Learning Reliability of Parses for Domain Adaptation of Dependency Parsing

Daisuke Kawahara and **Kiyotaka Uchimoto**
National Institute of Information and Communications Technology,
3-5 Hikaridai Seika-cho Soraku-gun, Kyoto, 619-0289, Japan
{dk, uchimoto}@nict.go.jp

Abstract

The accuracy of parsing has exceeded 90% recently, but this is not high enough to use parsing results practically in natural language processing (NLP) applications such as paraphrase acquisition and relation extraction. We present a method for detecting reliable parses out of the outputs of a single dependency parser. This technique is also applied to domain adaptation of dependency parsing. Our goal was to improve the performance of a state-of-the-art dependency parser on the data set of the domain adaptation track of the CoNLL 2007 shared task, a formidable challenge.

1 Introduction

Dependency parsing has been utilized in a variety of natural language processing (NLP) applications, such as paraphrase acquisition, relation extraction and machine translation. For newspaper articles, the accuracy of dependency parsers exceeds 90% (for English), but it is still not sufficient for practical use in these NLP applications. Moreover, the accuracy declines significantly for out-of-domain text, such as weblogs and web pages, which have commonly been used as corpora. From this point of view, it is important to consider the following points to use a parser practically in applications:

- to select reliable parses, especially for knowledge acquisition,
- and to adapt the parser to new domains.

This paper proposes a method for selecting reliable parses from parses output by a single dependency parser. We do not use an ensemble method based on multiple parsers, but use only a single parser, because speed and efficiency are important when processing a massive volume of text. The resulting highly reliable parses would be useful to automatically construct dictionaries and knowledge bases, such as case frames (Kawahara and Kurohashi, 2006). Furthermore, we incorporate the reliable parses we obtained into the dependency parser to achieve domain adaptation.

The CoNLL 2007 shared task tackled domain adaptation of dependency parsers for the first time (Nivre et al., 2007). Sagae and Tsujii applied an ensemble method to the domain adaptation track and achieved the highest score (Sagae and Tsujii, 2007). They first parsed in-domain unlabeled sentences using two parsers trained on out-of-domain labeled data. Then, they extracted identical parses that were produced by the two parsers and added them to the original (out-of-domain) training set to train a domain-adapted model.

Dredze et al. yielded the second highest score[1] in the domain adaptation track (Dredze et al., 2007). However, their results were obtained without adaptation. They concluded that it is very difficult to substantially improve the target domain performance over that of a state-of-the-art parser. To confirm this, we parsed the test set (CHEM) of the domain adaptation track by using one of the best dependency parsers, second-order MSTParser (McDonald et al.,

[1] Dredze et al. achieved the second highest score on the CHEM test set for unlabeled dependency accuracy.

2006)[2]. Though this parser was trained on the provided out-of-domain (Penn Treebank) labeled data, surprisingly, its accuracy slightly outperformed the highest score achieved by Sagae and Tsujii (unlabeled dependency accuracy: $83.58 > 83.42$ (Sagae and Tsujii, 2007)). Our goal is to improve a state-of-the-art parser on this domain adaptation track.

Dredze et al. also indicated that unlabeled dependency parsing is not robust to domain adaptation (Dredze et al., 2007). This paper therefore focuses on unlabeled dependency parsing.

2 Related Work

We have already described the domain adaptation track of the CoNLL 2007 shared task. For the multilingual dependency parsing track, which was the other track of the shared task, Nilsson et al. achieved the best performance using an ensemble method (Hall et al., 2007). They used a method of combining several parsers' outputs in the framework of MST parsing (Sagae and Lavie, 2006). This method does not select parses, but considers all the output parses with weights to decide a final parse of a given sentence.

Reichart and Rappoport also proposed an ensemble method to select high-quality parses from the outputs of constituency parsers (Reichart and Rappoport, 2007a). They regarded parses as being of high quality if 20 different parsers agreed. They did not apply their method to domain adaptation or other applications.

