Cascaded Classification for High Quality Head-modifier Pair Selection

Kun Yu
Graduate School of Informatics, Kyoto University
Kyoto, 606-8501, Japan
kunyu@nlp.kuee.kyoto-u.ac.jp

Daisuke Kawahara
National Institute of Information and Communications Technology
Kyoto, 619-0289, Japan
dk@nict.go.jp

Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Kyoto, 606-8501, Japan
kuro@i.kyoto-u.ac.jp

Abstract

This paper presents a cascaded classification approach for selecting head-modifier pairs with high quality from syntactically analyzed sentences. Experimental results show that the proposed approach achieved 76.11% on F-score of selected head-modifier pairs, which was 8.54% higher than the baseline approach that using sentence length as selection criteria. In addition, compared with using the head-modifier pairs extracted by the baseline approach, the parsing accuracy of a lexicalized dependency parser can be further improved by using the head-modifier pairs selected by the proposed approach.

1 Introduction

Head-modifier pairs acquired from large corpus by syntactic analysis can recognize lexical preference and improve lexicalized parsing consequently (Yu et al., 2007). While, due to the non-perfect accuracy of corpus syntactic analyzer, there always exist head-modifier pairs with low quality in analysis outputs. Applying these low quality head-modifier pairs in parsing will affect parsing accuracy. Good parse selection, which has been studied by many researchers recently (Reichart and Rappoport, 2007; Yates et al., 2006), extracts sentences with high parsing accuracy automatically. Using good parse selection is a feasible way to acquire high-quality head-modifier pairs. But when the parsing accuracy is not 100%, there still exist some pairs with low quality even if in the well-parsed sentences.

In this paper, we present a cascaded classification approach for not only choosing parsed sentences with high accuracy from syntactic analysis outputs, but also selecting head-modifier pairs with high quality from the chosen sentences. The proposed approach consists of two steps: (1) *high quality sentence selection by classification from syntactic analysis outputs*; (2) *high quality head-modifier pair selection through classification from the result of (1)*. We did experiments on Penn Chinese Treebank 5.1 (Xue et al., 2002). Results show that the proposed approach successfully increased the F-score of selected head-modifier pairs compared with only using sentence length as selection criteria. In addition, we apply the selected head-modifier pairs into a lexicalized dependency parser to check their effectiveness. Experimental results show that compared with simply using the head-modifier pairs extracted from short sentences, the head-modifier pairs selected by the proposed approach helped improve parsing accuracy further.

2 Cascaded Classification for High Quality Head-modifier Pair Selection

2.1 High Quality Sentence Selection

The first step of the proposed approach uses a classifier (*Sentence classifier*) based on SVM classification to select the parsed sentences with high accuracy from the syntactic analysis outputs. TinySVM[1] is selected as the SVM toolkit. A polynomial kernel is applied and degree is set as 2.

Table 1 shows the features used in *Sentence classifier*. To get training data for this classifier, we use a syntactic analyzer to parse the training corpus. The sentences whose parsing accuracy is higher than λ_{sent} are used as positive examples and the left sentences in the training corpus are used as negative examples. To guarantee the quality of training data, λ_{sent} is set as 0.95 empirically.

When selecting high quality sentences, we use SVM classification score as the selection criteria. Only when the SVM score $score_{sent}(sent_i)$ of sentence $sent_i$ is higher than θ_{sent}, we select sentence $sent_i$ as high quality sentence. Obviously, higher θ_{sent} means better precision but

[1] http://chasen.org/~taku/software/TinySVM/

worse recall for selected sentences. In the proposed approach, θ_{sent} is set as 0.

Table 1. Features for *Sentence classifier*.

Feature	Description
Pos_{head}/Pos_{mod}	Pos-tag pair of all the head-modifier pairs in this sentence
AvgDis	Average distance of all the head-modifier pairs in this sentence. It is calculated by $\sum_i dis_i / len$, where dis_i means the distance between the i_{th} head-modifier pair and *len* means the length of this sentence.
Length	Sentence length.
#Comma	The number of commas in this sentence.
#Colon	The number of colons in this sentence.
#Semi	The number of semi-colons in this sentence.

