F-04

Language-independent Approach to High Quality Dependency Selection From Automatic Parses

Gongye Jin†　　Daisuke Kawahara†　　Sadao Kurohashi†

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 (D.Kawahara and S.Kurohashi, 2006), relation extraction (S.D.Saeger et al., 2011) and paraphrase acquisition (C.Hashimoto et al., 2011). For these tasks, the accuracy of dependency parsing is vital. Although the accuracy of state-of-the-art dependency parsers for some languages like English and Japanese is over 90%, it is still not high enough to acquire accurate knowledge. Furthermore, 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.

Instead of using all the automatic parses, it is possible to use only high quality dependencies for knowledge acquisition. In this paper, we present a supervised language-independent approach for selecting high quality dependencies from automatic dependency parses. This method considers linguistic features that are related to the difficulty of dependency parsing. We do not require any other annotated data than a single set of dependency labeled data such as Treebank, part of which is used to train a dependency parser. We conduct experiments on English, Chinese and Japanese. The experimental results show that, for all the languages, our proposed method can select dependencies of higher quality than baseline methods.

The rest of the paper is organized as follows. Section 2 reviews some relevant research related to our approach. Section 3 describes the system of high quality dependency selection. Section 4 is a detailed description of the experiments on three different languages and discusses the evaluation results. Section 5 gives a conclusion of our approach and introduces future work.

2. Related Work

There have been a few approaches devoted to automatic selection of high quality parses or dependencies. According to selection algorithms, they can be categorized into supervised methods and unsupervised methods.

Supervised methods mainly focus on the construction of a machine learning classifier to predict the reliability of parses or dependencies based on various kinds of features both on syntactic and semantic level. Yates et al. (A.Yates et al., 2006) created WOODWARD which is a Web-based semantic filtering system. They first mapped the parses produced by a parser to a logic-based representation called *Relational conjunction* (RC). Then four different methods were employed for analyzing whether a conjunct in the RC is likely to be reasonable. Kawahara and Uchimoto (D.Kawahara and K.Uchimoto, 2008) built a binary classifier that classifies each parse of a sentence as reliable or not. The linguistic features they used for the classification, such as sentence length, number of unknown words and number of comma etc., are based on the inspiration that the reliability of parses is judged based on the degree of sentence difficulty. The work most related to ours is the work of Yu et al. (K.Yu et al., 2008). They proposed a framework that selects high quality parses in the first stage, and then selected high quality dependencies from the filtered parses. In comparison with their work, we consider that even some low quality sentences possibly contain high quality dependencies. Also, we take into account other aspects that can directly affect high quality dependency classification such as context information in order to create a new set of linguistic features for high quality dependency classification.

Among supervised methods, ensemble approaches were also proposed. Reichart and Rappoport (R.Reichart and A.Rappoport, 2007) detected parse quality by a Sample Ensemble Parse Assessment (SEPA) algorithm. They trained several different parsers by using different samples from training data. Then the level of agreement among these parsers is used to predict the quality of a parse. Another similar approach proposed by Sagae and Tsujii (K.Sagae and J.Tsujii, 2007) also selected high quality parses by computing the level of agreement on different parser outputs. But different from the former research which uses several constituency parsers trained on different sample data, they used parses produced by a different dependency parsing algorithm but the same training data. Different from those methods mentioned above, our method judges whether each dependency is reliable in the parse of each sentence outputted by a parser.

Also, unsupervised algorithms for detecting reliable dependency parses were proposed. Reichart and Rappoport (R.Reichart and A.Rappoport, 2009) proposed an unsupervised method for high quality parse selection. 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. They created PUPA (POS-based Unsupervised Parse Assessment Algorithm) to calculate the statistics about the POS tag sequences of parses produced by an unsupervised constituency parser. Dell'Orletta et al. (F.DellOrleta et al., 2011) proposed ULISSE (Unsupervised LInguiStically-driven Selection of dEpendency parses), which is also an unsupervised system. Different from the former research, they addressed the reliable parse selection task using an unsupervised method in a supervised parsing scenario. Also, instead of using constituency-related features such as ordered POS tag sequence, they used dependency-motivated features such as parse tree depth and length of dependency links. Although unsupervised methods may solve the domain adaption issue and do not use any annotated

