Log-linear Models of Non-projective Trees, k-best MST Parsing and Tree-ranking

Keith Hall[1] and **Jiří Havelka**[2] and **David A. Smith**[1]

[1]Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD USA
keith_hall@jhu.edu
dasmith@cs.jhu.edu

[2]Institute of Formal and Applied Linguistics
Charles University
Prague, Czech Republic
havelka@ufal.mff.cuni.cz

Abstract

We present our system used in the CoNLL 2007 shared task on multilingual parsing. The system is composed of three components: a k-best maximum spanning tree (MST) parser, a tree labeler, and a reranker that orders the k-best labeled trees. We present two techniques for training the MST parser: tree-normalized and graph-normalized conditional training. The tree-based reranking model allows us to explicitly model *global* syntactic phenomena. We describe the reranker features which include non-projective edge attributes. We provide an analysis of the errors made by our system and suggest changes to the models and features that might rectify the current system.

1 Introduction

Reranking the output of a k-best parser has been shown to improve upon the best results of a state-of-the-art constituency parser (Charniak and Johnson, 2005). This is primarily due to the ability to incorporate complex structural features that cannot be modeled under a CFG. Recent work shows that k-best maximum spanning tree (MST) parsing and reranking is also viable (Hall, 2007). In the current work, we explore the k-best MST parsing paradigm along with a tree-based reranker. A system using the parsing techniques presented in this paper was entered in the CoNLL 2007 shared task competition (Nivre et al., 2007). This task evaluated parsing performance on 10 languages: Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian, and Turkish using data originating from a wide variety of dependency treebanks, and transformations of constituency-based treebanks (Hajič et al., 2004; Aduriz et al., 2003; Martí et al., 2007; Chen et al., 2003; Böhmová et al., 2003; Marcus et al., 1993; Johansson and Nugues, 2007; Prokopidis et al., 2005; Csendes et al., 2005; Montemagni et al., 2003; Oflazer et al., 2003).

We show that oracle parse accuracy[1] of the output of our k-best parser is generally higher than the best reported results. We also present the results of a reranker based on a rich set of structural features, including features explicitly targeted at modeling non-projective configurations. Labeling of the dependency edges is accomplished by an edge labeler based on the same feature set as used in training the k-best MST parser.

2 Parser Description

Our parser is composed of three components: a k-best MST parser, a tree-labeler, and a tree-reranker. Log-linear models are used for each of the components independently. In this section we give an overview of the models, the training techniques, and the decoders.

2.1 MST Parsing, Reranking, and Labeling

The connection between the maximum spanning tree problem and dependency parsing stems from the observation that a dependency parse is simply an oriented spanning tree on the graph of all possible

[1]The oracle accuracy for a set of hypotheses is the maximal accuracy for any of the hypotheses.

Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 962–966,
Prague, June 2007.

dependency links (the fully connected dependency graph). Unfortunately, by mapping the problem to a graph, we assume that the scores associated with edges are independent, and thus, are limited to *edge-factored* models.

Edge-factored models are severely limited in their capacity to predict structure. In fact, they can only directly model parent-child links. In order to alleviate this, we use a k-best MST parser to generate a set of candidate hypotheses. Then, we rerank these trees using a model based on rich structural features that model features such as valency, subcategorization, ancestry relationships, and sibling interactions, as well as features capturing the global structure of dependency trees, aimed primarily at modeling language specific non-projective configurations.

We assign dependency labels to entire trees, rather than predicting the labels during tree construction. Given that we have a reranking process, we can label the k-best tree hypotheses output from our MST parser, and rerank the labeled trees. We have explored both labeled and unlabeled reranking. In the latter case, we simply label the maximal unlabeled tree.

2.1.1 MST Training

McDonald et al. (2005) present a technique for training discriminative models for dependency parsing. The edge-factored models we use for MST parsing are closely related to those described in the previous work, but allow for the efficient computation of normalization factors which are required for first and second-order (gradient-based) training techniques.