Reranking methods for parsing have a relation to parse selection. They rerank the n-best parses that are output by a generative parser using a lot of lexical and syntactic features (Collins and Koo, 2005; Charniak and Johnson, 2005). There are several related methods for 1-best outputs, such as revision learning (Nakagawa et al., 2002) and transformation-based learning (Brill, 1995) for part-of-speech tagging. Attardi and Ciaramita proposed a method of tree revision learning for dependency parsing (Attardi and Ciaramita, 2007).

As for the use of unlabeled data, self-training methods have been successful in recent years. McClosky et al. improved a state-of-the-art constituency parser by 1.1% using self-training (McClosky et al., 2006a). They also applied self-training to domain adaptation of a constituency parser (McClosky et al., 2006b). Their method simply adds parsed unlabeled data without selecting it to the training set. Reichart and Rappoport applied self-training to domain adaptation using a small set of in-domain training data (Reichart and Rappoport, 2007b).

Van Noord extracted bilexical preferences from a Dutch parsed corpus of 500M words without selection (van Noord, 2007). He added some features into an HPSG (head-driven phrase structure grammar) parser to consider the bilexical preferences, and obtained an improvement of 0.5% against a baseline.

Kawahara and Kurohashi extracted reliable dependencies from automatic parses of Japanese sentences on the web to construct large-scale case frames (Kawahara and Kurohashi, 2006). Then they incorporated the constructed case frames into a probabilistic dependency parser, and outperformed their baseline parser by 0.7%.

Table 1: Labeled and unlabeled data provided for the shared task. The labeled PTB data is used for training, and the labeled BIO data is used for development. The labeled CHEM data is used for the final test.

name	source	labeled	unlabeled
PTB	Penn Treebank	18,577	1,625,606
BIO	Penn BioIE	200	369,439
CHEM	Penn BioIE	200	396,128

3 The Data Set

This paper uses the data set that was used in the CoNLL 2007 shared task (Nivre et al., 2007). Table 1 lists the data set provided for the domain adaptation track.

We pre-processed all the unlabeled sentences using a conditional random fields (CRFs)-based part-of-speech tagger. This tagger is trained on the PTB training set that consists of 18,577 sentences. The features are the same as those in (Ratnaparkhi, 1996). As an implementation of CRFs, we used CRF++[3]. If a method of domain adaptation is applied to the tagger, the accuracy of parsing unlabeled sentences will improve (Yoshida et al., 2007). This

[2] http://sourceforge.net/projects/mstparser/

[3] http://crfpp.sourceforge.net/

paper, however, does not deal with domain adaptation of a tagger but focuses on that of a parser.

4 Learning Reliability of Parses

Our approach assesses automatic parses of a single parser in order to select only reliable parses from them. We compare automatic parses and their gold-standard ones, and regard accurate parses as positive examples and the remainder as negative examples. Based on these examples, we build a binary classifier that classifies each sentence as reliable or not. To precisely detect reliable parses, we make use of several linguistic features inspired by the notion of controlled language (Mitamura et al., 1991). That is to say, the reliability of parses is judged based on the degree of sentence difficulty.

Before describing our base dependency parser and the algorithm for detecting reliable parses, we first explain the data sets used for them. We prepared the following three labeled data sets to train the base dependency parser and the reliability detector.

PTB_base_train: training set for the base parser: 14,862 sentences
PTB_rel_train: training set for reliability detector: 2,500 sentences[4]
BIO_rel_dev: development set for reliability detector: 200 sentences (= labeled BIO data)

PTB_base_train is used to train the base dependency parser, and PTB_rel_train is used to train our reliability detector. BIO_rel_dev is used for tuning the parameters of the reliability detector.

4.1 Base Dependency Parser

We used the MSTParser (McDonald et al., 2006), which achieved top results in the CoNLL 2006 (CoNLL-X) shared task, as a base dependency parser. To enable second-order features, the parameter *order* was set to 2. The other parameters were set to default. We used PTB_base_train (14,862 sentences) to train this parser.

4.2 Algorithm to Detect Reliable Parses

We built a binary classifier for detecting reliable sentences from a set of automatic parses produced by the base dependency parser.

