Investigating Multilingual Dependency Parsing

Richard Johansson  
Department of Computer Science  
LTH, Lund University  
221 00 Lund, Sweden  
Richard.Johansson@cs.lth.se

Pierre Nugues  
Department of Computer Science  
LTH, Lund University  
221 00 Lund, Sweden  
Pierre.Nugues@cs.lth.se

Abstract
In this paper, we describe a system for the CoNLL-X shared task of multilingual dependency parsing. It uses a baseline Nivre’s parser (Nivre, 2003) that first identifies the parse actions and then labels the dependency arcs. These two steps are implemented as SVM classifiers using LIBSVM. Features take into account the static context as well as relations dynamically built during parsing.

We experimented two main additions to our implementation of Nivre’s parser: N-best search and bidirectional parsing. We trained the parser in both left-right and right-left directions and we combined the results. To construct a single-head, rooted, and cycle-free tree, we applied the Chu-Liu/Edmonds optimization algorithm. We ran the same algorithm with the same parameters on all the languages.

1 Nivre’s Parser
Nivre (2003) proposed a dependency parser that creates a projective and acyclic graph. The parser is an extension to the shift–reduce algorithm. As with the regular shift–reduce, it uses a stack $S$ and a list of input words $W$. However, instead of finding constituents, it builds a set of arcs $G$ representing the graph of dependencies.

Nivre’s parser uses two operations in addition to shift and reduce: left-arc and right-arc. Given a sequence of words, possibly annotated with their part of speech, parsing simply consists in applying a sequence of operations: left-arc (la), right-arc (ra), reduce (re), and shift (sh) to the input sequence.

2 Parsing an Annotated Corpus
The algorithm to parse an annotated corpus is straightforward from Nivre’s parser and enables us to obtain, for any projective sentence, a sequence of actions taken in the set \{la, ra, re, sh\} that parses it. At a given step of the parsing process, let $TOP$ be the top of the stack and $FIRST$, the first token of the input list, and $arc$, the relation holding between a head and a dependent.

1. if $arc(TOP,FIRST) \in G$, then ra;
2. else if $arc(FIRST, TOP) \in G$, then la;
3. else if $\exists k \in Stack, arc(FIRST, k) \in G$ or $arc(k, FIRST) \in G$, then re;
4. else sh.

Using the first sentence of the Swedish corpus as input (Table 1), this algorithm produces the sequence of 24 actions: sh, sh, la, ra, re, la, sh, sh, sh, la, la, ra, ra, sh, la, re, ra, ra, re, re, ra, re, and ra (Table 2).

3 Adapting Nivre’s Algorithm to Machine–Learning
3.1 Overview
We used support vector machines to predict the parse action sequence and a two step procedure to
produce the graph. We first ran the classifier to select unlabeled actions, la, ra, sh, re. We then ran a second classifier to assign a function to ra and la parse actions.

We used the LIBSVM implementation of the SVM learning algorithm (Chang and Lin, 2001). We used the Gaussian kernel throughout. Optimal values for the parameters ($C$ and $\gamma$) were found using a grid search. The first predicted action is not always possible, given the parser’s constraints. We trained the model using probability estimates to select the next possible action.

### 3.2 Feature Set

We used the following set of features for the classifiers:

- The word, POS, and dependency type of the leftmost child of TOP to TOP, if any
- The word, POS, and dependency type of the rightmost child of TOP to TOP, if any
- The word, POS, and dependency type of the leftmost child of FIRST to FIRST, if any

For the POS, we used the Coarse POS, the Fine POS, and all the features (encoded as boolean flags). We did not use the lemma.

### 4 Extensions to Nivre’s Algorithm

#### 4.1 N-best Search

We extended Nivre’s original algorithm with a beam search strategy. For each action, la, ra, sh and re,
we computed a probability score using LIBSVM. These scores can then be used to carry out an $N$-best search through the set of possible sequences of actions.

We measured the improvement over a best-first strategy incrementing values of $N$. We observed the largest difference between $N = 1$ and $N = 2$, then leveling off and we used the latter value.

4.2 Bidirectionality and Voting
Tesnière (1966) classified languages as centrifuge (head to the left) and centripetal (head to the right) in a table (page 33 of his book) that nearly exactly fits corpus evidence from the CONLL data. Nivre’s parser is inherently left-right. This may not fit all the languages. Some dependencies may be easier to capture when proceeding from the reverse direction. Jin et al. (2005) is an example of it for Chinese, where the authors describe an adaptation of Nivre’s parser to bidirectionality.

