Multilingual Dependency Analysis with a Two-Stage Discriminative Parser

Ryan McDonald **Kevin Lerman** **Fernando Pereira**
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA
{ryantm,klerman,pereira}@cis.upenn.edu

Abstract

We present a two-stage multilingual dependency parser and evaluate it on 13 diverse languages. The first stage is based on the unlabeled dependency parsing models described by McDonald and Pereira (2006) augmented with morphological features for a subset of the languages. The second stage takes the output from the first and labels all the edges in the dependency graph with appropriate syntactic categories using a globally trained sequence classifier over components of the graph. We report results on the CoNLL-X shared task (Buchholz et al., 2006) data sets and present an error analysis.

1 Introduction

Often in language processing we require a deep syntactic representation of a sentence in order to assist further processing. With the availability of resources such as the Penn WSJ Treebank, much of the focus in the parsing community had been on producing syntactic representations based on phrase-structure. However, recently their has been a revived interest in parsing models that produce dependency graph representations of sentences, which model words and their arguments through directed edges (Hudson, 1984; Mel′čuk, 1988). This interest has generally come about due to the computationally efficient and flexible nature of dependency graphs and their ability to easily model non-projectivity in freer-word order languages. Nivre (2005) gives an introduction to dependency representations of sentences and recent developments in dependency parsing strategies.

Dependency graphs also encode much of the deep syntactic information needed for further processing. This has been shown through their successful use in many standard natural language processing tasks, including machine translation (Ding and Palmer, 2005), sentence compression (McDonald, 2006), and textual inference (Haghighi et al., 2005).

In this paper we describe a two-stage discriminative parsing approach consisting of an unlabeled parser and a subsequent edge labeler. We evaluate this parser on a diverse set of 13 languages using data provided by the CoNLL-X shared-task organizers (Buchholz et al., 2006; Hajič et al., 2004; Simov et al., 2005; Simov and Osenova, 2003; Chen et al., 2003; Böhmová et al., 2003; Kromann, 2003; van der Beek et al., 2002; Brants et al., 2002; Kawata and Bartels, 2000; Afonso et al., 2002; Džeroski et al., 2006; Civit Torruella and Martí Antonín, 2002; Nilsson et al., 2005; Oflazer et al., 2003; Atalay et al., 2003). The results are promising and show the language independence of our system under the assumption of a labeled dependency corpus in the target language.

For the remainder of this paper, we denote by $\boldsymbol{x} = x_1, \ldots x_n$ a sentence with n words and by $\boldsymbol{y}$ a corresponding dependency graph. A dependency graph is represented by a set of ordered pairs $(i, j) \in \boldsymbol{y}$ in which x_j is a dependent and x_i is the corresponding head. Each edge can be assigned a label $l_{(i,j)}$ from a finite set L of predefined labels. We assume that all dependency graphs are trees but may be non-projective, both of which are true in the data sets we use.

2 Stage 1: Unlabeled Parsing

The first stage of our system creates an unlabeled parse $\boldsymbol{y}$ for an input sentence $\boldsymbol{x}$. This system is primarily based on the parsing models described by McDonald and Pereira (2006). That work extends the maximum spanning tree dependency parsing framework (McDonald et al., 2005a; McDonald et al., 2005b) to incorporate features over multiple edges in the dependency graph. An exact projective and an approximate non-projective parsing algorithm are presented, since it is shown that non-projective dependency parsing becomes NP-hard when features are extended beyond a single edge.

That system uses MIRA, an online large-margin learning algorithm, to compute model parameters. Its power lies in the ability to define a rich set of features over parsing decisions, as well as surface level features relative to these decisions. For instance, the system of McDonald et al. (2005a) incorporates features over the part of speech of words occurring between and around a possible head-dependent relation. These features are highly important to overall accuracy since they eliminate unlikely scenarios such as a preposition modifying a noun not directly to its left, or a noun modifying a verb with another verb occurring between them.

