Selecting High Quality Dependencies from Automatic Parses

Gongye Jin　Daisuke Kawahara　Sadao Kurohashi

Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, Japan
jin@nlp.ist.i.kyoto-u.ac.jp,{dk,kuro}@.i.kyoto-u.ac.jp

Abstract

In this paper, we present a method for selecting high quality dependencies from parsed sentences. By considering many aspects that a ect the accuracy of dependency parsing, we created a new set of features for supervised classification of reliable parses. Experimental results show that our approach can select dependency parses from the result analyzed by a dependency parser.

1 Introduction

Knowledge acquisition from a large corpus has been actively studied recently. Fundamental analysis techniques are applied to the corpus and knowledge is acquired from the analysis. In particular, dependency parsing has been used for some tasks like case frame compilation, relation extraction and paraphrase acquisition[1, 2, 3]. For these tasks, the accuracy of dependency parsing is vital. Although the accuracy of state-of-the-art dependency parsers for English and Japanese is over 90%, it is not high enough to acquire accurate knowledge. If one tries to apply a method of knowledge acquisition to difficult-to-analyze languages like Chinese and Arabic, the quality of the resulting knowledge will get much worse.

In this paper, we present a supervised method for selecting high quality dependencies from automatic dependency parses. This method considers language-independent linguistic features that are related to the difficulty of dependency parsing. We do not require any other annotated data than a treebank, part of which is used to train a dependency parser. We conducted experiments on English using the Penn Treebank and the experimental results show that our proposed method can select dependencies of higher quality than a baseline method.

2 Related Work

There have been several approaches to select high quality parses. One research detected parse quality by a Sample Ensemble Parse Assessment (SEPA) algorithm. in order to choose a good parser and obtain a good parsing performance[6]. Yates et al.[8] proposed a Web-based semantic filtering method, which made use of mutual information calculated from the Web to create a classifier to filter out unreliable parses. Also, an unsupervised algorithm for detecting reliable dependency parses was proposed. This method was based on the idea that, syntactic structures that are frequently created by a parser are more likely to be correct than structures produced less frequently[7].

The most related work to ours is the work of Yu et al.[9]. They proposed a framework that selects high quality parsed sentences in the first stage, and then selects high quality dependencies from the filtered sentences. In comparison with their work, we consider that even some low quality sentences possibly contain high quality pairs and take into account other aspects that can directly a ect high quality dependencies classification.

3 High Quality Dependencies Selection

In this section, we present the framework of highly reliable dependencies selection from automatic parses. Figure 1 shows the overview of our approach. We use a part of treebank to train a parser and the other part to train a binary classifier which judges a dependency to be reliable or not. We use Support Vector Machines (SVM) for the classification.

3.1 Training Data Collection

In order to train a classifier for selecting highly reliable dependencies from parsing output, we collect

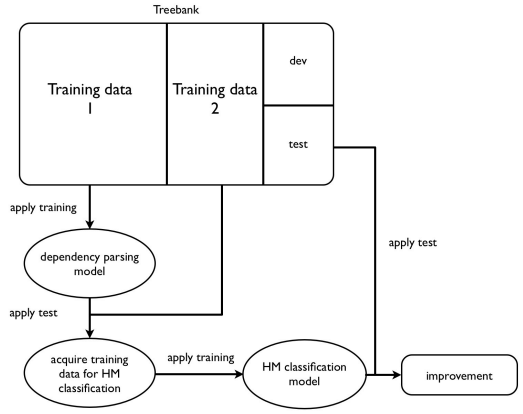

Figure 1: Overview of High Quality Dependencies selection

training data from the same corpus which is also used in dependency parsing. We first divide the traditional training data into two parts. The first part is used to train a dependency parser, the second part is used to apply dependency parsing using the model which is trained by the first part. From the parsing outputs of the second part, we acquire classification training data by collecting each dependencies. We label each of the training data by judging weather each dependency relation is correct according to the gold standard data.

3.2 Dependencies Classification

We re-judge each dependencies in parsing outputs as high quality or not and only keep correct ones. There are many factors that a ect the parsing performance. By taking these factors into consideration, we create a new set of features for classification based on the previous work (Yu et al., 2008). Table 1 lists the features of our appoach.

In the previous research (Yu et al., 2008), besides these features mentioned above, they only consider the part of speech tag of the head and modifier but without observing the context.