We used support vector machines (SVMs) as a binary classifier with a third-degree polynomial kernel. We parsed PTB_rel_train (2,500 sentences) using the base parser, and evaluated each sentence with the metric of unlabeled dependency accuracy. We regarded the sentences whose accuracy is better than a threshold, τ, as positive examples, and the others as negative ones. In this experiment, we set the accuracy threshold τ at 100%. As a result, 736 out of 2,500 examples (sentences) were judged to be positive.

To evaluate the reliability of parses, we take advantage of the following features that can be related to the difficulty of sentences.

sentence length: The longer the sentence is, the poorer the parser performs (McDonald and Nivre, 2007). We determine sentence length by the number of words.

dependency lengths: Long-distance dependencies exhibit bad performance (McDonald and Nivre, 2007). We calculate the average of the dependency length of each word.

difficulty of vocabulary: It is hard for supervised parsers to learn dependencies that include low-frequency words. We count word frequencies in the training data and make a word list in descending order of frequency. For a given sentence, we calculate the average frequency rank of each word.

number of unknown words: Similarly, dependency accuracy for unknown words is notoriously poor. We count the number of unknown words in a given sentence.

number of commas: Sentences with multiple commas are difficult to parse. We count the number of commas in a given sentence.

number of conjunctions (*and/or*): Sentences with coordinate structures are also difficult to parse (Kurohashi and Nagao, 1994). We count the number of coordinate conjunctions (*and/or*) in a given sentence.

To apply these features to SVMs in practice, the numbers are binned at a certain interval for each feature. For instance, the number of conjunctions is split into four bins: 0, 1, 2 and more than 2.

[4] 1,215 labeled PTB sentences are left as another development set for the reliability detector, but they are not used in this paper.

Table 2: Example BIO sentences judged as reliable. The underlined words have incorrect modifying heads.

dep. accuracy	sentences judged as reliable
12/12 (100%)	No mutations resulting in truncation of the APC protein were found .
12/13 (92%)	Conventional imaging techniques did not show two in 10 of these patients .
6/6 (100%)	Pancreatic juice was sampled endoscopically .
11/12 (92%)	The specificity of p53 mutation for pancreatic cancer is very high .
9/10 (90%)	K-ras mutations are early genetic changes in colon cancer .

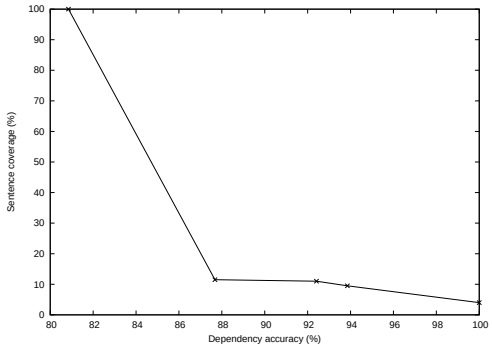

Figure 1: Accuracy-coverage curve on BIO_rel_dev.

4.3 Experiments on Detecting Reliable Parses

We conducted an experiment on detecting the reliability of parses. Our detector was applied to the automatic parses of BIO_rel_dev, and only reliable parses were selected from them. When parsing this set, the POS tags contained in the set were substituted with automatic POS tags because it is preferable to have the same environment as when applying the parser to unlabeled data.

We evaluated unlabeled dependency accuracy of the extracted parses. The accuracy-coverage curve shown in Figure 1 was obtained by changing the soft margin parameter C [5] of SVMs from 0.0001 to 10. In this figure, the coverage is the ratio of selected sentences out of all the sentences (200 sentences), and the accuracy is unlabeled dependency accuracy. A coverage of 100% indicates that the accuracy of 200 sentences without any selection was 80.85%.

If the soft margin parameter C is set to 0.001, we can obtain 19 sentences out of 200 at a dependency accuracy of 93.85% (183/195). The average sentence length was 10.3 words. Out of obtained 19 sentences, 14 sentences achieved a dependency accuracy of 100%, and thus the precision of the reliability detector itself was 73.7% (14/19). Out of 200 sentences, 36 sentences were correctly parsed by the base parser, and thus the recall is 38.9% (14/36).

Table 2 shows some sentences that were evaluated as reliable using the above setting ($C = 0.001$). Major errors were caused by prepositional phrase (PP)-attachment. To improve the accuracy of detecting reliable parses, it would be necessary to consider the number of PP-attachment ambiguities in a given sentence as a feature.