We augmented this model to incorporate morphological features derived from each token. Consider a proposed dependency of a dependent x_j on the head x_i, each with morphological features M_j and M_i respectively. We then add to the representation of the edge: M_i as head features, M_j as dependent features, and also each conjunction of a feature from both sets. These features play the obvious role of explicitly modeling consistencies and commonalities between a head and its dependents in terms of attributes like gender, case, or number. Not all data sets in our experiments include morphological features, so we use them only when available.

3 Stage 2: Label Classification

The second stage takes the output parse $\boldsymbol{y}$ for sentence $\boldsymbol{x}$ and classifies each edge $(i, j) \in \boldsymbol{y}$ with a particular label $l_{(i,j)}$. Ideally one would like to make all parsing and labeling decisions jointly so that the shared knowledge of both decisions will help resolve any ambiguities. However, the parser is fundamentally limited by the scope of local factorizations that make inference tractable. In our case this means we are forced only to consider features over single edges or pairs of edges. However, in a two stage system we can incorporate features over the entire output of the unlabeled parser since that structure is fixed as input. The simplest labeler would be to take as input an edge $(i, j) \in \boldsymbol{y}$ for sentence $\boldsymbol{x}$ and find the label with highest score,

$$l_{(i,j)} = \arg\max_{l} \; s(l, (i,j), \boldsymbol{y}, \boldsymbol{x})$$

Doing this for each edge in the tree would produce the final output. Such a model could easily be trained using the provided training data for each language. However, it might be advantageous to know the labels of other nearby edges. For instance, if we consider a head x_i with dependents $x_{j_1}, \ldots, x_{j_M}$, it is often the case that many of these dependencies will have correlated labels. To model this we treat the labeling of the edges $(i, j_1), \ldots, (i, j_M)$ as a sequence labeling problem,

$$(l_{(i,j_1)}, \ldots, l_{(i,j_M)}) = \bar{l} = \arg\max_{\bar{l}} \; s(\bar{l}, i, \boldsymbol{y}, \boldsymbol{x})$$

We use a first-order Markov factorization of the score

$$\bar{l} = \arg\max_{\bar{l}} \; \sum_{m=2}^{M} s(l_{(i,j_m)}, l_{(i,j_{m-1})}, i, \boldsymbol{y}, \boldsymbol{x})$$

in which each factor is the score of labeling the adjacent edges (i, j_m) and (i, j_{m-1}) in the tree $\boldsymbol{y}$. We attempted higher-order Markov factorizations but they did not improve performance uniformly across languages and training became significantly slower.

For score functions, we use simple dot products between high dimensional feature representations and a weight vector

$$\begin{aligned} s(l_{(i,j_m)}, l_{(i,j_{m-1})}, i, \boldsymbol{y}, \boldsymbol{x}) &= \\ &\mathbf{w} \cdot \mathbf{f}(l_{(i,j_m)}, l_{(i,j_{m-1})}, i, \boldsymbol{y}, \boldsymbol{x}) \end{aligned}$$

Assuming we have an appropriate feature representation, we can find the highest scoring label sequence with Viterbi's algorithm. We use the MIRA online learner to set the weights (Crammer and Singer, 2003; McDonald et al., 2005a) since we found it trained quickly and provide good performance. Furthermore, it made the system homogeneous in terms of learning algorithms since that is what is used to train our unlabeled parser (McDonald and Pereira, 2006). Of course, we have to define a set of suitable features. We used the following:

- **Edge Features:** Word/pre-suffix/part-of-speech (POS)/morphological feature identity of the head and the dependent (affix lengths 2 and 3). Does the head and its dependent share a prefix/suffix? Attachment direction. What morphological features do head and dependent have the same value for? Is the dependent the first/last word in the sentence?

- **Sibling Features:** Word/POS/pre-suffix/morphological feature identity of the dependent's nearest left/right siblings in the tree (siblings are words with same parent in the tree). Do any of the dependent's siblings share its POS?

- **Context Features:** POS tag of each intervening word between head and dependent. Do any of the words between the head and the dependent have a parent other than the head? Are any of the words between the head and the dependent not a descendant of the head (i.e. non-projective edge)?

