Two Statistical Parsing Models Applied to the Chinese Treebank

Daniel M. Bikel **David Chiang**
Department of Computer & Information Science
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104-6389
{dbikel,dchiang}@cis.upenn.edu

Abstract

This paper presents the first-ever results of applying statistical parsing models to the newly-available Chinese Treebank. We have employed two models, one extracted and adapted from BBN's SIFT System (Miller et al., 1998) and a TAG-based parsing model, adapted from (Chiang, 2000). On sentences with $\leq$40 words, the former model performs at 69% precision, 75% recall, and the latter at 77% precision and 78% recall.

1 Introduction

Ever since the success of HMMs' application to part-of-speech tagging in (Church, 1988), machine learning approaches to natural language processing have steadily become more widespread. This increase has of course been due to their proven efficacy in many tasks, but also to their *engineering* efficacy. Many machine learning approaches let the data speak for itself (*data ipsa loquuntur*), as it were, allowing the modeler to focus on what features of the data are important, rather than on the complicated interaction of such features, as had often been the case with hand-crafted NLP systems. The success of statistical methods in particular has been quite evident in the area of syntactic parsing, most recently with the outstanding results of (Charniak, 2000) and (Collins, 2000) on the now-standard English test set of the Penn Treebank (Marcus et al., 1993). A significant trend in parsing models has been the incorporation of linguistically-motivated features; however, it is important to note that "linguistically-motivated" does not necessarily mean "language-dependent"—often, it means just the opposite. For example, almost all statistical parsers make use of lexicalized nonterminals in some way, which allows lexical items' indiosyncratic parsing preferences to be modeled, but the paring between head words and their parent nonterminals is determined almost entirely by the training data, thereby making this *feature*—which models preferences of particular words of a particular language—almost entirely language-independent. In this paper, we will explore the use of two parsing models, which were originally designed for English parsing, on parsing Chinese, using the newly-available Chinese Treebank. We will show that the language-dependent components of these parsers are quite compact, and that with little effort they can be adapted to produce promising results for Chinese parsing. We also discuss directions for future work.

2 Models and Modifications

We will briefly describe the two parsing models employed (for a full description of the BBN model, see (Miller et al., 1998) and also (Bikel, 2000); for a full description of the TAG model, see (Chiang, 2000)).

2.1 Model 2 of (Collins, 1997)

Both parsing models discussed in this paper inherit a great deal from this model, so we briefly describe its "progenitive" features here, describing only how each of the two models of this paper differ in the subsequent two sections.

The lexicalized PCFG that sits behind Model 2 of (Collins, 1997) has rules of the form

$$P \rightarrow L_n L_{n-1} \cdots L_1 H R_1 \cdots R_{n-1} R_n \quad (1)$$

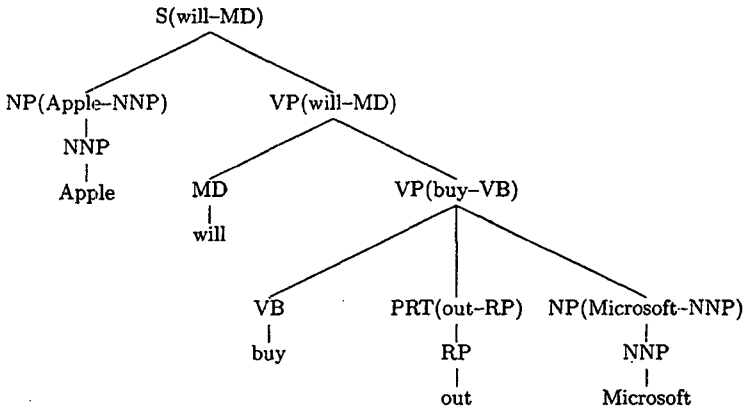

Figure 1: A sample sentence with parse tree.

where P, L_i, R_i and H are all lexicalized nonterminals, and P inherits its lexical head from its distinguished head child, H. In this generative model, first P is generated, then its head-child H, then each of the left- and right-modifying nonterminals are generated from the head outward. The modifying nonterminals L_i and R_i are generated conditioning on P and H, as well as a distance metric (based on what material intervenes between the currently-generated modifying nonterminal and H) and an incremental subcat frame feature (a multiset containing the complements of H that have yet to be generated on the side of H in which the currently-generated nonterminal falls). Note that if the modifying nonterminals were generated completely independently, the model would be very impoverished, but in actuality, by including the distance and subcat frame features, the model captures a crucial bit of linguistic reality, *viz.*, that words often have well-defined sets of complements and adjuncts, dispersed with some well-defined distribution in the right hand sides of a (context-free) rewriting system.

