Self-training for Swedish Dependency Parsing – Initial Results and Analysis

Anders Björkelund[§] **Özlem Çetinoğlu**[§] **Agnieszka Faleńska**[◇,§] **Richárd Farkas**[†]
Thomas Müller[‡] **Wolfgang Seeker**[§] **Zsolt Szántó**[†]

[§]Institute for Natural Language Processing, University of Stuttgart, Germany
[◇]Institute of Computer Science, University of Wrocław, Poland
[†]Department of Informatics, University of Szeged, Hungary
[‡]Center for Information and Language Processing, University of Munich, Germany
anders@ims.uni-stuttgart.de

1. Introduction

Supervised learning techniques rely on the availability of annotated data for training. It is well-known that, in a general sense, the more training data, the better (Domingos, 2012). Creating such training data is, however, a time-consuming and tedious task. A simple idea of extending a training set using additional unlabeled data is **self-training**, where a supervised learner is applied on unlabeled data and then re-trained on the combination of the original training data and the automatically annotated data (McClosky et al., 2006).

Self-training is an appealing method as it is both cheap and simple. Unfortunately the method rarely works as intended and often does not yield any improvements as shown, e.g., for POS-tagging (Clark et al., 2003) and constituency parsing (Charniak, 1997). A common hypothesis for these negative results is that errors are amplified when the supervised learner is re-trained on its own output. However, McClosky et al. (2006) show positive improvements using a two-stage constituency parser that uses a standard PCFG followed by parse tree reranking. They apply the two-stage parser to unlabeled data and then use the output to re-train the PCFG. Re-training the reranker on the auto-parsed data does not render any improvements though.

The recent 2014 SPMRL Shared Task (Seddah et al., 2014) was devoted to (constituency and dependency) parsing of 9 morphologically rich languages in a supervised setting. Additionally, unlabeled data was provided to the participants with the hope that this could be used to increase parsing performance. In our contribution (Björkelund et al., 2014) to the dependency parsing track, we made experiments with self-training. For most languages the self-trained parsers performed roughly equal to the baseline parsers, corroborating previous results on self-training. However, in the case of Swedish we observed considerable improvements using self-training. In this paper we make an initial analysis of the differences between a baseline and self-trained parser. Our results indicate that, surprisingly, the self-trained parser does not improve on unknown words that are covered by auto-parsed data. Moreover, the improvement seems to apply to most part of speech tags and does not seem to be limited to certain parts of speech or specific linguistic phenomena.

2. Experimental Setup

For all experiments we use the mate parser (Bohnet, 2010), which is a state-of-the-art second-order graph-based parser. The parser has been further augmented to utilize features from dependency-based supertags following Ouchi et al. (2014). The data originates from the Talbanken corpus (Nivre et al., 2006) and we use the train/dev/test split from the 2014 SPMRL Shared Task. Using the training set, we trained the mate parser as the baseline.

The unlabeled data we used was also provided by the shared task organizers and originate from the PAROLE corpus, which is about 1.6 million sentences. While this is orders of magnitude larger than the original training set (which is 5,000 sentences), we wanted to keep the training set balanced between gold standard and automatic annotations. We therefore selected 5,000 additional auto-parsed sentences and added them to the original training set thereby doubling the amount of training data. The 5,000 auto-parsed sentences were selected as follows: We filtered the PAROLE corpus according to a number of criteria such as sentence length being between 5 and 20 tokens, the sentences contain at most 2 tokens not seen in the training data and so on (the specific filtering criteria are described in (Björkelund et al., 2014)). We then parsed 400,000 of the remaining sentences with both the baseline parser and the TurboParser (Martins et al., 2010). The two parsers had identical analyses for roughly a third of the sentences, from which we sampled 5,000. The mate parser was then re-trained on the combination of the training set and the 5,000 auto-parsed sentences. Since the mate parser uses a perceptron learning algorithm and does no internal shuffling of training instances, the 10,000 sentences were shuffled before training in order to have a less skewed training set.

Overall UAS and LAS results for the baseline parser and the self-trained parser on the development and test sets are shown in Table 1. All improvements are significant at the 0.05 level using a randomized approximation test.

2.1 Analysis

We now present a further analysis on the development set. It should be noted that the development set is relatively small (c. 9,400 tokens and 494 sentences) but recall that the overall improvements are significant. Since the data set is very small, we report absolute counts of correctly labeled

	Development		Test	
	UAS	LAS	UAS	LAS
Baseline	83.51	77.25	86.99	80.69
Self-trained	84.23	78.10	87.42	81.27

Table 1: Overall results of baseline and self-trained parsers on the development and test sets.