2.2 High Quality Head-modifier Pair Selection

In this step, we apply another classifier (*HM classifier*) to select high quality head-modifier pairs from the sentences selected in step 1.

This classifier is also based on SVM classification. Table 2 lists all the features.

Table 2. Features for *HM classifier*.

Feature	Description
Pos_{head}/Pos_{mod}	Pos-tag pair of head and modifier
Distance	Distance between head and 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

The training corpus of this classifier is also analyzed by the same syntactic analyzer used in the first step. The correct head-modifier pairs created by the analyzer are used as positive examples and the incorrect pairs are used as negative examples. We still use TinySVM as the toolkit with polynomial kernel and set degree as 2.

When using this classifier to select high quality head-modifier pairs, we also use SVM classification score as criteria. If the SVM score $score_{hm}(hm_j)$ of head-modifier pair hm_j is higher than θ_{hm}, we select hm_j as high quality head-modifier pair. Likewise, higher θ_{hm} means higher precision but lower recall of selected head-modifier pairs. In the proposed approach, θ_{hm} is set as 0.

3 Integrating Selected Head-modifier Pairs into Parsing

To show the effectiveness of the head-modifier pairs selected by the proposed approach, we integrate them into a lexicalized Chinese dependency parser.

This parser gives a probability $P(T|S)$ to each possible dependency tree T of an input sentence $S=w_1,w_2,\ldots,w_n$, and outputs the dependency tree T^* that maximizes $P(T|S)$ (see equation 1). CKY algorithm is applied to decode the dependency tree from bottom to up.

$$T^* = \arg\max_T P(T \mid S) \tag{1}$$

$P(T|S)$ is then divided as equation 2.

$$\begin{aligned} &P(T \mid S) \\ &= P(w_{ROOT} \mid S) \times \prod_{i=1}^{n} P(w_i, dis_i, comma_i \mid w_{ROOT}) \\ &= P(w_{ROOT} \mid S) \times \prod_{i=1}^{n} (P(w_i \mid w_{ROOT}) \times P(dis_i, comma_i \mid w_{ROOT}, w_i)) \end{aligned} \tag{2}$$

$P(w_{ROOT}|S)$ is estimated using the training corpus by equation 3. $C(w_{ROOT}=ROOT)$ is the number of dependency trees in the corpus where w_{ROOT} is root node. $C(ROOT)$ is the total number of dependency trees in the corpus.

$$\hat{P}(w_{ROOT} \mid S) = \frac{C(w_{ROOT} = ROOT)}{C(ROOT)} \tag{3}$$

$P(w_j|w_{ROOT})$ and $P(dis_j, comma_j|w_{ROOT},w_j)$ are estimated by both the training corpus and the head-modifier pairs selected by the proposed approach (see equation 4, 5, 6).

$$\hat{P}(w_j \mid w_{ROOT}) = \frac{\hat{F}(w_j \mid w_{ROOT})}{\sum_k \hat{F}(w_k \mid w_{ROOT})} \tag{4}$$

$$\hat{F}(w_j \mid w_{ROOT}) = \frac{C(w_{ROOT} \to w_j)}{C(w_{ROOT}, w_j)} \tag{5}$$

$$\begin{aligned} &\hat{P}(dis_j, comma_j \mid w_{ROOT}, w_j) \\ &= \frac{C(w_{ROOT} \to w_j, dis = dis_j, comma = comma_j)}{C(w_{ROOT}, w_j)} \end{aligned} \tag{6}$$

Here $C(w_{ROOT} \rightarrow w_j)$ is the number of times that w_{ROOT} generates w_j in the corpus, and $C(w_{ROOT},w_j)$ is the total co-occurrence of w_{ROOT} and w_j in the corpus. $C(w_{ROOT} \rightarrow w_j,\ dis=dis_j,\ comma=comma_j)$ means the number of times that w_{ROOT} generates w_j in the corpus when the distance between w_{ROOT} and w_j is dis_j and there exists $comma_j$'s comma between w_{ROOT} and w_j. A normalization factor is applied in equation 4 to ensure $\sum_j P(w_j \mid w_{ROOT}) = 1$.