2.2 BBN Model

2.2.1 Overview

The BBN model is also of the lexicalized PCFG variety. In the BBN model, as with Model 2 of (Collins, 1997), modifying nonterminals are generated conditioning both on the parent P and its head child H. Unlike Model 2 of (Collins, 1997), they are also generated conditioning on the previously generated modifying nonterminal, L_{i-1} or R_{i-1}, and there is no subcat frame or distance feature. While the BBN model does not perform at the level of Model 2 of (Collins, 1997) on Wall Street Journal text, it is also less language-dependent, eschewing the distance metric (which relied on specific features of the English Treebank) in favor of the "bigrams on nonterminals" model.

2.2.2 Model Parameters

This section briefly describes the top-level parameters used in the BBN parsing model. We use p to denote the unlexicalized nonterminal corresponding to P in (1), and similarly for l_i, r_i and h. We now present the top-level generation probabilities, along with examples from Figure 1. For brevity, we omit the smoothing details of BBN's model (see (Miller et al., 1998) for a complete description); we note that all smoothing weights are computed via the technique described in (Bikel et al., 1997).

The probability of generating p as the root label is predicted conditioning on only +TOP+, which is the hidden root of all parse trees:

$$P(p \mid +\text{TOP}+), \textit{ e.g., } P(\text{S} \mid +\text{TOP}+). \quad (2)$$

The probability of generating a head node h with a parent p is

$$P(h \mid p), \textit{ e.g., } P(\text{VP} \mid \text{S}). \quad (3)$$

The probability of generating a left-modifier l_i is

$$P_L(l_i \mid l_{i-1}, p, h, w_h),, \textit{ e.g., } \quad (4)$$
$$P_L(\text{NP} \mid +\text{BEGIN}+, \text{S}, \text{VP}, \text{will})$$

when generating the NP for NP(Apple–NNP), and the probability of generating a right modifier r_i is

$$P_R(r_i \mid r_{i-1}, p, h, w_h), \text{ e.g.,} \quad P_R(\text{NP} \mid \text{PRT}, \text{VP}, \text{VB}, \text{buy}) \tag{5}$$

when generating the NP for NP(Microsoft–NNP).[1]

The probabilities for generating lexical elements (part-of-speech tags and words) are as follows. The part of speech tag of the head of the entire sentence, t_h, is computed conditioning only on the top-most symbol p:[2]

$$P(t_h \mid p). \tag{6}$$

Part of speech tags of modifier constituents, t_{l_i} and t_{r_i}, are predicted conditioning on the modifier constituent l_i or r_i, the tag of the head constituent, t_h, and the word of the head constituent, w_h

$$P(t_{l_i} \mid l_i, t_h, w_h) \text{ and } P(t_{r_i} \mid r_i, t_h, w_h). \tag{7}$$

The head word of the entire sentence, w_h, is predicted conditioning only on the top-most symbol p and t_h:

$$P(w_h \mid t_h, p). \tag{8}$$

Head words of modifier constituents, w_{l_i} and w_{r_i}, are predicted conditioning on all the context used for predicting parts of speech in (7), as well as the parts of speech themsleves

$$P(w_{l_i} \mid t_{l_i}, l_i, t_h, w_h) \quad \text{and} \quad P(w_{r_i} \mid t_{r_i}, r_i, t_h, w_h). \tag{9}$$

The original English model also included a word feature to help reduce part-of-speech ambiguity for unknown words, but this component of the model was removed for Chinese, as it was language-dependent.

The probability of an entire parse tree is the product of the probabilities of generating all of the elements of that parse tree, where an element is either a constituent label, a part of speech tag or a word. We obtain maximum-likelihood estimates of the parameters of this model using frequencies gathered from the training data.