We consider two estimation procedures for *parent-prediction* models. A parent-prediction model assigns a conditional score $s(g|d)$ for every parent-child pair (we denote the parent/governor g, and the child/dependent d), where $s(g|d) = s(g,d)/\sum_{g'} s(g',d)$. In our work, we compute probabilities $p(g|d)$ based on conditional log-linear models. This is an approximation to a generative model that predicts each node once (i.e., $\prod_d p(d|g)$).

In the graph-normalized model, we assume that the conditional distributions are independent of one another. In particular, we find the model parameters that maximize the likelihood of $p(g^*|d)$, where g^* is the correct parent in the training data. We perform the optimization over the entire training set, tying the feature parameters. In particular, we perform maximum entropy (MaxEnt) estimation over the conditional distribution using second-order gradient descent optimization techniques.[2] An advantage of the parent-prediction model is that we can frame the estimation problem as that of minimum-error training with a zero-one loss term:

$$p(e,g|d) = \frac{\exp(\sum_i \lambda_i f_i(e,g,d))}{Z_d} \qquad (1)$$

where $e \in \{0,1\}$ is the error term (e is 1 for the correct parent and 0 for all other nodes) and $Z_d = \sum_j \exp(\sum_i \lambda_i f_i(e_j, g_j, d))$ is the normalization constant for node d. Note that the normalization factor considers all graphs with in-degree zero for the root node and in-degree one for other nodes.

At parsing time, of course, our parent predictions are constrained to produce a (non-projective) tree structure. We can sum over all non-projective spanning trees by taking the determinant of the Kirchhoff matrix of the graph defined above, minus the row and column corresponding to the root node (Smith and Smith, 2007). Training graph-normalized and tree-normalized models under identical conditions, we find tree normalization wins by 0.5% to 1% absolute dependency accuracy. Although tree normalization also shows a (smaller) advantage in k-best oracle accuracy, we do not believe it would have a large effect on our reranking results.

2.1.2 Reranker Training

The reranker is based on a conditional log-linear model subject to the MaxEnt constraints using the same second-order optimization procedures as the graph-normalized MST models. The primary difference here is that there is no single correct tree in the set of k candidate parse trees. Instead, we have k trees that are generated by our k-best parser, each with a score assigned by the parser. If we are performing labeled reranking, we label each of these hypotheses with l possible labelings, each with a score assigned by the labeler.

As with the parent-prediction, graph-normalized model, we perform minimum-error training. The

[2]For the graph-normalized models, we use L-BFGS optimization provided through the TAO/PETSC optimization library (Benson et al., 2005; Balay et al., 2004).

optimization is achieved by assuming the oracle-best parse(s) are correct and the remaining hypotheses are incorrect. Furthermore, the feature values are scaled according to the relative difference between the oracle-best score and the score assigned to the non-oracle-best hypothesis.

Note that any reranker could be used in place of our current model. We have chosen to keep the reranker model closely related to the MST parsing model so that we can share feature representations and training procedures.

2.1.3 Labeler Training

We used the same edge features to train a separate log-linear labeling model. Each edge feature was conjoined with a potential label, and we then maximized the likelihood of the labeling in the training data. Since this model is also edge-factored, we can store the labeler scores for each of the n^2 potential edges in the dependency tree. In the submitted system, we simply extracted the Viterbi predictions of the labeler for the unlabeled trees selected by the reranker. We also (see below) ran experiments where each entry in the k-best lists input as training data to the reranker was augmented by its l-best labelings. We hoped thereby to inject more diversity into the resulting structures.

2.1.4 Model Features

Our MST models are based on the features described in (Hall, 2007); specifically, we use features based on a dependency nodes' form, lemma, coarse and fine part-of-speech tag, and morphological-string attributes. Additionally, we use surface-string distance between the parent and child, buckets of features indicating if a particular form/lemma/tag occurred between or next to the parent and child, and a branching feature indicating whether the child is to the left or right of the parent. Composite features, combining the above features are also included (e.g., a single feature combining branching, parent & child form, parent & child tag).