5 Domain Adaptation of Dependency Parsing

For domain adaptation, we adopt a self-training method. We combine in-domain unlabeled (automatically labeled) data with out-of-domain labeled data to make a training set. There are many possible methods for combining unlabeled and labeled data (Daumé III, 2007), but we simply concatenate unlabeled data with labeled data to see the effectiveness of the selected reliable parses. The in-domain unlabeled data to be added are selected by the reliability detector. We set the soft margin parameter at 0.001 to extract highly reliable parses. As mentioned in the previous section, the accuracy of selected parses was approximately 94%.

We parsed the unlabeled sentences of BIO and CHEM (approximately 400K sentences for each) using the base dependency parser that is trained on the entire PTB labeled data. Then, we applied the reliability detector to these parsed sentences to obtain 31,266 sentences for BIO and 31,470 sentences for CHEM. We call the two sets of obtained sentences "BIO pool" and "CHEM pool".

For each training set of the experiments described below, a certain number of sentences are randomly selected from the pool and combined with the entire out-of-domain (PTB) labeled data.

5.1 Experiment on BIO Development Data

We first conducted an experiment of domain adaptation using the BIO development set.

[5] A higher soft margin value allows more classification errors, and thus leads to the increase of recall and the decrease of precision.

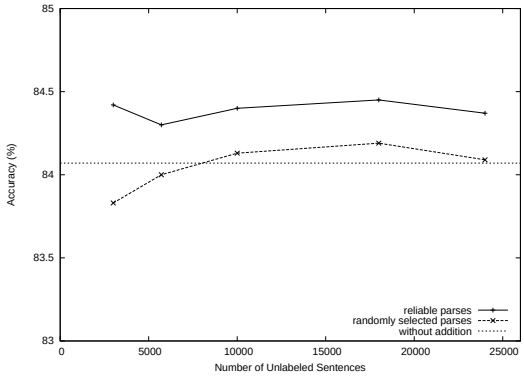

Figure 2: Dependency accuracies on BIO when the number of added unlabeled data is changed.

Figure 2 shows how the accuracy changes when the number of added reliable parses is changed. The solid line represents our proposed method, and the dotted line with points represents a baseline method. This baseline is a self-training method that simply adds unlabeled data without selection to the PTB labeled data. Each experimental result is the average of five trials done to randomly select a certain number of parses from the BIO pool. The horizontal dotted line (84.07%) represents the accuracy of the parser without adding unlabeled data (trained only on the PTB labeled data).

From this figure, we can see that the proposed method always outperforms the baseline by approximately 0.4%. The best accuracy was achieved when 18,000 unlabeled parses were added. However, if more than 18,000 sentences are added, the accuracy declines. This can be attributed to the balance of the number of labeled data and unlabeled data. Since the number of added unlabeled data is more than the number of labeled data, the entire training set might be unreliable, though the accuracy of added unlabeled data is relatively high. To address this problem, it is necessary to weigh labeled data or to change the way information from acquired unlabeled data is handled.

5.2 Experiment on CHEM Test Data

The addition of 18,000 sentences showed the highest accuracy for the BIO development data. To adapt the parser to the CHEM test set, we used 18,000 reliable unlabeled sentences from the CHEM pool with the PTB labeled sentences to train the parser. Table 3 lists the experimental results. In this table, the third row lists the three highest scores of the domain adaptation track of the CoNLL 2007 shared task.

Table 3: Experimental results on CHEM test data.

system	accuracy
PTB+unlabel (18,000 sents.)	84.12
only PTB (baseline)	83.58
1st (Sagae and Tsujii, 2007)	83.42
2nd (Dredze et al., 2007)	83.38
3rd (Attardi et al., 2007)	83.08

The baseline parser was trained only on the PTB labeled data (as described in Section 1). The proposed method (PTB+unlabel (18,000 sents.)) outperformed the baseline by approximately 0.5%, and also beat all the systems submitted to the domain adaptation track. These systems include an ensemble method (Sagae and Tsujii, 2007) and an approach of tree revision learning with a selection method of only using short training sentences (shorter than 30 words) (Attardi et al., 2007).