[1]The hidden nonterminal +BEGIN+ is used to provide a convenient mechanism for determining the initial probability of the underlying Markov process generating the modifying nonterminals; the hidden nonterminal +END+ is used to provide consistency to the underlying Markov process, *i.e.*, so that the probabilities of all possible nonterminal sequences sum to 1.

[2]This is the one place where we altered the original model, as the lexical components of the head of the entire sentence were all being estimated incorrectly, causing an inconsistency in the model. We corrected the estimation of t_h and w_h in our implementation.

2.3 TAG Model

The model of (Chiang, 2000) is based on stochastic TAG (Resnik, 1992; Schabes, 1992). In this model a parse tree is built up not out of lexicalized phrase-structure rules but by tree fragments (called *elementary trees*) which are lexicalized in the sense that each fragment contains exactly one lexical item (its *anchor*).

In the variant of TAG we use, there are three kinds of elementary tree: initial, (predicative) auxiliary, and modifier, and three composition operations: substitution, adjunction, and sister-adjunction. Figure 2 illustrates all three of these operations. α_1 is an initial tree which substitutes at the leftmost node labeled NP↓; β is an auxiliary tree which adjoins at the node labeled VP. See (Joshi and Schabes, 1997) for a more detailed explanation.

Sister-adjunction is not a standard TAG operation, but borrowed from D-Tree Grammar (Rambow et al., 1995). In Figure 2 the modifier tree γ is sister adjoined between the nodes labeled VB and NP↓. Multiple modifier trees can adjoin at the same place, in the spirit of (Schabes and Shieber, 1994).

In stochastic TAG, the probability of generating an elementary tree depends on the elementary tree itself and the elementary tree it attaches to. The parameters are as follows:

$$\sum_{\alpha} P_i(\alpha) = 1$$

$$\sum_{\alpha} P_s(\alpha \mid \eta) = 1$$

$$\sum_{\beta} P_a(\beta \mid \eta) + P_a(NONE \mid \eta) = 1$$

where α ranges over initial trees, β over auxiliary trees, γ over modifier trees, and η over nodes. $P_i(\alpha)$ is the probability of beginning a derivation with α; $P_s(\alpha \mid \eta)$ is the probability of substituting α at η; $P_a(\beta \mid \eta)$ is the probability of adjoining β at η; finally, $P_a(NONE \mid \eta)$ is the probability of nothing adjoining at η.

Our variant adds another set of parameters:

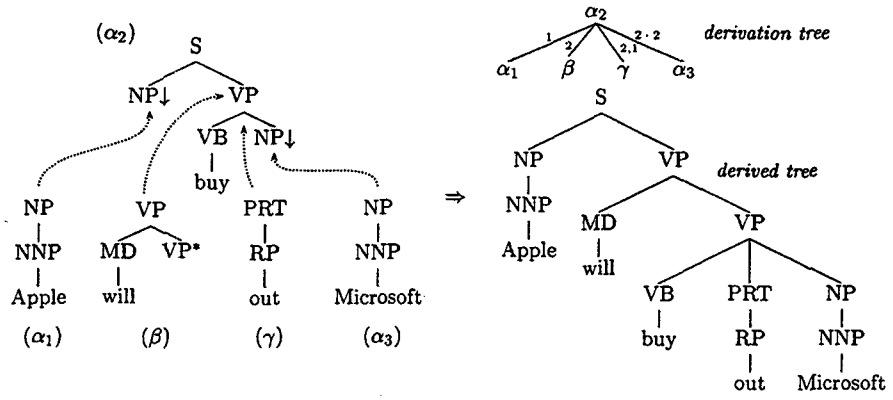

Figure 2: Grammar and derivation for "Apple will buy out Microsoft."

$$\sum_{\gamma} P_{sa}(\gamma \mid \eta, i, f) + P_{sa}(STOP \mid \eta, i, f) = 1$$

This is the probability of sister-adjoining γ between the ith and $i+1$th children of η (allowing for two imaginary children beyond the leftmost and rightmost children). Since multiple modifier trees can adjoin at the same location, $P_{sa}(\gamma)$ is also conditioned on a flag f which indicates whether γ is the first modifier tree (i.e., the one closest to the head) to adjoin at that location.