To solve the data sparseness problem, the same smoothing strategy as what was used in (Collins, 1996) is applied in this parser.

4 Results and Discussion

4.1 Experimental Setting

Two experiments are conducted to prove the feasibility of the proposed approach.

- ***Head-modifier pair evaluation***

The aim of this experiment is to validate the effectiveness of the proposed approach on high quality head-modifier pair selection.

We tested two approaches in this experiment, which are:

(1) *Baseline*

As one of the simplest ways for good parse selection, this approach looks all the head-modifier pairs existing in the sentences with no more than *k* (*k*=30) words as high quality head-modifier pairs. It is based on the assumption that parsing short sentences is easier than parsing long sentences.

(2) *Proposed*

It is the proposed approach that applies both *Sentence classifie*r and *HM classifier* in sequence to select high quality head-modifier pairs.

In this experiment, 6,204 sentences from Section 400-931 of Penn Chinese Treebank 5.1 are used to train *Sentence Classifier*. 3,480 sentences from Section 001-270 are used to train *HM Classifier*. 346 sentences from Section 271-300 are used as testing data. Both training data and testing data are analyzed by a dependency parser (Yu et al., 2007) trained on Penn Chinese Treebank. Gold word segmentation and pos-tagging are applied in syntactic analysis for both data sets.

Three evaluation metrics are used here, which are *precision* (see equation 7), *recall* (see equation 8) and *F-score*= $\frac{2 \times precision \times recall}{precision + recall}$.

$$\text{precision} = \frac{\#\text{ of correct HM pairs in output}}{\#\text{ of selected HM pairs}} \qquad (7)$$

$$\text{recall} = \frac{\#\text{ of correct HM pairs in output}}{\#\text{ of correct HM pairs in gold standard}} \qquad (8)$$

- ***Parsing evaluation***

In this experiment, we hope to prove that compared with using the head-modifier pairs acquired by the baseline approach (see *Head-modifier pair evaluation*), using the head-modifier pairs selected by the proposed approach can give more help to improve parsing accuracy.

9,684 sentences from Section 001-270 and 400-931 of Penn Chinese Treebank 5.1 are used to train the parser. The toolkit Penn2Malt[2] is applied to transfer the phrase structure of Penn Chinese Treebank to dependency structure. 346 sentences with gold word segmentation and pos-tagging from Section 271-300 are used to test parsing. All the head-modifier pairs integrated into the parser are extracted from Chinese Gigaword (Graff et al., 2005), which is analyzed by the same syntactic analyzer (Yu et al., 2007) used in previous test. Unlabeled attachment score (*UAS*) (Buchholz and Marsi, 2006) is used to evaluate parsing accuracy.

4.2 Results of Head-modifier Pair Evaluation

Table 3 lists the evaluation results of selected head-modifier pairs. It shows that the baseline approach achieved good precision but suffered from low recall. Compared with the baseline approach, the proposed approach increased both precision and recall by using cascaded classification, and finally improved F-score greatly.

Table 3. Results of selected head-modifier pairs.

Approach	*Precision*(%)	*Recall*(%)	*F-score*(%)
Baseline	93.55	52.88	67.57
Proposed	96.88(+3.33)	62.67(+9.79)	76.11(+8.54)

4.3 Results of Parsing Evaluation

Table 4 lists the parsing result with head-modifier pairs selected by the different approaches. '*N/A*' means not integrating head-modifier pairs into parser.

Results show that even if we only used the baseline approach for head-modifier pair selection, integrating head-modifier pairs helped increase parsing accuracy. It proves the effectiveness of applying head-modifier pairs into lexicalized parsing. In addition, our proposed approach can acquire head-modifier pairs with higher F-score. Thus compared with using head-modifier pairs acquired by the baseline approach, using head-modifier pairs selected by our proposed approach could give more improvement on parsing accuracy.