- **Non-local:** How many children does the dependent have? What morphological features do the grandparent and the dependent have identical values? Is this the left/rightmost dependent for the head? Is this the first dependent to the left/right of the head?

Various conjunctions of these were included based on performance on held-out data. Note that many of these features are beyond the scope of the edge based factorizations of the unlabeled parser. Thus a joint model of parsing and labeling could not easily include them without some form of re-ranking or approximate parameter estimation.

4 Results

We trained models for all 13 languages provided by the CoNLL organizers (Buchholz et al., 2006). Based on performance from a held-out section of the training data, we used non-projective parsing algorithms for Czech, Danish, Dutch, German, Japanese, Portuguese and Slovene, and projective parsing algorithms for Arabic, Bulgarian, Chinese, Spanish, Swedish and Turkish. Furthermore, for Arabic and Spanish, we used lemmas instead of inflected word forms, again based on performance on held-out data[1].

DATA SET	UA	LA
ARABIC	79.3	66.9
BULGARIAN	92.0	87.6
CHINESE	91.1	85.9
CZECH	87.3	80.2
DANISH	90.6	84.8
DUTCH	83.6	79.2
GERMAN	90.4	87.3
JAPANESE	92.8	90.7
PORTUGUESE	91.4	86.8
SLOVENE	83.2	73.4
SPANISH	86.1	82.3
SWEDISH	88.9	82.5
TURKISH	74.7	63.2
AVERAGE	87.0	80.8

Table 1: Dependency accuracy on 13 languages. Unlabeled (UA) and Labeled Accuracy (LA).

Results on the test set are given in Table 1. Performance is measured through unlabeled accuracy, which is the percentage of words that modify the correct head in the dependency graph, and labeled accuracy, which is the percentage of words that modify the correct head *and* label the dependency edge correctly in the graph. These results show that the discriminative spanning tree parsing framework (McDonald et al., 2005b; McDonald and Pereira, 2006) is easily adapted across all these languages. Only Arabic, Turkish and Slovene have parsing accuracies significantly below 80%, and these languages have relatively small training sets and/or are highly inflected with little to no word order constraints. Furthermore, these results show that a two-stage system can achieve a relatively high performance. In fact, for every language our models perform significantly higher than the average performance for all the systems reported in Buchholz et al. (2006).

For the remainder of the paper we provide a general error analysis across a wide set of languages plus a detailed error analysis of Spanish and Arabic.

5 General Error Analysis

Our system has several components, including the ability to produce non-projective edges, sequential

[1] Using the non-projective parser for all languages does not effect performance significantly. Similarly, using the inflected word form instead of the lemma for all languages does not change performance significantly.

SYSTEM	UA	LA
N+S+M	86.3	79.7
P+S+M	85.6	79.2
N+S+B	85.5	78.6
N+A+M	86.3	79.4
P+A+B	84.8	77.7

Table 2: Error analysis of parser components averaged over Arabic, Bulgarian, Danish, Dutch, Japanese, Portuguese, Slovene, Spanish, Swedish and Turkish. N/P: Allow non-projective/Force projective, S/A: Sequential labeling/Atomic labeling, M/B: Include morphology features/No morphology features.

assignment of edge labels instead of individual assignment, and a rich feature set that incorporates morphological properties when available. The benefit of each of these is shown in Table 2. These results report the average labeled and unlabeled precision for the 10 languages with the smallest training sets. This allowed us to train new models quickly.

Table 2 shows that each component of our system does not change performance significantly (rows 2-4 versus row 1). However, if we only allow projective parses, do not use morphological features and label edges with a simple atomic classifier, the overall drop in performance becomes significant (row 5 versus row 1). Allowing non-projective parses helped with freer word order languages like Dutch (78.8%/74.7% to 83.6%/79.2%, unlabeled/labeled accuracy). Including rich morphology features naturally helped with highly inflected languages, in particular Spanish, Arabic, Turkish, Slovene and to a lesser extent Dutch and Portuguese. Derived morphological features improved accuracy in all these languages by 1-3% absolute.

