated data.

Treebank Feature Augmentation

Our methodology for augmenting the treebank with features involves parsing with a feature-constraint grammar whose backbone is the context-free grammar obtained from the treebank. The feature constraint annotations are similar to those used in LFG frameworks like O'Donovan et al. (2005)– however our goal is to realize a PCFG in the end. In the first step, for each sentence in the treebank, a shared forest data structure is constructed. This forest represents the set of trees licenced by the context-free backbone grammar whose yield is the sentence. In the second stage, constraints are solved in the shared forest. For solving constraints, we use the feature grammar parser Yap (Schmid, 2000). This stage adds features and may split a tree into several solutions. In order to realize the second step, we

[1] While we focus on verbal valence, there are many such lexically oriented features, such as attachment preferences of adverbs (nominal, verbal, sentential), the valence of nouns, classes of adjectives, etc.

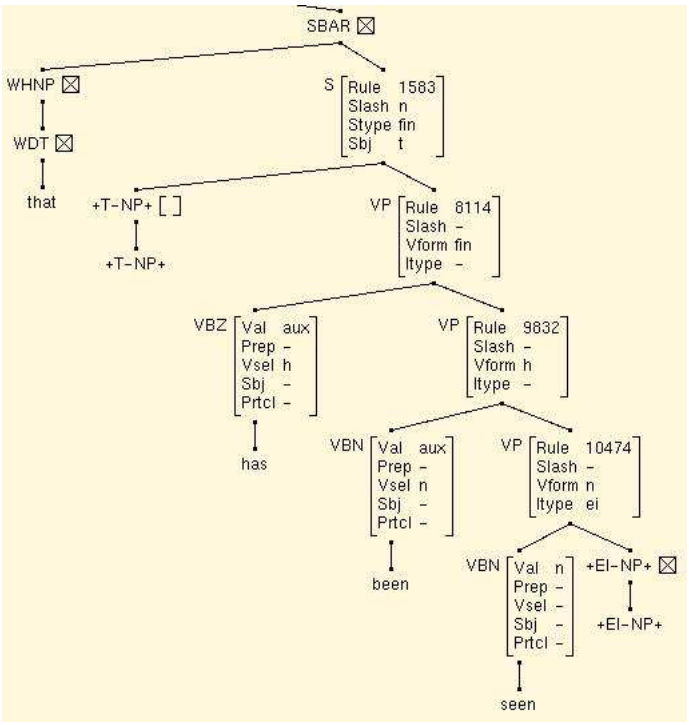

Figure 1: A relative clause in the transformed treebank: Empty categories are flanked by plus signs.

build a feature constraint grammar whose context free backbone is extracted from local tree configurations in the treebank. We automatically add feature constraints rules (following the Yap formalism) to the treebank grammar rule using programs written in Perl and Lisp. Adding these constraints requires checking treebank conventions and exploiting regular patterns in rule shapes. For instance, in the treebank convention, any local tree with a VP parent and VP and verb children is an auxiliary verb construction, so the constraint "Val=aux;" identifying an auxiliary verb may be placed on the verb in the corresponding rule. Below is an example of a feature constraint rule for an auxiliary construction. The VP has a Vform feature which marks finite or infinite VPs, among other things. The Slash feature propagates from the daughter VP to the mother VP with a variable *sl*. The Vform of the complement VP is assigned to the auxiliary verb using a variable *vf*, the auxiliary verb is also marked with the valence feature *aux*. The Prep (preposition) and Sbj (subject) features on the verb have a default value in this rule – Prep would get a value if the verb had a PP complement, and Sbj would be marked with the subject of an S complement. Figure 1 and 2 show sample trees in the transformed treebank.

```
VP {Vform=base; Slash=sl;} ->
                  'VB {Val=aux;Prep=-;Vsel=vf;Sbj=-;}
                   ADVP {}
                   VP {Slash=sl;Vform=vf;};
```

We have a number of such constraints that are linguistically motivated and take advantage of information in the treebank. For example, there are features which constrain the distribution of common empty categories, using a standard slash mechanism for long-distance dependencies. Dependencies such as passive and raising are constrained with local features such as Vform and Vsel, and are in effect lexicalized. Other examples of features are those which mark temporal or locative nouns, a valence feature on nouns marking if the complement of the noun is an S, SBAR or PP category, and another feature marking the preposition of the PP complement on the noun. We also have other features which are tree-geometric but not linguistic in nature (in the style of Johnson (1998); Klein & Manning (2003))- they are relevant to producing a good PCFG model and are not described here.