	Known	Known$_{+st}$	Unknown
Total	8,229	169	941
Baseline	6,308	140	766
Self-trained	+75	-3	+8

Table 2: Breakdown of correct labeled attachments over known, known from self-training, and unknown words.

Tag	Total	Baseline	Self-trained
ADJ	872	740	+1
ADP	985	575	-1
ADV	859	602	+15
CONJ	658	401	+13
DET	482	453	-3
NOUN	2,125	1,801	+7
NUM	69	49	+3
PRON	799	695	+13
PRT	207	147	+12
VERB	1,343	1,105	+6
X	12	5	0
.	928	641	+14

Table 3: Breakdown of improvements by gold POS.

Bucket	Total	Baseline	Self-trained
1	4,517	3,846	+30
2-3	2,788	2,159	+8
4-5	781	480	+24
6-10	708	404	+10
11+	514	325	+8

Table 4: Breakdown of improvements by bucketed gold dependency length.

and attached tokens (i.e., the correctness criterion used for the LAS metric) rather than percentages.

One hypothesis for the improvement is that the self-trained parser is able to pick up information about unknown words that are not in the training data but in the auto-parsed data. Table 2 shows the absolute number of correctly labeled attachments broken down by whether a word is known from the training data, known from the self-training data, or completely unknown. The table suggests that the parser improves on known and unknown words, but not on the words that were added into the training data through the self-training procedure. A further analysis looking at the inverse, i.e., dependents of words that are unknown or only known from self-training also gives the same picture (these numbers are omitted from the table due to space restrictions). We thus find that the numbers do not confirm our hypothesis.

Table 3 shows the total number of tokens and the performance of the baseline and self-trained parsers broken down by gold standard part-of-speech tag (mapped down to the universal tag set, cf. Petrov et al. (2012)). While the differences are generally very small, the table does not point in a specific direction, but rather that improvements occur across most tags. Similarly, Table 4 shows an accuracy breakdown by bucketed distance in tokens (in the gold standard). Also here the results do not suggest a particular category that profits from self-training, but rather that it is helpful all across the board.

3. Conclusion

We presented experiments on self-training of a dependency parser for Swedish, showing that self-training significantly outperforms a baseline parser. Further analysis indicates that the self-trained parser improves on known and unknown words, but not new words that are introduced through the auto-parsed data. The improvements also do not seem to be associated with certain parts of speech or arc length, but rather occur across the board.

Acknowledgements

Agnieszka Faleńska is funded by the Project International computer science and applied mathematics for business study program at the University of Wrocław co-financed with EU funds within the European Social Fund (POKL.04.01.01-00-005/13). Richárd Farkas and Zsolt Szántó are funded by the EU and the European Social Fund through the project FuturICT.hu (TÁMOP-4.2.2.C-11/1/KONV-2012-0013). Thomas Müller is supported by a Google Europe Fellowship in NLP. The remaining authors are funded by the Deutsche Forschungsgemeinschaft (DFG) via the SFB 732, projects D2 and D8 (PI: Jonas Kuhn).

References

Anders Björkelund, Özlem Çetinoğlu, Agnieszka Faleńska, Richárd Farkas, Thomas Müller, Wolfgang Seeker, and Zsolt Szántó. 2014. The IMS-Wrocław-Szeged-CIS entry at the SPMRL 2014 Shared Task: Reranking and Morphosyntax meet Unlabeled Data. In *Notes of the SPMRL 2014 Shared Task*.

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *COLING*.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97.

Stephen Clark, James Curran, and Miles Osborne. 2003. Bootstrapping pos-taggers using unlabelled data. In *HLT-NAACL*.

Pedro Domingos. 2012. A few useful things to know about machine learning. *Commun. ACM*, 55(10):78–87, October.

Andre Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. 2010. Turbo Parsers: Dependency Parsing by Approximate Variational Inference. In *EMNLP*.

David McClosky, Eugene Charniak, and Mark Johnson.

2006. Effective self-training for parsing. In *HLT-NAACL*.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *LREC*.

Hiroki Ouchi, Kevin Duh, and Yuji Matsumoto. 2014. Improving dependency parsers with supertags. In *EACL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *LREC*.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Matthieu Constant, Richárd Farkas, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2014. Overview of the SPMRL 2014 shared task on parsing morphologically rich languages. In *Notes of the SPMRL 2014 Shared Task*.