We trained the model and ran the algorithm in both directions (left to right and right to left). We used a voting strategy based on probability scores. Each link was assigned a probability score (simply by using the probability of the $la$ or $ra$ actions for each link). We then summed the probability scores of the links from all four trees. To construct a single-head, rooted, and cycle-free tree, we finally applied the Chu-Liu/Edmonds optimization algorithm (Chu and Liu, 1965; Edmonds, 1967).

5 Analysis
5.1 Experimental Settings
We trained the models on “projectivized” graphs following Nivre and Nilsson (2005) method. We used the complete annotated data for nine languages. Due to time limitations, we could not complete the training for three languages, Chinese, Czech, and German.

5.2 Overview of the Results
We parsed the 12 languages using exactly the same algorithms and parameters. We obtained an average score of 74.93 for the labeled arcs and of 80.39 for the unlabeled ones (resp. 74.98 and 80.80 for the languages where we could train the model using the complete annotated data sets). Table 3 shows the results per language. As a possible explanation of the differences between languages, the three lowest figures correspond to the three smallest corpora. It is reasonable to assume that if corpora would have been of equal sizes, results would have been more similar. Czech is an exception to this rule that applies to all the participants. We have no explanation for this. This language, or its annotation, seems to be more complex than the others.

The percentage of nonprojective arcs also seems to play a role. Due to time limitations, we trained the Dutch and German models with approximately the same quantity of data. While both languages are closely related, the Dutch corpus shows twice as much nonprojective arcs. The score for Dutch is significantly lower than for German.

Our results across the languages are consistent with the other participants’ mean scores, where we are above the average by a margin of 2 to 3% except for Japanese and even more for Chinese where we obtain results that are nearly 7% less than the average for labeled relations. Results are similar for unlabeled data. We retrained the data with the complete Chinese corpus and you obtained 74.41 for the labeled arcs, still far from the average. We have no explanation for this dip with Chinese.

5.3 Analysis of Swedish and Portuguese Results
5.3.1 Swedish
We obtained a score of 78.13% for the labeled attachments in Swedish. The error breakdown shows significant differences between the parts of speech. While we reach 89% of correct head and dependents for the adjectives, we obtain 55% for the prepositions. The same applies to dependency types, 84% precision for subjects, and 46% for the OA type of prepositional attachment.

There is no significant score differences for the left and right dependencies, which could attributed to the bidirectional parsing (Table 4). Distance plays a dramatic role in the error score (Table 5). Prepositions are the main source of errors (Table 6).

5.3.2 Portuguese
We obtained a score 84.57% for the labeled attachments in Portuguese. As for Swedish, error distribution shows significant variations across the
parts of speech, with a score of 94% for adjectives and only 67% for prepositions.

As for Swedish, there is no significant score differences for the left and right dependencies (Table 7). Distance also degrades results but the slope is not as steep as with Swedish (Table 8). Prepositions are also the main source of errors (Table 9).

5.4 Acknowledgments

This work was made possible because of the annotated corpora that were kindly provided to us: Arabic (Hajič et al., 2004), Bulgarian (Simov et al., 2005; Simov and Osenova, 2003), Chinese (Chen et al., 2003), Czech (Böhmová et al., 2003), Danish (Kromann, 2003), Dutch (van der Beek et al., 2002), German (Brants et al., 2002), Japanese (Kawata and Bartels, 2000), Portuguese (Afonso et al., 2002), Slovene (Džeroski et al., 2006), Spanish (Civit Torruella and Martí Antonín, 2002), Swedish (Nilsson et al., 2005), and Turkish (Oflazer et al., 2003; Atalay et al., 2003).
Table 9: Focus words where most of the errors occur. Portuguese.

<table>
<thead>
<tr>
<th>Word</th>
<th>POS</th>
<th>Any</th>
<th>Head</th>
<th>Dep</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>em</td>
<td>prp</td>
<td>66</td>
<td>38</td>
<td>47</td>
<td>19</td>
</tr>
<tr>
<td>de</td>
<td>prp</td>
<td>51</td>
<td>37</td>
<td>35</td>
<td>21</td>
</tr>
<tr>
<td>a</td>
<td>prp</td>
<td>46</td>
<td>30</td>
<td>39</td>
<td>23</td>
</tr>
<tr>
<td>e</td>
<td>conj</td>
<td>28</td>
<td>28</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>para</td>
<td>prp</td>
<td>21</td>
<td>13</td>
<td>18</td>
<td>10</td>
</tr>
</tbody>
</table>

References