† Graduate School of Informatics, Kyoto University

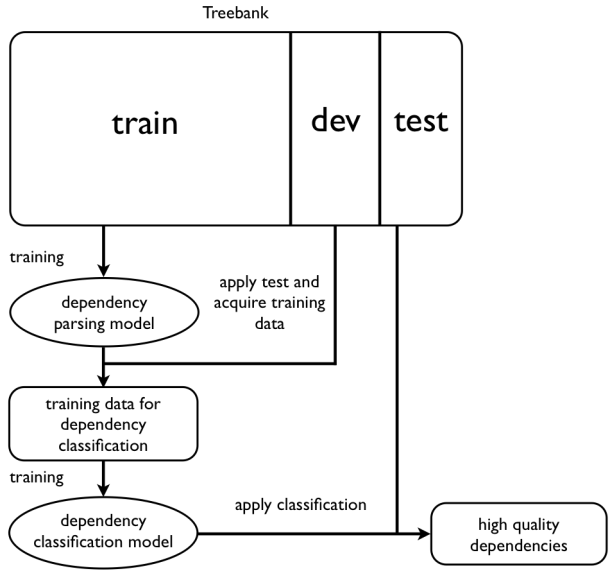

Figure 1: Overview of high quality dependency selection

data, which are always costly to acquire, the accuracy of selected parses, which is under 95%, still needs to be improved for knowledge acquisition tasks.

3. High Quality Dependency Selection

In this section, we present a framework of highly re- liable dependency selection from automatic parses. Figure 1 shows the overview of our approach. We use a part of a treebank to train a parser and another 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 for Dependency classification

Supervised methods always require manually annotated training data that is usually very costly to obtain. Owing to the limitation of existing resources, in order to train a classifier for selecting highly reliable dependencies from parsing output, we collect training data from the same corpus which is also used in dependency parsing in the first stage. First, the training section is used to train a dependency parser and the development section is used to apply dependency parsing using the model which is trained by the training section. From the output parses of the development section, we acquire training data for dependency classification by collecting each dependency. We label each element of the training data by judging weather each dependency relation in output parses is correct according to the gold standard data. All the correct dependencies are defined as reliable and vice versa.

3.2 Dependency Classification

We judge each dependency in parsing outputs as high quality or not and only keep high quality ones. There are many factors that affect the parsing performance such as distance between dependencies. By taking these factors into consideration, we create sets of features for classification. Table 1 and Table 2 list the features of our approach.

Feature	Description
POShead, POSmod	Part of speech pair of head and modifier
Wordhead, Wordmod	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

Table 1: Basic features for dependency classification

3.2.1 Basic Features

Most basic features consider the fact that if there is a comma, colon or semi-colon between two words, they are much less likely to have a dependency relation than those pairs that do not have any punctuation between them. Those dependencies that contain punctuation between them are always much more difficult for a parser to analyze correctly than those do not contain any punctuation. In other words, whether a dependency contains punctuation reflects the difficulty of parsing and the output's reliability. We use the most common punctuation as features for classification. On the other hand, based on the fact that a word has a higher possibility to have a dependency relation with a word argument nearby rather than a word far away, dependencies with longer distance always show worse parsing performance (R.McDonald and J.Nivre, 2007). Thus distance is another important factor that reflects the difficulty of judging whether two words have a dependency relation. Yu et al. (2008) used the features mentioned above except the word features (*Wordhead* and *Wordmod*) and did not use the context features, which are described in the next section.

3.2.2 Context Features

In addition to these basic features, we consider context features that are thought to affect the parsing performance. Table 2 lists these context features. Take the two sentences "they eat salad with a fork" and "they eat salad with sauce" as examples. These examples have the PP-attachment ambiguity problem, which is one of the most difficult problems 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 these kinds of structure 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' but only on the verbal phrase 'eat', this case can be clearly judged as a highly reliable dependency pair. In some more complex cases, it is also necessary to observe larger span of context. In order to learn such linguistic characteristics automatically, besides POS tags the head and modifier in a dependency, we also use their preceding and following one and two words along with their POS tags.

Feature	Description
HasVerb	If there exists verb between head and modifier, set as 1; otherwise set as 0
POSprehead/premod	Part of speech tag of the preceding one and two words of head and modifier
POSposthead/postmod	Part of speech tag of the following one and two word of head and modifier
Wordprehead/premod	The preceding one and two words of head and modifier
Wordposthead/postmod	The following one and two words of head and modifier

Table 2: Context features for dependency classification

Another important fact is that verbal phrases in the dependency tree structure of a parse are normally the root node of the whole dependency tree or the parent node of a subtree. When a word pair that contains a verbal phrase between them, the two words are always on different sides of a parent node. Thus, these kinds of word pairs will always have no dependency link between them. For example, in SVO languages such as English and Chinese, the subject comes first, the verb second and the object third. The most common case is that subjects and objects located on both sides of the verb are the modifiers of the verb. This leads to the fact that argument pairs that have a verb between them rarely have a dependency relation. Observing whether there are verbal phrases between a head-modifier pairs can help judge whether the dependency between them is reliable.