The tree-based reranker includes the features described in (Hall, 2007) as well as features based on non-projective edge attributes explored in (Havelka, 2007a; Havelka, 2007b). One set of features models relationships of nodes with their siblings, including valency and subcategorization. A second set of features models global tree structure and includes features based on a node's ancestors and the depth and size of its subtree. A third set of features models the interaction of word order and tree structure as manifested on individual edges, i.e., the features model language specific projective and non-projective configurations. They include edge-based features corresponding to the global constraints of projectivity, planarity and well-nestedness, and for non-projective edges, they furthermore include level type, level signature and ancestor-in-gap features. All features allow for an arbitrary degree of lexicalization; in the reported results, the first two sets of features use coarse and fine part-of-speech lexicalizations, while the features in the third set are used in their unlexicalized form due to time limitations.

3 Results and Analysis

Hall (2007) shows that the oracle parsing accuracy of a k-best edge-factored MST parser is considerably higher than the one-best score of the same parser, even when k is small. We have verified that this is true for the CoNLL shared-task data by evaluating the oracle rates on a randomly sampled development set for each language.

In order to select optimal model parameters for the MST parser, the labeler, and reranker, we sampled approximately 200 sentences from each training set to use as a development test set. Training the reranker requires a jackknife n-fold training procedure where $n-1$ partitions are used to train a model that parses the remaining partition. This is done n times to generate k-best parses for the entire training set without using models trained on the data they are run on.

For lack of space, we report only results on the CoNLL evaluation data set here, but note that the trends observed on the evaluation data are identical to those observed on our development sets.

In Table 1 we present results for labeled (and unlabeled) dependency accuracy on the CoNLL 2007 evaluation data set. We report the oracle accuracy for different sized k-best hypothesis sets. The columns are labeled by the number of trees output from the MST parser, k;[3] and by the number of al-

[3]All results are reported for the graph-normalized training technique.

Language	Oracle Accuracy				New	CoNLL07	CoNLL07
	$k = 1, l = 1$	$k = 10, l = 5$	$k = 50, l = 1$	$k = 50, l = 2$	Reranked	Reported	Best
Arabic	(83.10)	(85.56)	(86.96)		(83.67)	73.40 (83.45)	76.52 (86.09)
Basque	67.92 (76.88)	76.25 (82.19)	69.93 (84.99)	76.81	(77.76)	69.80 (78.52)	76.92 (82.80)
Catalan	82.28 (87.82)	85.11 (90.87)	86.82 (92.68)	86.82	(89.43)	82.38 (87.80)	88.70 (93.40)
Chinese	73.86 (85.58)	91.32 (93.39)	82.39 (95.80)	92.21	(87.87)	82.77 (87.91)	84.69 (88.94)
Czech	74.05 (80.21)	78.58 (85.08)	80.97 (87.60)	80.97	(82.20)	72.27 (78.47)	80.19 (86.28)
English	82.21 (83.63)	85.95 (87.59)	87.99 (89.75)	87.99	(85.31)	81.93 (83.21)	89.61 (90.63)
Greek	72.21 (81.16)	78.58 (84.89)	74.13 (86.95)	79.48	(81.81)	74.21 (82.04)	76.31 (84.08)
Hungarian	71.68 (78.57)	79.70 (83.03)	74.32 (85.12)	80.75	(80.05)	74.20 (79.34)	80.27 (83.55)
Italian	77.92 (83.16)	85.05 (87.54)	80.30 (89.66)	86.42	(84.71)	80.69 (84.81)	84.40 (87.91)
Turkish	75.34 (83.63)	83.96 (89.65)	77.78 (92.40)	84.98	(84.13)	77.42 (85.18)	79.81 (86.22)