Most basic features onsider the fact that if there is a comma, colon or semi-colon between two arguments, they are much less likely to have a dependency relation than those pairs that does not have any punctuations between. We use these most common punctuations as features for classification. On the other hand, based on the hypothesis that a word has a higher possibility to have a dependency relation with a word argument nearby rather than a word far away, distance is another important factor for judging whether two words have a dependency relation.

In addition to these basic features, we consider other aspect such as context that a ect the parsing performance. Take two sentences “they eat salad with a fork” and “they eat salad with sauce” as examples. These examplse contain the PP-attachment ambiguity problem[4], which is one of the most difcult problem in parsing. The two prepositional phrases ‘with a fork’ and ‘with sauce’ depend on the verb ‘eat’ and the noun ‘sauce’ respectively. However, these two cases can hardly be distinguished by a dependency parser. Therefore, we want to judge them to be unreliable. Consider another similar sentence “they eat it with a fork”. Since the prepositional phrase ‘with a fork’ cannot depend on the pronoun ‘it’, this case can be clearly judged as a highly reliable pair. In order to learn such linguistic characteristics automatically, besides the part of speech tags of the head and modifier, we also use their preceding and following words and their part-of-speech tags.

Another important fact is that languages such as English and Chinese have SVO sentence structure where the subject comes first, the verb second, and the object third. The most common case is that, subject and object which locate on both side of the verb are the modifiers of the verb. This leads to the fact that arguments pairs that have a verb between can hardly have a dependency relation. By observing whether there is a verb between a head-modifier pairs can help judging whether the dependency between them is reliable. The advanced feature set we created are shown in Table 1.

4 Experiments

4.1 Experimental Setting

We employ MSTparser[1] as a base dependency parser and use section 02 to 21 from Penn Treebank to train a dependency parsing model. Then, we use section 00 to apply a dependency parsing using the trained model. From the outputs of dependency parsing, we collect training data for high quality head-modifier classification. We utilize SVM with di erent degrees which are 2 and 3 to complete the binary classification task. Section 23 is used as test set. Also, in order to compare with the previous work, we use original feature set which are the first five features in Table 1 as baseline. Moreover, we experiment on the data which is using automatic part of speech tagging by MXPOST[2] tagger.

4.2 Evaluation

According to the output of SVM, in each parsed sentence, we only select dependencies which have the output score over a threshold and discard the rest.

[1] http://www.seas.upenn.edu/ strctlrn/MSTParser/MSTParser.html
[2] http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html

Feature	Description
Basic Features	
PoS_{head},PoS_{mod}	Part of speech pair of head and modifier
$Word_{head}$,$Word_{mod}$	Word pair of head and modifier
Distance	Distance between the head and its modifier
HasComma	If there exists comma between head and modifier, set as 1; otherwise set as 0
HasColon	If there exists colon between head and modifier, set as 1; otherwise set as 0
HasSemi	If there exists semi-colon between head and modifier, set as 1; otherwise set as 0
Context Features	
HasVerb	If there exists verb between head and modifier, set as 1; otherwise set as 0
$PoS_{prehead/premod}$	Part of speech tag of the previos word of head and modifier
$PoS_{posthead/postmod}$	Part of speech tag of the post word of head and modifier
$Word_{prehead/premod}$	The previous word of head and modifier
$Word_{posthead/postmod}$	The post word of head and modifier

Table 1: Features for Dependencies Classification

The threshold is set as 1 in the experiment. We evaluate the filtered parses by calculating the percentage of correct head-modifier dependencies according to the gold standard data. Precision and recall are calculated as follows.

$$precision = \frac{\#\,of\,correct\,pairs}{\#\,of\,pairs\,acquired}$$

$$recall = \frac{\#\,of\,correct\,pairs}{\#\,of\,pairs\,in\,gold\,standard\,data}$$

4.3 Experimental Results

In our experiment, MSTparser achieves an unlabeled attachment score of 0.922, and MXPOST tagger has a tagging accuracy of 0.967. Figure 2 shows the precision-recall learning curve of the classification using SVM. 'd=2' and 'd=3' mean degree 2 and degree 3, 'baseline' and 'proposed' mean classification using basic features and our proposed feature set. Table 2 shows the evaluation results of filtered parses using

