Dependency Parsing with Reference to Slovene, Spanish and Swedish

Simon Corston-Oliver
Natural Language Processing
Microsoft Research
One Microsoft Way
Redmond WA 98052
simonco@microsoft.com

Anthony Aue
Natural Language Processing
Microsoft Research
One Microsoft Way
Redmond WA 98052
anthaue@microsoft.com

Abstract

We describe a parser used in the CoNLL 2006 Shared Task, "Multingual Dependency Parsing." The parser first identifies syntactic dependencies and then labels those dependencies using a maximum entropy classifier. We consider the impact of feature engineering and the choice of machine learning algorithm, with particular focus on Slovene, Spanish and Swedish.

1 Introduction

The system that we submitted for the CoNLL 2006 Shared Task, "Multingual Dependency Parsing," (Buchholz et al., 2006) is a two stage pipeline. The first stage identifies unlabeled directed dependencies using an extension of the parser described in (Corston-Oliver et al., 2006). The second stage is a maximum entropy classifier that labels the directed dependencies. The system was trained on the twelve obligatory languages, as well as the optional language, Bulgarian (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).

Table 1 presents the results of the system described in the current paper on the CoNLL shared task, including the optional evaluation on Bulgarian. For Slovene, we ranked second with a labeled dependency accuracy of 72.42%. This was not statistically significantly different from the top-ranked score of 73.44%. For Spanish, our labeled dependency accuracy of 80.36% is within 0.1% of the third-ranked score of 80.46%. Our unlabeled dependency accuracy for Swedish was the best of all the systems at 89.54%. Our labeled accuracy for Swedish, however, at 79.69%, fell far short of the third-best score of 82.31%. We therefore focus on Swedish when considering the impact of our choice of learning algorithm on our label accuracy.

Language	Unlabeled Attachment	Labeled Attachment
Arabic	78.40	63.53
Bulgarian	90.09	83.36
Chinese	90.00	79.92
Czech	83.02	74.48
Danish	87.94	81.74
Dutch	74.83	71.43
German	87.20	83.47
Japanese	92.84	89.95
Portugese	88.96	84.59
Slovene	81.77	72.42
Spanish	84.87	80.36
Swedish	89.54	79.69
Turkish	73.11	61.74

Table 1: Results on CoNLL 2006 shared task.

2 Data

We divided the shared data into training and development test sets, using larger development test sets

Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X),
pages 196–200, New York City, June 2006.

for the languages supplied with more data. The development test set consisted of 250 sentences for Arabic, Slovene, Spanish and Turkish, 500 sentences for Danish and Portuguese, and 1,000 sentences for the other languages.

3 The Parser

The baseline parser predicts unlabeled directed dependencies. As described in (Corston-Oliver et al., 2006), we reimplemented the parser described in (McDonald et al., 2005) and validated their results for Czech and English.

The parser finds the highest-scoring parse $\hat{y}$ among all possible parses $y \in Y$ for a given sentence:

$$\hat{y} = \arg\max_{y \in Y} s(y) \qquad (1)$$

The score s of a given parse y is the sum of the scores of all the dependency links (i,j) $\in y$:

$$s(y) = \sum_{(i,j) \in y} d(i,j) = \sum_{(i,j) \in y} \mathbf{w} \cdot \mathbf{f}(i,j) \qquad (2)$$

where the link (i,j) indicates a parent-child dependency between the token at position i and the token at position j. The score $d(i,j)$ of each dependency link (i,j) is further decomposed as the weighted sum of its features $\mathbf{f}(i,j)$.

To set **w**, we trained twenty averaged perceptrons on different shuffles of the training data, using the development test set to determine when the perceptrons had converged. The averaged perceptrons were then combined to make a Bayes Point Machine (Harrington et al., 2003). At both training and run time, edges are scored independently, and Eisner's $O(N^3)$ decoder (Eisner, 1996) is used to find the optimal parse. This decoder produces only projective analyses, although it does allow for analyses with multiple roots.

The features used for scoring the edges prior to applying Eisner's algorithm are extracted from each possible parent-child dependency. The features include the case-normalized original form and lemma[1] of each token , the part of speech (POS) tag of each token, the POS tag of each intervening token and of each token to the left and right of the parent and child. Additional features are created by combining these atomic features, as described in (McDonald et al., 2005). All features are in turn combined with the direction of attachment and the distance between tokens. Distance was discretized, with individual buckets for distances 0-4, a single bucket for 5-9, and a single bucket for 10+. In sections 3.1 and 3.2 we discuss the feature engineering we performed.

3.1 Part of Speech Features

We experimented with using the coarse POS tag and the fine POS tag. In our official submission, we used fine POS tags for all languages except Dutch and Turkish. For Dutch and Turkish, using the fine POS tag resulted in a reduction in unlabeled dependency accuracy of 0.12% and 0.43% respectively on the development test sets, apparently because of the sparsity of the fine POS tags. For German and Swedish, the fine and coarse POS tags are the same so using the fine POS tag had no effect. For other languages, using the fine POS tag showed modest improvements in unlabeled dependency accuracy.