Table 1: Labeled (unlabeled) attachment accuracy for k-best MST oracle results and reranked data on the evaluation set. The 1-best results ($k = 1, l = 1$) represent the performance of the MST parser without reranking. The *New Reranked* field shows recent unlabeled reranking results of 50-best trees using a modified feature set. For arabic, we only report unlabeled accuracy for different k and l.

ternative labelings for each tree, l. When $k = 1$, the score is the best achievable by the edge-factored MST parser using our models. As k increases, the oracle parsing accuracy increases. The most extreme difference between the one-best accuracy and the 50-best oracle accuracy can be seen for Turkish where there is a difference of 9.64 points of accuracy (8.77 for the unlabeled trees). This means that the reranker need only select the correct tree from a set of 50 to increase the score by 9.64%. As our reranking results show, this is not as simple as it may appear.

We report the results for our CoNLL submission as well as recent results based on alternative parameters optimization on the development set. We report the latest results only for unlabeled accuracy of reranking 50-best MST output.

4 Conclusion

Our submission to the CoNLL 2007 shared task on multilingual parsing supports the hypothesis that edge-factored MST parsing is viable given an effective reranker. The reranker used in our submission was unable to achieve the oracle rates. We believe this is primarily related to a relatively impoverished feature set. Due to time constraints, we have not been able to train lexicalized reranking models. The introduction of lexicalized features in the reranker should influence the selection of better trees, which we know exist in the k-best hypothesis sets.

References

A. Abeillé, editor. 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer.

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Diaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank. In *Proc. of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 201–204.

Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. 2004. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory.

Steven J. Benson, Lois Curfman McInnes, Jorge Moré, and Jason Sarich. 2005. TAO user manual (revision 1.8). Technical Report ANL/MCS-TM-242, Mathematics and Computer Science Division, Argonne National Laboratory. http://www.mcs.anl.gov/tao.

A. Böhmová, J. Hajič, E. Hajičová, and B. Hladká. 2003. The PDT: a 3-level annotation scenario. In Abeillé (Abeillé, 2003), chapter 7, pages 103–127.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

K. Chen, C. Luo, M. Chang, F. Chen, C. Chen, C. Huang, and Z. Gao. 2003. Sinica treebank: Design criteria, representational issues and implementation. In Abeillé (Abeillé, 2003), chapter 13, pages 231–248.

D. Csendes, J. Csirik, T. Gyimóthy, and A. Kocsor. 2005. *The Szeged Treebank*. Springer.

J. Hajič, O. Smrž, P. Zemánek, J. Šnaidauf, and E. Beška. 2004. Prague Arabic dependency treebank: Development in data and tools. In *Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*, pages 110–117.

Keith Hall. 2007. k-best spanning tree parsing. In *(To Appear) Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Jiří Havelka. 2007a. Beyond projectivity: Multilingual evaluation of constraints and measures on non-projective structures. In *(To Appear) Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Jiří Havelka. 2007b. Relationship between non-projective edges, their level types, and well-nestedness. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 61–64.

R. Johansson and P. Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA)*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

M. A. Martí, M. Taulé, L. Màrquez, and M. Bertran. 2007. CESS-ECE: A multilingual and multilevel annotated corpus. Available for download from: http://www.lsi.upc.edu/∼mbertran/cess-ece/.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*.

S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Lenci, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Nana, F. Pianesi, and R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Abeillé (Abeillé, 2003), chapter 11, pages 189–210.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

K. Oflazer, B. Say, D. Zeynep Hakkani-Tür, and G. Tür. 2003. Building a Turkish treebank. In Abeillé (Abeillé, 2003), chapter 15, pages 261–277.

P. Prokopidis, E. Desypri, M. Koutsombogera, H. Papageorgiou, and S. Piperidis. 2005. Theoretical and practical issues in the construction of a Greek dependency treebank. In *Proc. of the 4th Workshop on Treebanks and Linguistic Theories (TLT)*, pages 149–160.

David A. Smith and Noah A. Smith. 2007. Probabilistic models of nonprojective dependency trees. In *(To Appear) Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing*.