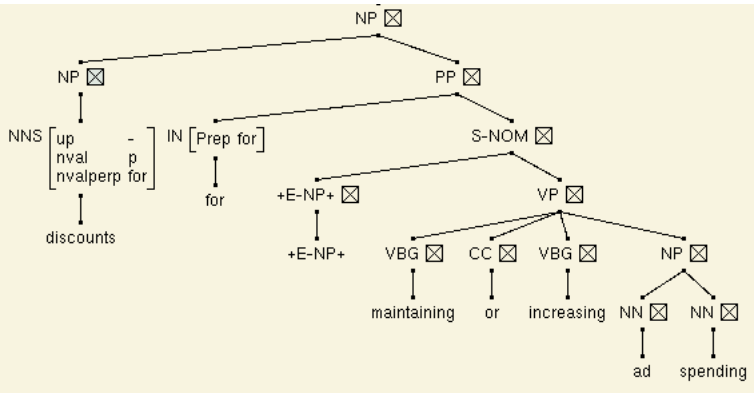

Figure 2: Prepositional complements of nouns are marked on the noun *discounts* (nval=p) along with the preposition (nvalperp=for)

The methodology as we have presented it relies on a pre-existing treebank. While developing the feature constraints requires an understanding of linguistic analyses and treebank conventions, we found that the environment was a comfortable one. The fact that constraints are solved in the treebank nearly eliminates the issue of ambiguity, allowing the computational linguist to concentrate on correct analyses while developing the constraint grammar. We envision this platform as a standard platform for easily augmenting existing treebanks with features of interest to the computational linguist.

PCFG Compilation and Parsing application

In treebank parsing applications, PCFGs are often created by incorporating features into context free grammar symbols Klein & Manning (2003). We implemented a method which compiles a frequency table for a PCFG from the annotated treebank database. For each symbol, a list of attributes to be incorporated is stipulated. For instance, it may be stipulated that VP incorporates the attributes Vform and Slash, and that verbs incorporate valence and Vform. A program reads the shared forest structures produced by constraint solving, and collects frequencies of occurrences of local tree configurations, including context free symbols and incorporated features. In cases where constraint solving introduced ambiguity, frequencies are split by a non-probabilistic version of inside-outside. The result is a rule frequency table and frequency lexicon which can be used by a probabilistic parser.

PCFGs derived in this way can be used by a parser to construct maximal probability (Viterbi) parses. We evaluate the quality of the transformed treebank and the utility of our feature annotation using standard PARSEVAL measures. Our grammar scores are comparable to state-of-the art unlexicalized grammars (the current best f-score for an unlexicalized treebank grammar to our knowledge is 86.6 in Schmid (2006)) . Our labelled bracketing score over section 23 of the Penn Treebank is 86.03 (recall), 86.21 (precision) and 86.12 (f-score).

Re-estimating lexical parameters

The PCFG trained over the transformed treebank has parameters related to lexical properties of words such as subcategorization features on verbs, attachment preference of adverbs (sentential, nominal, verbal adverbs), valence and prepositional preferences of nouns. However, since these parameters are tied to particular words, they are not well estimated in a treebank PCFG. In order to have a large-scale lexicon with accurate information, it is necessary to learn parameters from data of a much larger magnitude than available treebanks. We have experimented with learning these parameters over a large unannotated corpus using unsupervised training based on the inside-outside algorithm. The inside-outside algorithm iteratively re-estimates the parameters of a PCFG, given unannotated data. We used a modified version of the inside-outside algorithm in which we re-estimated lexical parameters

Iteration i	Standard Inside-Outside	Modified Inside-Outside
0 (smoothed treebank PCFG)	48.810	48.810
1	39.921	37.698
2	38.928	30.159
3	37.041	29.167
4	37.239	28.869
5	37.835	28.472