For our model we break down these probabilities further: first the elementary tree is generated without its anchor, and then its anchor is generated. See (Chiang, 2000) for more details.

During training each example is broken into elementary trees using head rules and argument/adjunct rules similar to those of (Collins, 1997). The rules are interpreted as follows: a head is kept in the same elementary tree in its parent, an argument is broken off into a separate initial tree, leaving a substitution node, and an adjunct is broken off into a separate modifier tree. A different rule is used for extracting auxiliary trees; see (Chiang, 2000) for details. Xia (1999) describes a similar process, and in fact our rules for the Xinhua corpus are based on hers.

2.4 Modifications

The primary language-dependent component that had to be changed in both models was the head table, used to determine heads when training. We modified the head rules described in (Xia, 1999) for the Xinhua corpus and substituted these new rules into both models.

The (Chiang, 2000) model had the following additional modifications.

- The new corpus had to be prepared for use with the trainer and parser. Aside from technicalities, this involved retraining the part-of-speech tagger described in (Ratnaparkhi, 1997), which was used for tagging unknown words. We also lowered the unknown word threshold from 4 to 2 because the Xinhua corpus was smaller than the WSJ corpus.
- In addition to the change to the head-finding rules, we also changed the rules for classifying modifiers as arguments or adjuncts. In both cases the new rules were adapted from (Xia, 1999).
- For the tests done in this paper, a beam width of 10^{-4} was used.

The BBN model had the following additional modifications:

- As with the (Chiang, 2000) model, we similarly lowered the unknown word threshold of the BBN model from its default 5 to 2.
- The language-dependent word-feature was eliminated, causing parts of speech for unknown words to be predicted solely on the head relations in the model.
- The default beam size in the probabilistic CKY parsing algorithm was widened. The default beam pruned away chart entries whose scores were not within a factor of e^{-5} of the top-ranked subtree; this

Model, test set	≤40 words				
	LR	LP	$\overline{\text{CB}}$	0CB	≤2CB
BBN-all†, WSJ-all	84.7	86.5	1.12	60.6	83.2
BBN-small†, WSJ-small*	79.0	80.7	1.66	47.0	74.6
BBN, Xinhua‡	**69.0**	**74.8**	**2.05**	**45.0**	**68.5**
Chiang-all, WSJ-all	86.9	86.6	1.09	63.2	84.3
Chiang-small, WSJ-small	78.9	79.6	1.75	44.8	72.4
Chiang, Xinhua	**76.8**	**77.8**	**1.99**	**50.8**	**74.1**
	≤100 words				
	LR	LP	$\overline{\text{CB}}$	0CB	≤2CB
BBN-all†, WSJ-all	83.9	85.7	1.31	57.8	80.8
BBN-small†, WSJ-small*	78.4	80.0	1.92	44.3	71.3
BBN, Xinhua‡	**67.5**	**73.5**	**2.87**	**39.9**	**61.8**
Chiang-all, WSJ-all	86.2	85.8	1.29	60.4	81.8
Chiang-small, WSJ-small	77.1	78.8	2.00	43.25	70.5
Chiang, Xinhua	**73.3**	**74.6**	**3.03**	**44.8**	**66.8**

Table 1: Results for both parsing models on all test sets. Key: LR = labeled recall, LP = labeled precision, $\overline{\text{CB}}$ = avg. crossing brackets, 0CB = zero crossing brackets, ≤2CB = ≤2 crossing brackets. All results are percentages, except for those in the $\overline{\text{CB}}$ column. †Used larger beam settings and lower unknown word threshold than the defaults. *3 of the 400 sentences were not parsed due to timeouts and/or pruning problems. ‡3 of the 348 sentences did not get parsed due to pruning problems, and 2 other sentences had length mismatches (scoring program errors).

tight limit was changed to e^{-9}. Also, the default decoder pruned away all but the top 25-ranked chart entries in each cell; this limit was expanded to 50.

3 Experiments and Results

The Chinese Treebank consists of 4185 sentences of Xinhua newswire text. We blindly separated this into training, devtest and test sets, with a roughly 80/10/10 split, putting files 001–270 (3484 sentences, 84,873 words) into the training set, 301–325 (353 sentences, 6776 words) into the development test set and reserving 271–300 (348 sentences, 7980 words) for testing. See Table 1 for results.