For Swedish, we performed an additional manipulation on the POS tags, normalizing the distinct POS tags assigned to each verbal auxiliary and modal to a single tag "aux". For example, in the Swedish data all inflected forms of the verb "vara" ("be") are tagged as AV, and all inflected forms of the modal "måste" ("must") are tagged as MV. This normalization caused unlabeled dependency accuracy on the Swedish development set to improve from 89.23% to 89.45%.

3.2 Features for Root Identification

Analysis of the baseline parser's errors suggested the need for additional feature types to improve the identification of the root of the sentence. In particular, the parser was frequently making errors in identifying the root of periphrastic constructions involving an auxiliary verb or modal and a participle. In Germanic languages, for example, the auxiliary or modal typically occurs in second position in declarative main clauses or in initial position in cases of subject-aux inversion. We added a collection of features intended to improve the identification of the root. The hope was that improved root identification would have a positive cascading effect in the

[1] If no lemma was specified, we truncated the original form by taking the first two characters for Chinese words consisting of two characters or more and the first five characters for words consisting of five characters or more in the other languages.

identification of other dependencies, since a failure to correctly identify the root of the sentence usually means that the parse will have many other errors.

We extracted four feature types, the original form of the first and last tokens in the sentence and the POS of the first and last tokens in the sentence. These features were intended to identify declarative vs. interrogative sentences.

For each child and parent token being scored, we also noted the following four features: "child/parent is first non-punctuation token in sentence", "child/parent is second non-punctuation token in sentence". The features that identify the second token in the sentence were intended to improve the identification of verb-second phenomena. Of course, this is a linguistic oversimplification. Verb-second phenomena are actually sensitive to the order of constituents, not words. We therefore added four feature types that considered the sequence of POS tags to the left of the child or parent if they occurred within ten tokens of the beginning of the sentence and the sequence of POS tags to the right of the child or parent if they occurred within ten tokens of the end of the sentence.

We also added features intended to improve the identification of the root in sentences without a finite verb. For example, the Dutch training data contained many simple responses to a question-answering task, consisting of a single noun phrase. Four simple features were used "Child/Parent is the leftmost noun in the sentence", "Child/Parent is a noun but not the leftmost noun in the sentence". These features were combined with an indicator "Sentence contains/does not contain a finite verb".

Child or parent tokens that were finite verbs were flagged as likely candidates for being the root of the sentence if they were the leftmost finite verb in the sentence and not preceded by a subordinating conjunction or relative pronoun. Finite verbs were identified by POS tags and morphological features, e.g. in Spanish, verbs without the morphological feature "mod=n" were identified as finite, while in Portuguese the fine POS tag "v-fin" was used.

Similarly, various sets of POS tags were used to identify subordinating conjunctions or relative pronouns for different languages. For example, in Bulgarian the fine POS tag "pr" (relative pronoun) and "cs" (subordinating conjunction) were used. For Dutch, the morphological features "onder", "betr" and "voorinf" were used to identify subordinating conjunctions and relative pronouns.

These features wreaked havoc with Turkish, a verb-final language. For certain other languages, dependency accuracy measured on the development test set improved by a modest amount, with more dramatic improvements in root accuracy (F1 measure combining precision and recall for non-punctuation root tokens).

Since the addition of these features had been motivated by verb-second phenomena in Germanic languages, we were surprised to discover that the only Germanic language to demonstrate a marked improvement in unlabeled dependency accuracy was Danish, whose accuracy on the development set rose from 87.51% to 87.72%, while root accuracy F1 rose from 94.12% to 94.72%. Spanish showed a modest improvement in unlabeled dependency accuracy, from 85.08% to 85.13%, but root F1 rose from 80.08% to 83.57%.

The features described above for identifying the leftmost finite verb not preceded by a subordinating conjunction or relative pronoun did not improve Slovene unlabeled dependency accuracy, and so were not included in the set of root-identifying features in our Slovene CoNLL submission. Closer examination of the Slovene corpus revealed that periphrastic constructions consisting of one or more auxiliaries followed by a participle were annotated with the participle as the head, whereas for other languages in the shared task the consensus view appears to be that the auxiliary should be annotated as the head. Singling out the leftmost finite verb in Slovene when a participle ought to be selected as the root of the sentence is therefore counter-productive. The other root identification features did improve root F1 in Slovene. Root F1 on the development test set rose from 45.82% to 46.43%, although overall unlabeled dependency accuracy on the development test set fell slightly from 80.24% to 79.94%.

3.3 Morphological Features

As the preceding discussion shows, morphological information was occasionally used to assist in making finer-grained POS distinctions than were made in the POS tags, e.g., for distinguishing subordinating vs. coordinating conjunctions. Aside from these surgical uses of the morphological information present in the CoNLL data, morphology was not explicitly used by the baseline parser. For example, there were no features that considered subject-verb agreement nor agreement of an adjective with the number or lexical gender of the noun it modified. However, it is possible that morphological information influenced the training of edge weights if the information was implicit in the POS tags.