Sequential classification of labels had very little effect on overall labeled accuracy (79.4% to 79.7%)[2]. The major contribution was in helping to distinguish subjects, objects and other dependents of main verbs, which is the most common labeling error. This is not surprising since these edge labels typically are the most correlated (i.e., if you already know which noun dependent is the subject, then it should be easy to find the object). For instance, sequential labeling improves the labeling of objects from 81.7%/75.6% to 84.2%/81.3% (labeled precision/recall) and the labeling of subjects from 86.8%/88.2% to 90.5%/90.4% for Swedish. Similar improvements are common across all languages, though not as dramatic. Even with this improvement, the labeling of verb dependents remains the highest source of error.

6 Detailed Analysis

6.1 Spanish

Although overall unlabeled accuracy is 86%, most verbs and some conjunctions attach to their head words with much lower accuracy: 69% for main verbs, 75% for the verb *ser*, and 65% for coordinating conjunctions. These words form 17% of the test corpus. Other high-frequency word classes with relatively low attachment accuracy are prepositions (80%), adverbs (82%) and subordinating conjunctions (80%), for a total of another 23% of the test corpus. These weaknesses are not surprising, since these decisions encode the more global aspects of sentence structure: arrangement of clauses and adverbial dependents in multi-clause sentences, and prepositional phrase attachment. In a preliminary test of this hypothesis, we looked at all of the sentences from a development set in which a main verb is incorrectly attached. We confirmed that the main clause is often misidentified in multi-clause sentences, or that one of several conjoined clauses is incorrectly taken as the main clause. To test this further, we added features to count the number of commas and conjunctions between a dependent verb and its candidate head. Unlabeled accuracy for all verbs increases from 71% to 73% and for all conjunctions from 71% to 74%. Unfortunately, accuracy for other word types decreases somewhat, resulting in no significant net accuracy change. Nevertheless, this very preliminary experiment suggests that wider-range features may be useful in improving the recognition of overall sentence structure.

Another common verb attachment error is a switch between head and dependent verb in phrasal verb forms like *dejan intrigar* or *qiero decir*, possibly because the non-finite verb in these cases is often a main verb in training sentences. We need to look more carefully at verb features that may be useful here, in particular features that distinguish finite and

[2] This difference was much larger for experiments in which gold standard unlabeled dependencies are used.

non-finite forms.

In doing this preliminary analysis, we noticed some inconsistencies in the reference dependency structures. For example, in the test sentence *Lo que decia Mae West de si misma podríamos decirlo también los hombres:...*, *decia*'s head is given as *decirlo*, although the main verbs of relative clauses are normally dependent on what the relative modifies, in this case the article *Lo*.

6.2 Arabic

A quick look at unlabeled attachment accuracies indicate that errors in Arabic parsing are the most common across all languages: prepositions (62%), conjunctions (69%) and to a lesser extent verbs (73%). Similarly, for labeled accuracy, the hardest edges to label are for dependents of verbs, i.e., subjects, objects and adverbials. Note the difference in error between the unlabeled parser and the edge labeler: the former makes mistakes on edges into prepositions, conjunctions and verbs, and the latter makes mistakes on edges into nouns (subject/objects). Each stage by itself is relatively accurate (unlabeled accuracy is 79% and labeling accuracy[3] is also 79%), but since there is very little overlap in the kinds of errors each makes, overall labeled accuracy drops to 67%. This drop is not nearly as significant for other languages.

Another source of potential error is that the average sentence length of Arabic is much higher than other languages (around 37 words/sentence). However, if we only look at performance for sentences of length less than 30, the labeled accuracy is still only 71%. The fact that Arabic has only 1500 training instances might also be problematic. For example if we train on 200, 400, 800 and the full training set, labeled accuracies are 54%, 60%, 62% and 67%. Clearly adding more data is improving performance. However, when compared to the performance of Slovene (1500 training instances) and Spanish (3300 instances), it appears that Arabic parsing is lagging.