In order to put the new Chinese Treebank results into context with the unmodified (English) parsing models, we present results on two test sets from the Wall Street Journal: WSJ-all, which is the complete Section 23 (the *de facto* standard test set for English parsing), and WSJ-small, which is the first 400 sentences of Section 23 and which is roughly comparable in size to the Chinese test set. Furthermore, when testing on WSJ-small, we trained on a subset of our English training data roughly equivalent in size to our Chinese training set (Sections 02 and 03 of the Penn Treebank); we have indicated models trained on all English training with "-all", and models trained with the reduced English training set with "-small". Therefore, by comparing the WSJ-small results with the Chinese results, one can reasonably gauge the performance gap between English parsing on the Penn Treebank and Chinese parsing on the Chinese Treebank.

The reader will note that the modified BBN model does significantly poorer than (Chiang, 2000) on Chinese. While more investigation is required, we suspect part of the difference may be due to the fact that currently, the BBN model uses language-specific rules to guess part of speech tags for unknown words.

4 Conclusions and Future Work

There is no question that a great deal of care and expertise went into creating the Chinese Treebank, and that it is a source of important grammatical information that is unique to the Chinese language. However, there are definite similarities between the grammars of English and Chinese, especially when viewed through the lens of the statistical models we employed

here. In both languages, the nouns, adjectives, adverbs, and verbs have preferences for certain arguments and adjuncts, and these preferences—in spite of the potentially vastly-different configurations of these items—are effectively modeled. As discussed in the introduction, lexical items' idiosyncratic parsing preferences are modeled by lexicalizing the grammar formalism, using a lexicalized PCFG in one case and a lexicalized stochastic TAG in the other. Linguistically-reasonable independence assumptions are made, such as the independence of grammar productions in the case of the PCFG model, or the independence of the composition operations in the case of the LTAG model, and we would argue that these assumptions are no less reasonable for the Chinese grammar than they are for that of English. While results for the two languages are far from equal, we believe that further tuning of the head rules, and analysis of development test set errors will yield significant performance gains on Chinese to close the gap. Finally, we fully expect that absolute performance will increase greatly as additional high-quality Chinese parse data becomes available.

5 Acknowledgements

This research was funded in part by NSF grant SBR-89-20230-15. We would greatly like to acknowledge the researchers at BBN who allowed us to use their model: Ralph Weischedel, Scott Miller, Lance Ramshaw, Heidi Fox and Sean Boisen. We would also like to thank Mike Collins and our advisors Aravind Joshi and Mitch Marcus.

References

Daniel M. Bikel, Richard Schwartz, Ralph Weischedel, and Scott Miller. 1997. Nymble: A high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201, , Washington, D.C.

Daniel M. Bikel. 2000. A statistical model for parsing and word-sense disambiguation. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong, October.

Eugene Charniak. 2000. A maximum entropy–inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, Seattle, Washington, April 29 to May 4.

David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics.*

Kenneth Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL-EACL '97*, pages 16–23.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *International Conference on Machine Learning.* (to appear).

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In A. Salomma and G. Rosenberg, editors, *Handbook of Formal Languages and Automata*, volume 3, pages 69–124. Springer-Verlag, Heidelberg.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 1998. SIFT – Statistically-derived Information From Text. In *Seventh Message Understanding Conference (MUC-7)*, Washington, D.C.

Owen Rambow, K. Vijay-Shanker, and David Weir. 1995. D-tree grammars. In *Proceedings of the 33rd Annual Meeting of the Assocation for Computational Linguistics*, pages 151–158.

Adwait Ratnaparkhi. 1997. A simple introduction to maximum entropy models for natural language processing. Technical Report IRCS Report 97–08, Institute for Research in Cognitive Science, May.

Philip Resnik. 1992. Probabilistic tree-adjoining grammar as a framework for statistical natural language processing. In *Proceedings of COLING-92*, pages 418–424.

Yves Schabes and Stuart M. Shieber. 1994. An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20(1):91–124.

Yves Schabes. 1992. Stochastic lexicalized tree-adjoining grammars. In *Proceedings of COLING-92*, pages 426–432.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99).*