3.2.3 Tree-based Features

The input of our high quality dependency selection method is a dependency tree. It is very natural to use tree-based features to identify the quality of dependencies. Based on a head-modifier dependency pair, we observe modifier's modifiers, a.k.a children nodes. We use the leftmost and rightmost of children nodes. We also take head's parent node into consideration, which we call a modifier's grandparent node. Furthermore, nodes that we call a modifier's uncle nodes are also considered as other features. Similarly, we use leftmost and rightmost uncle nodes.

4. Experiments

4.1 Experimental Settings

We experiment on English, Chinese and Japanese. For English, we employ MSTparser[1] as a base dependency parser and use sections 02 to 21 from Wall Street Journal (WSJ) corpus in Penn Treebank (PTB) to train a dependency parsing model. Then, we use section 22 from WSJ to apply the dependency parsing model to acquire the training data for dependency classification. MXPOST[2] tagger is used for English automatic POS tagging. For Chinese, we use CNP[3] parser to train a dependency parser using section 1 to 270, 400 to 931 and 1001 to 1151 from Penn Chinese Treebank (CTB). Sections 301 to 325 are used to apply dependency parsing to acquire training data for dependency classification. We use MMA (C.Kruengkrai et al., 2009) to apply both segmentation and POS tagging. Different from the previous two languages which take *words* as the basic unit, experiments are based on the unit of the phrase segments *bunsetsu.* We first use JUMAN[4] for Japanese morphological analysis. Then KNP[5] is utilized for Japanese dependency parsing. Section 950101 and 950103 from Kyoto Corpus are used to apply dependency parsing and acquire training data for dependency selection.

From the outputs of dependency parser, we collect training data for high quality dependency classification. All the correct dependencies according to the gold standard data are defined as positive examples and vice versa. We utilize SVM to complete the binary classification task. We employ SVM-Light[6] with polynomial kernel (degree 3) to solve the binary classification. In order to compare with previous work by Yu et.al (2008), we use the basic feature set as a baseline. For English, section 23 from WSJ is used as a test set. Section 271 to 300 from CTB, section 950104 and 950105 are used to test the classification approach in Chinese and Japanese respectively.

4.2 Evaluation Metrics

According to the output of the SVM, we only select dependencies that have the output score over a threshold and discard the rest. The higher the output score is, the more reliable the dependency is judged as. As a result, high threshold means low recall. Then we evaluate the filtered dependencies by calculating the percentage of correct head-modifier dependencies according to the gold standard data. Precision is calculated as ratio of correct dependencies in retrieved ones, recall is the ratio of correct dependencies in total. In Chinese and Japanese automatic tagged and parsed data, due to the performance of segmentation, there are many segments that are incorrectly produced. In these cases, we treat them as incorrect examples. Note that the maximum recall value equals the precision of base dependency parser without dependency selection.

4.3 Experimental Results

Figure 2 shows the precision-recall curves of the classification using SVM for English, Japanese and Chinese. In these graphs, 'basic' means the method using the basic features. 'context' stands for considering context information. 'context+tree' means using additional tree-based features. We achieved dependency precisions of 99%, 96%and 98% for English, Chinese and Japanese automatic tagged data if we adopt a recall of 20%. These results are quite promising for subsequential NLP tasks such as knowledge acquisition. Our proposed context features show a significant advantage over the original feature set proposed in the previous work. By taking context information into account, we can effectively help the system learn the reliability of dependencies in automatic dependency parses.

[1] http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html
[2] http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html
[3] http://alaginrc.nict.go.jp/cnp/
[4] http://nlp.ist.i.kyoto- u.ac.jp/EN/ index.php?JUMAN
[5] http://nlp.ist.i.kyoto- u.ac.jp/EN/ index.php?KNP
[6] http://svmlight.joachims.org