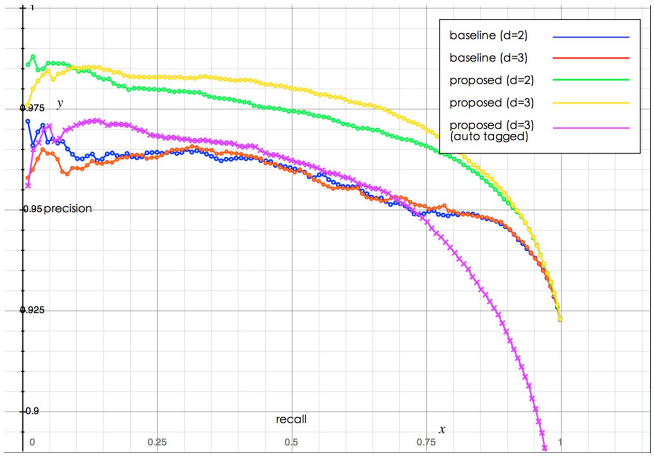

Figure 2: Precision-recall Curve of Classification

Method	Precision	Recall
baseline (d=2)	0.952	0.657
baseline (d=3)	0.953	0.651
proposed (d=2)	0.968	0.719
proposed (d=3)	**0.978**	0.594

Table 2: Result (with gold standard PoS tags)

Method	Precision	Recall
proposed (d=3)	0.958	0.573

Table 3: Result (with automatic PoS tags)

gold standard data and Table 3 shows the evaluation results by using automatic tagged data.

As we can see from the experimental results, SVM with degree 3 performs better than degree 2. Also, the method using proposed feature set reaches the highest precision. In this experiment, in order to compare di erent criteria, we set a unique threshold which is 1 for SVM output score. However, by observing the precision-recall learning curve in Figure 2, we can see that if we set threshold much higher which means decreasing the recall. For example, in the experiment, when we set the threshold as 1.1, although the recall becomes 0.277, the precision is up to 0.983. This would be considerably suitable for those tasks such as building knowledge bases from very large corpora such as the Web, where low recall would be tolerable but high precision is essentially needed.

5 Conclusion and Future Work

In this paper, we proposed a classification approach for high quality dependencies selection. We created a new set of features for classification and directly

select highly reliable dependencies from each parsed sentence through a parser. This approach can extract high quality dependencies even from some low parsing quality sentences. The experiment shows our further consideration of other aspects that a ect parsing quality and advanced feature set can improve the precision of dependencies selection.

This approach can help extract highly reliable parses from a large corpus such as the Web and subsequently assist some other tasks such as recognizing lexical preferences or predicate-argument structure construction[5] which usually highly depend on the parsing quality. We are planning to improve these subsequent tasks and also use a bootstrapping strategy to realize a improvement of dependdency parsing based on extracted high quality knowledge.

References

[1] D.Kawahara and S.Kurohashi. 2006. A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. In *Proceeding of HLT-NAACL2006*, pages 176–183.

[2] S.D.Saeger, K.Torisawa, M.Tsuchida, J.Kazama, C.Hashimoto, Ichiro Yamada, Jong-Hoon Oh, Istvan Varga, Yulan Yan. 2011. Relation Acquisition using Word Classes and Partial Patterns. In *Proceedings of EMNLP 2011*, pages 825–835.

[3] Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jun'ichi Kazama, and Sadao Kurohashi. 2011. Extracting Paraphrases from Definition Sentences on the Web. In *Proceedings of ACL2011*, pages 1087–1097

[4] D.Kawahara and S.Kurohashi. 2005. PP-attachment disambiguation boosted by a gigantic volume of unambiguous examples. In *Proceeding of the 2nd International Joint Conference on Natural Language Processing*, pages 188–198.

[5] D.Kawahara and S.Kurohashi. 2010. Acquiring Reliable Predicate-argument Structures from Raw Corpora for Case Frame Compilation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 1389–1393.

[6] R.Reichart and A.Rappoport. 2007. An Ensemble Method for Selection of High Quality Parses. In *Proceedings of the ACL 2007*, pages 408-415.

[7] F.DellOrletta, . Venturi and . Montemagn. 2011. ULISSE: an unsupervised algorithm for detecting reliable dependency parses In *Proceedings of the CoNLL 2011*, pages 115-124.

[8] A.Yates, S.Schoenmackers, and O.Etzioni. 2006. Detecting Parser Errors Using Web-based Semantic Filters. In *Proceedings of EMNLP 2006*, pages 27–34.

[9] K.Yu, D.Kawahara, and S.Kurohashi. 2008. Cascaded Classification for High Quality Head-modifier Pair Selection. In *Proceedings of NLP 2008*, pages 1–8.