4 The Dependency Labeler

4.1 Classifier

We used a maximum entropy classifier (Berger et al., 1996) to assign labels to the unlabeled dependencies produced by the Bayes Point Machine. We used the same training and development test split that was used to train the dependency parser. We chose to use maximum entropy classifiers because they can be trained relatively quickly while still offering reasonable classification accuracy and are robust in the face of large numbers of superfluous features, a desirable property given the requirement that the same parser handle multiple languages. Furthermore, maximum entropy classifiers provide good probability distributions over class labels. This was important to us because we had initially hoped to find the optimal set of dependency labels for the children of a given node by modeling the probability of each set of labels conditioned on the lemma and POS of the parent. For example, labeling each dependant of a parent node independently might result in three OBJECT relations dependent on a single verb; modeling sets of relations ought to prevent this. Unfortunately, this approach did not outperform labeling each node independently.

Therefore, the system we submitted labeled each dependency independently, using the most probable label from the maximum entropy classifier. We have noted in previous experiments that our SVM implementation often gives better one-best classification accuracy than our maximum entropy implementation, but did not have time to train SVM classifiers.

To see how much the choice of classification algorithm affected our official results, we trained a linear SVM classifier for Swedish after the competition had ended, tuning parameters on the development test set. As noted in section 1, our system scored highest for Swedish in unlabeled dependency accuracy at 89.54% but fell well short of the third-ranked system when measuring labeled dependency accuracy. Using an SVM classifier instead of a maximum entropy classifier, Swedish label accuracy rose from 82.33% to 86.06%, and labeled attachment accuracy rose from 79.69% to 82.95%, which falls between the first-ranked score of 84.58% and the second-ranked score of 82.55%. Similarly, Japanese label accuracy rose from 93.20% to 93.96%, and labeled attachment accuracy rose from 89.95% to 90.77% when we replaced a maximum entropy classifier with an SVM. This labeled attachment result of 90.77% is comparable to the official second place result of 90.71% for Japanese. We conclude that a two stage pipeline such as ours, in which the second stage labels dependencies in isolation, is greatly impacted by the choice of classifier.

4.2 Features Used for Labeling

We extracted features from individual nodes in the dependency tree, parent-child features and features that took nodes other than the parent and child into account.

The features extracted from each individual parent and child node were the original surface form, the lemma (see footnote 1 above), the coarse and fine POS tags and each morphological feature.

The parent-child features are the direction of modification, the combination of the parent and child lemmata, all combinations of parent and child lemma and coarse POS tag (e.g. child lemma combined with coarse POS tag of the parent) and all pairwise combinations of parent and child morphology features (e.g. parent is feminine and child is plural).

Additional features were verb position (whether the parent or child is the first or last verb in the sentence), coarse POS and lemma of the left and right neighbors of the parent and child, coarse POS and lemma of the grandparent, number and coarse POS tag sequence of siblings to the left and to the right of the child, total number of siblings of the child, number of tokens governed by child, whether the parent has a verbal ancestor, lemma and morphological features of the verb governing the child (if any), and coarse POS tag combined with relative offset of each sibling (e.g., the sibling two to the left of the child is a determiner).

For Slovene, the label accuracy using all of the features above was 81.91%. We retrained our maximum entropy classifier by removing certain classes of features in order to determine their contribution. Removing the weight features caused a notable drop, with label accuracy on the development test set falling 0.52% to 81.39%. Removing the grandparent features (but including weight features) caused an even greater drop of 1.03% to 80.88%. One place where the grandparent features were important was in distinguishing between Adv and Atr relations. It appears that the relation between a noun and its governing preposition or between a verb and its governing conjunction is sensitive to the part of speech of the grandparent. For example, we observed a number of cases where the relation between a noun and its governing preposition had been incorrectly labeled as Adv when it should have been Atr. The addition of grandparent features allowed the classifier to make the distinction by looking at the POS of the grandparent; when the POS was noun, the classifier tended to correctly choose the Atr label.

5 Conclusion

We have described a two stage pipeline that first predicts directed unlabeled dependencies and then labels them. The system performed well on Slovene, Spanish and Swedish. Feature engineering played an important role both in predicting dependencies and in labeling them. Finally, replacing the maximum entropy classifier used to label dependencies with an SVM improves upon our official results.

References

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

S. Buchholz, E. Marsi, A. Dubey, and Y. Krymolowski. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of the Tenth Conf. on Computational Natural Language Learning (CoNLL-X)*. SIGNLL.

Simon Corston-Oliver, Anthony Aue, Kevin Duh, and Eric Ringger. 2006. Multilingual dependency parsing using bayes point machines. In *Proc. of HLT-NAACL 2006*.

J. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. of the 16th Intern. Conf. on Computational Linguistics (COLING)*, pages 340–345.

Edward Harrington, Ralf Herbrich, Jyrki Kivinen, John C. Platt, and Robert C. Williamson. 2003. Online bayes point machines. In *Proceedings of Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 241–252.

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