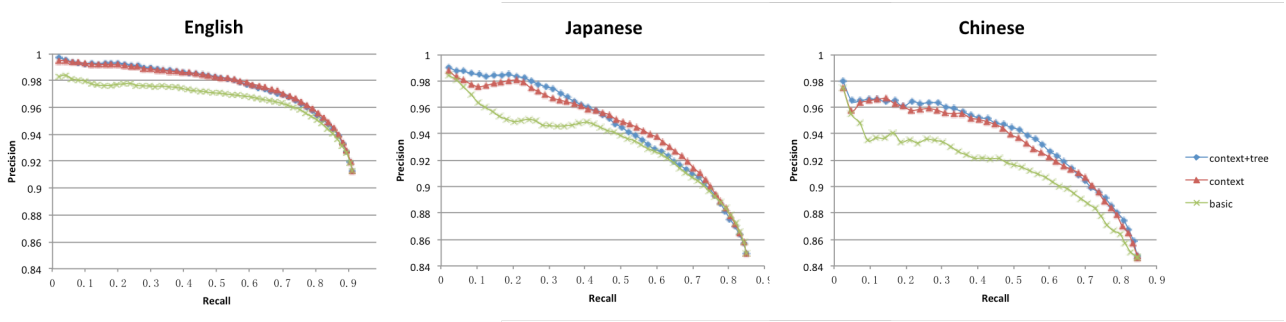

Figure 2: Precision-recall curve of dependency classification for English (left), Japanese (middle) and Chinese (right)

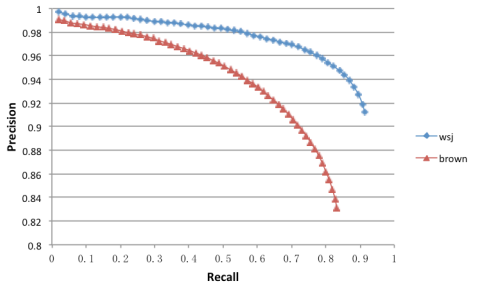

Figure 4: Comparison with Brown corpus

One of the biggest problems that most data-driven parsers are facing is the domain adaption problem. When they are applied to a text of a different domain, their accuracy decreases significantly due to the lack of domain-specific training data. We applied the dependency parsing model trained on WSJ to the Brown corpus, and obtained an unlabeled attachment score of 0.832, which is significantly lower than the in-domain score by 8.1%. We applied the same dependency selection model trained on WSJ to the Brown corpus. Figure 4 shows the precision-recall curves of dependency selection on the Brown corpus. From the results, we can see that when the recall is 40% for example, high quality dependencies with a precision of over 95% can be acquired. This shows that our method works well on data from different domains. This fact creates a good way to acquire knowledge from a large raw corpus in different domains (e.g., the Web).

5. Conclusion and Future Work

In this paper, we proposed a classification approach for high quality dependency selection. We created a set of features with the consideration of context information to select highly reliable dependencies from each parsed sentence through a parser. This approach can extract high quality dependencies even from some low parse quality sentences. The experiments showed that our method worked for in-domain parses and also out-of-domain parses.

We can extract high quality dependencies from a large corpus such as the Web and subsequently assist knowledge acquisition tasks, such as subcategorization frame acquisition and case frame compilation, which depends highly on the parse quality. We also plan to use a bootstrapping strategy to realize an improvement of a dependency parser based on acquired high quality knowledge from large corpora.

References

A.Yates, S.Schoenmackers, and O.Etzioni. 2006. Detecting parser errors using web-based semantic filters. In *Proceedings of EMNLP 2006,* pages 27–34.

C.Hashimoto, K.Torisawa, S.D.Saeger, J.Kazama, and S.Kurohashi. 2011. Extracting paraphrases from definition sentences on the web. In *Proceedings of ACL 2011,* pages 1087-1097.

C.Kruengkrai, K.Uchimoto, J.Kazama, Y.Wang, K.Torisawa, and H.Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and pos tagging. In *Proceedings of ACL-IJCNLP 2009,* pages 513–521.

D.Kawahara and K.Uchimoto. 2008. Learning reliability of parses for domain adaptation. In *Proceedings of IJCNLP 2008,* pages 709–714.

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

F.DellOrleta, G.Venturi, and S.Montema. 2011. Ulisse: an unsupervised algorithm for detecting reliable de- pendency parses. In *Proceeding of CoNLL 2011,* pages 115–124.

K.Sagae and J.Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensemble. In *Proceedings of EMNLP-CoNLL 2007,* pages 408–415.

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.

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

R.Reichart and A.Rappoport. 2007. An ensemble method for selection of high quality parses. In *Proceedings of ACL 2007,* pages 408–415.

R.Reichart and A.Rappoport. 2009. Automatic selection of high quality parses created by a fully unsupervised parser. In *Proceedings of CoNLL 2009,* pages 156– 164.

S.D.Saeger, K.Torisawa, M.Tsuchida, J.Kazama, C.Hashimoto, I.Yamada, J.Oh, I.Varga, and Y.Yan. 2011. Relation acquisition using word classes and partial patterns. In *Proceedings of EMNLP 2011,* pages 825–835.