7 Conclusions

We have presented results showing that the spanning tree dependency parsing framework of McDonald et al. (McDonald et al., 2005b; McDonald and Pereira, 2006) generalizes well to languages other than English. In the future we plan to extend these models in two ways. First, we plan on examining the performance difference between two-staged dependency parsing (as presented here) and joint parsing plus labeling. It is our hypothesis that for languages with fine-grained label sets, joint parsing and labeling will improve performance. Second, we plan on integrating any available morphological features in a more principled manner. The current system simply includes all morphological bi-gram features. It is our hope that a better morphological feature set will help with both unlabeled parsing and labeling for highly inflected languages.

[3] Labeling accuracy is the percentage of words that correctly label the dependency between the head that they modify, even if the right head was not identified.

References

A. Abeillé, editor. 2003. *Treebanks: Building and Using Parsed Corpora*, volume 20 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht.

S. Afonso, E. Bick, R. Haber, and D. Santos. 2002. "Floresta sintá(c)tica": a treebank for Portuguese. In *Proc. of the Third Intern. Conf. on Language Resources and Evaluation (LREC)*, pages 1698–1703.

N. B. Atalay, K. Oflazer, and B. Say. 2003. The annotation process in the Turkish treebank. In *Proc. of the 4th Intern. Workshop on Linguistically Interpreteted Corpora (LINC)*.

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

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER treebank. In *Proc. of the First Workshop on Treebanks and Linguistic Theories (TLT)*.

S. Buchholz, E. Marsi, A. Dubey, and Y. Krymolowski. 2006. CoNLL-X shared task on multilingual dependency parsing. SIGNLL.

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.

M. Civit Torruella and M^a A. Martí Antonín. 2002. Design principles for a Spanish treebank. In *Proc. of the First Workshop on Treebanks and Linguistic Theories (TLT)*.

K. Crammer and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *JMLR*.

Y. Ding and M. Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. ACL*.

S. Džeroski, T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky, and A. Žele. 2006. Towards a Slovene dependency treebank. In *Proc. of the Fifth Intern. Conf. on Language Resources and Evaluation (LREC)*.

A. Haghighi, A. Ng, and C. Manning. 2005. Robust textual inference via graph matching. In *Proc. HTL-EMNLP*.

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.

R. Hudson. 1984. *Word Grammar*. Blackwell.

Y. Kawata and J. Bartels. 2000. Stylebook for the Japanese treebank in VERBMOBIL. Verbmobil-Report 240, Seminar für Sprachwissenschaft, Universität Tübingen.

M. T. Kromann. 2003. The Danish dependency treebank and the underlying linguistic theory. In *Proc. of the Second Workshop on Treebanks and Linguistic Theories (TLT)*.

R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. EACL*.

R. McDonald, K. Crammer, and F. Pereira. 2005a. Online large-margin training of dependency parsers. In *Proc. ACL*.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proc. HLT-EMNLP*.

R. McDonald. 2006. Discriminative sentence compression with soft syntactic constraints. In *Proc. EACL*.

I.A. Mel′čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.

J. Nilsson, J. Hall, and J. Nivre. 2005. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. In *Proc. of the NODALIDA Special Session on Treebanks*.

J. Nivre. 2005. Dependency grammar and dependency parsing. Technical Report MSI report 05133, Växjö University: School of Mathematics and Systems Engineering.

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

K. Simov and P. Osenova. 2003. Practical annotation scheme for an HPSG treebank of Bulgarian. In *Proc. of the 4th Intern. Workshop on Linguistically Interpreteted Corpora (LINC)*, pages 17–24.

K. Simov, P. Osenova, A. Simov, and M. Kouylekov. 2005. Design and implementation of the Bulgarian HPSG-based treebank. In *Journal of Research on Language and Computation – Special Issue*, pages 495–522. Kluwer Academic Publishers.

L. van der Beek, G. Bouma, R. Malouf, and G. van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*.