6 Discussion and Conclusion

This paper described a method for detecting reliable parses out of the outputs of a single dependency parser. This technique was also applied to domain adaptation of dependency parsing.

To extract reliable parses, we did not adopt an ensemble method, but used a single-parser approach because speed and efficiency are important in processing a gigantic volume of text to benefit knowledge acquisition. In this paper, we employed the MSTParser, which can process 3.9 sentences/s on a XEON 3.0GHz machine in spite of the time complexity of $O(n^3)$. If greater efficiency is required, it is possible to apply a pre-filter that removes long sentences (e.g., longer than 30 words), which are seldom selected by the reliability detector. In addition, our method does not depend on a particular parser, and can be applied to other state-of-the-art parsers, such as Malt Parser (Nivre et al., 2006), which is a feature-rich linear-time parser.

In general, it is very difficult to improve the accuracy of the best performing systems by using unlabeled data. There are only a few successful studies, such as (Ando and Zhang, 2005) for chunking and (McClosky et al., 2006a; McClosky et al., 2006b) on constituency parsing. We succeeded in boosting the accuracy of the second-order MST parser, which is

a state-of-the-art dependency parser, in the CoNLL 2007 domain adaptation task. This was a difficult challenge as many participants in the task failed to obtain any meaningful gains from unlabeled data (Dredze et al., 2007). The key factor in our success was the extraction of only reliable information from unlabeled data.

However, that improvement was not satisfactory. In order to achieve more gains, it is necessary to exploit a much larger number of unlabeled data. In this paper, we adopted a simple method to combine unlabeled data with labeled data. To use this method more effectively, we need to balance the labeled and unlabeled data very carefully. However, this method is not scalable because the training time increases significantly as the size of a training set expands. We can consider the information from more unlabeled data as features of machine learning techniques. Another approach is to formalize a probabilistic model based on unlabeled data.

References

Rie Ando and Tong Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *Proceedings of ACL2005*, pages 1–9.

Giuseppe Attardi and Massimiliano Ciaramita. 2007. Tree revision learning for dependency parsing. In *Proceedings of NAACL-HLT2007*, pages 388–395.

Giuseppe Attardi, Felice Dell'Orletta, Maria Simi, Atanas Chanev, and Massimiliano Ciaramita. 2007. Multilingual dependency parsing and domain adaptation using DeSR. In *Proceedings of EMNLP-CoNLL2007*, pages 1112–1118.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing. *Computational Linguistics*, 21(4):543–565.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL2005*, pages 173–180.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of ACL2007*, pages 256–263.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João V. Graça, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of EMNLP-CoNLL2007*, pages 1051–1055.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryigit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? a study in multilingual parser optimization. In *Proceedings of EMNLP-CoNLL2007*, pages 933–939.

Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of HLT-NAACL2006*, pages 176–183.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

David McClosky, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proceedings of HLT-NAACL2006*, pages 152–159.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of COLING-ACL2006*, pages 337–344.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL2007*, pages 122–131.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL-X*, pages 216–220.

Teruko Mitamura, Eric Nyberg, and Jaime Carbonell. 1991. An efficient interlingua translation system for multi-lingual document production. In *Proceedings of MT Summit III*, pages 55–61.

Tetsuji Nakagawa, Taku Kudo, and Yuji Matsumoto. 2002. Revision learning and its application to part-of-speech tagging. In *Proceedings of ACL2002*, pages 497–504.

Joakim Nivre, Johan Hall, Jens Nilsson, Gül sen Eryi git, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of CoNLL-X*, pages 221–225.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL2007*, pages 915–932.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP1996*, pages 133–142.

Roi Reichart and Ari Rappoport. 2007a. An ensemble method for selection of high quality parses. In *Proceedings of ACL2007*, pages 408–415.

Roi Reichart and Ari Rappoport. 2007b. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of ACL2007*, pages 616–623.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of the Companion Volume to HLT-NAACL2006*, pages 129–132.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of EMNLP-CoNLL2007*, pages 1044–1050.

Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of IWPT2007*, pages 1–10.

Kazuhiro Yoshida, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers. In *Proceedings of IJCAI-07*, pages 1783–1788.