[2] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

Table 4. Parsing results with different head-modifier pair selection approaches.

Head-modifier pair Selection approach	*UAS* (%)
N/A	85.13
Baseline	86.11(+0.98)
Proposed	86.43(+1.30)

4.4 Discussion

In the proposed approach, we use parameters (θ_{sent}=0, θ_{hm}=0) to adjust the ratio between precision and recall of head-modifier pair selection. Table 5 shows the results of the proposed approach with different parameters. These results prove the feasibility of our current setting. Besides, it shows that setting higher θ_{hm} improved precision of selected head-modifier pairs and affected the recall at the same time, but increasing θ_{sent} gave no help to either precision or recall of selected head-modifier pairs. While, only four different parameter settings are tested here. Using a development set to get learning curve and set the appropriate parameters is more reasonable. We will do this work in the future.

Table 5. Results of selected head-modifier pairs with different parameters.

Parameter		*Precision(%)*	*Recall(%)*	*F-score(%)*
θ_{sent}	θ_{hm}			
0	0	96.88	62.67	76.11
0	0.5	98.77	56.01	71.48
0.5	0	96.87	56.78	67.27
0.5	0.5	98.77	51.00	71.60

In addition, the proposed approach is trained on the syntactic analysis of sentences with gold word segmentation and pos-tagging currently. But when using the proposed approach to select head-modifier pairs for the lexicalized parser, the raw corpus is segmented and pos-tagged by a morphological analyzer. This difference may affect the quality of selected head-modifier pairs. We will train the proposed approach on the sentences with real word segmentation and pos-tagging in the future to resolve this issue.

5 Related Work

Most of the related works focused on good parse selection, which is similar to the *Sentence classifier* proposed in our approach. For example, Reichart and Rappoport (2007) presented a sample ensemble parse assessment algorithm, which used the level of agreement among several copies of a parser, to predict the quality of a parse. Yates et al. (2006) proposed an algorithm which filtered out high quality parses by performing semantic analysis. Compared with them, our proposed approach uses a SVM-based classifier (*Sentence classifier*) to get sentences with high parsing quality. In addition, the proposed approach applies another classifier (*HM classifier*) on the acquired sentences to selected high quality head-modifier pairs further.

6 Conclusion and Future Work

High quality head-modifier pairs can help improve accuracy for lexicalized parsing. But previous work on good parse selection cannot guarantee the quality of all the head-modifier pairs existing in selected sentences. This paper proposes a cascaded classification approach to (1) select high quality parsed sentences by a *Sentence classifier*; (2) select high quality head-modifier pairs by a *HM classifier* from the sentences acquired in the first step. Experimental results prove that compared with simply selecting head-modifier pairs by sentence length, the proposed approach could improve F-score of selected head-modifier pairs greatly. In addition, the selected high quality head-modifier pairs could give more help to lexicalized parsing than the head-modifier pairs selected by sentence length.

There are still some works that we should consider to complete the proposed approach, which includes finding proper parameter setting by learning curve, training the proposed approach on sentences with real word segmentation and pos-tagging, and comparing the proposed approach with other good parse selection methods.

References

S.Buchholz and E.Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL-X*.

M.Collins. 1996. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of ACL 1996*.

D.Graff et al.. 2005. Chinese Gigaword Second Edition. *Linguistic Data Consortium*, Philadelphia.

R.Reichart and A.Rappoport. 2007. An Ensemble Method for Selection of High Quality Parses. In *Proceedings of the ACL 2007*.

N.Xue, F.Chiou and M.Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of COLING 2002*.

A.Yates, S.Schoenmackers, and O.Etzioni. 2006. Detecting Parser Errors Using Web-based Semantic Filters. In *Proceedings of EMNLP 2006*.

K.Yu et al.. 2007. Learning Head-modifier Pairs to Improve Lexicalized Dependency Parsing on a Chinese Treebank. In *Proceedings of Treebanks and Linguistic Theories 2007*.