Figure 3: Valence error percentages for novel verbs

from unannotated data, but retain syntactic parameters originally learnt from the treebank (Deoskar & Rooth, 2006). Here we report results on learning the subcategorization frames for 100 test verbs. All tokens of these verbs were held out from the original treebank, so that in effect they are novel verbs which are not represented in the Treebank PCFG. The training data for the unsupervised algorithm was a set of 10000 unannotated sentences from the New York times containing occurrences of these verbs. We ran the modified inside-outside procedure using a smoothed version of a PCFG derived by the method described above. After each iteration, we obtained a PCFG model. We obtained the maximum probability parses for the test sentences using these PCFGs. The sub-cat frame of each verb token was compared to that on the transformed treebank. We found that the detection of the correct frame improved significantly after each iteration of our unsupervised procedure (Figure 3). The problem of subcategorization induction has been addressed in several approaches before. The most comprehensive evaluation of subcategorization frames acquired automatically for English is O'Donovan et al. (2005) – they have a large number of verb lemmas (4362), their frames are not pre-specified, and are fine-grained (they include specific preposition and particle use). Their approach is parallel to ours since they annotate the Penn treebank with LFG f-structure information. They also state that their system is a bootstrap to learn sub-cat information from larger corpora by parsing annotated data in their probabilistic LFG framework, but do not give results of induction of frames from unannotated data. Our results on verbal valence show that large-scale induction using a mathematically well-defined framework like inside-outside estimation of PCFGs is promising. Our evaluation is different from previous approaches in that we evaluate the subcategorization of tokens of verbs in maximum probability parses, and not over existing dictionaries. We believe that this evaluation over token occurrences is directly relevant to NLP tasks.

We have presented a framework that allows for augmentation of a treebank with linguistically motivated features which also allows the building of a PCFG that can be further used in applications for learning of lexical information. The framework can be applied to languages with existing treebanks in order to obtain treebank-aligned resources and to bootstrap induction of lexical information from unannotated data. We plan to use the framework to learn other lexically dependent parameters such as the prepositional attachment preference of verbs and nouns, attachment preference (sentential, nominal, verbal) of adverbs, valence of nouns, etc. in order to create probabilistic lexicons useful for parsing where this type of information about lexical items is represented.

Distribution

The resources and programs used to build the augmented treebank and the treebank-aligned PCFG (along with Make files to build them) are being released. The procedure for building the PCFG is parameterized by the features which are incorporated in the PCFG. The components are modular and can be used in ways other than the ones discussed here. For instance, feature constraints can be solved in a Viterbi tree resulting from parsing with the PCFG. The components used in the compilation and induction procedure are listed below. Bitpar and yap-compiler have been distributed by Helmut Schmid.

1. Regularize treebank
2. Map output of 1 to feature constraint grammar
3. Map each regularized treebank tree to trivial shared forests representing one tree
4. Solve feature constraints in the shared forest (yap-solver, (Schmid, 2000)
5. Map feature shared forest to PCFG rules and lexical entries with incorporated features (Privman, 2003)
6. Smoothing of PCFG lexicon
7. PCFG viterbi parsing and inside-outside re-estimation (bitpar, (Schmid, 2004)
8. Lexicon smoothing for modified inside-outside procedure

Acknowledgements

Thanks to Helmut Schmid for providing distributions and source code for his context free and feature constraint parsers, and for responding to numerous requests. The program for building the PCFG was written by Lior Privman.

References

Beil, F., Carroll, G., Prescher, D., Riezler, S., & Rooth, M. (1999). Inside-outside estimation of a lexicalized PCFG for German. In *Proceedings of ACL 1999*.

Briscoe, T. & Carroll, J. (1997). Automatic Extraction of Subcategorization from Corpora. In *Proceedings of the 5th ACL Conference on Applied NLP*.

Deoskar, T. & Rooth, M. (2006). Corpus Induction of Lexicons for Treebank PCFGs by Inside-Outside Estimation and Frequency Transformations. Ms.

im Walde, S. S. (2002). A Subcategorisation Lexicon for German Verbs induced from a Lexicalised PCFG. In *LREC 2002*.

Johnson, M. (1998). PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4).

Klein, D. & Manning, C. (2003). Accurate unlexicalized parsing. In *ACL-03*. Sapporo, Japan.

Lari, K. & Young, S. J. (1990). The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35–56.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*.

O'Donovan, R., Burke, M., Cahill, A., van Genabith, J., & Way, A. (2005). Large-Scale Induction and Evaluation of Lexical Resources from the Penn-II and Penn-III Treebanks. *Computational Linguistics*, 31:329–365.

Privman, L. (2003). Yappffun: Java implemention of shared forest algorithms. Computational Linguistics Lab, Cornell University.

Schmid, H. (2000). *YAP - Parsing and Disambiguation With Feature-Based Grammars*. Ph.D. thesis, University of Stuttgart.

Schmid, H. (2004). Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of COLING 2004*.

Schmid, H. (2006). Trace Prediction and Recovery with Unlexicalised PCFGs and Slash Features. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 177–184. Sydney, Australia.