NJU-Parser: Achievements on Semantic Dependency Parsing

Guangchao Tang[1] Bin Li[1,2] Shuaishuai Xu[1] Xinyu Dai[1] Jiajun Chen[1]
[1] State Key Lab for Novel Software Technology, Nanjing University
[2] Research Center of Language and Informatics, Nanjing Normal University
Nanjing, Jiangsu, China
{tanggc, lib, xuss, dxy, chenjj}@nlp.nju.edu.cn

Abstract

In this paper, we introduce our work on SemEval-2012 task 5: Chinese Semantic Dependency Parsing. Our system is based on MSTParser and two effective methods are proposed: splitting sentence by punctuations and extracting last character of word as lemma. The experiments show that, with a combination of the two proposed methods, our system can improve LAS about one percent and finally get the second prize out of nine participating systems. We also try to handle the multi-level labels, but with no improvement.

1 Introduction

Task 5 of SemEval-2012 tries to find approaches to improve Chinese sematic dependency parsing (SDP). SDP is a kind of dependency parsing. Currently, there are many dependency parsers available, such as Eisner's probabilistic dependency parser (Eisner, 1996), McDonald's MSTParser (McDonald et al. 2005a; McDonald et al. 2005b) and Nivre's MaltParser (Nivre, 2006).

Despite of elaborate models, lots of problems still exist in dependency parsing. For example, sentence length has been proved to show great impact on the parsing performance. (Li et al., 2010) used a two-stage approach based on sentence fragment for high-order graph-based dependency parsing. Lacking of linguistic knowledge is also blamed.

Three methods are promoted in this paper trying to improve the performance: splitting sentence by commas and semicolons, extracting last character of word as lemma and handling multi-level labels. Improvements could be achieved through the first two methods while not for the third.

2 Overview of Our System

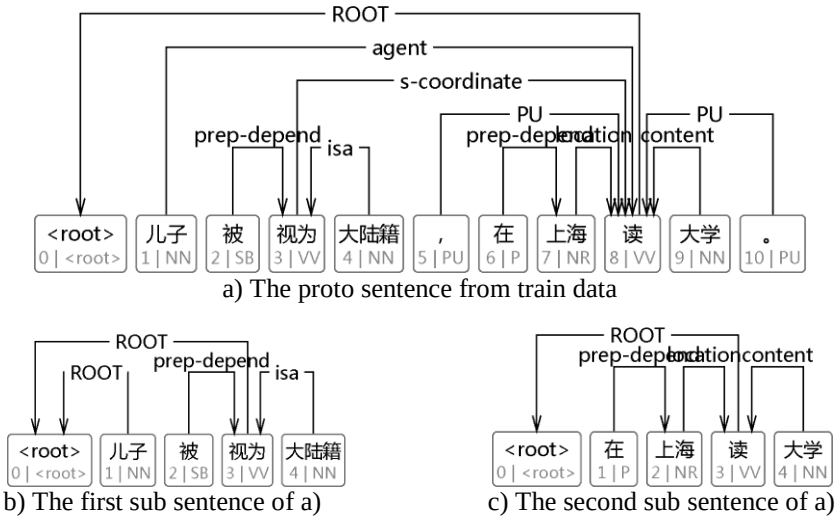
Our system is based on MSTParser which is one of the state-of-the-art parsers. MSTParser tries to obtain the maximum spanning tree of a sentence. For projective parsing task, it takes Eisner's algorithm (Eisner, 1996) to get the dependency tree in $O(n^3)$ time. Meanwhile, Chu-Liu-Edmond's algorithm (Chu and Liu, 1965) is applied for non-projective task, which takes $O(n^2)$ time.

Three methods are adopted to MSTParser in our system:

1) Sentences are split into sub-sentences by commas and semicolons, for which there are two ways. Splitting sentences by all commas and semicolons is used in our primary system. In our contrast system, we use a classifier to determine whether a comma or semicolon can be used to split the sentence. In the primary and contrast system, the proto sentences and the sub-sentences are trained and tested separately and the outputs are merged in the end.
2) In a Chinese word, the last character usually contains main sense or semantic class. We treat the last character of the word as word lemma and find it gets a slightly improvement in the experiment.
3) An experiment trying to solve the problem of multi-level labels was conducted by parsing different levels separately and consequently merging the outputs together.

The experiment results have shown that the first two methods could enhance the system performance while further improvements could be obtained through a combination of them in our sub-submitted systems.

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 519–523,
Montréal, Canada, June 7-8, 2012.



a) The proto sentence from train data

b) The first sub sentence of a)

c) The second sub sentence of a)

Figure 1. An example of the split procedure.

3 Experiments

3.1 Split sentences by commas and semicolons

It is observed that the performance decreases as the length of the sentences increases. Table 1 shows the statistical analysis on the data including SemEval-2012, Conll-07's Chinese corpus and a subset extracted from CTB using Penn2Malt. Long sentence can be split into sub-sentences to get better parsing result.

Items	SemEval -2012	Conll-07 CN	CTB
Postages count	35	13	33
Dependency labels count	122	69	12
Average sentence length	30.15	5.92	25.89
Average dependency length	4.80	1.71	4.36
LAS	61.37	82.89	67.35
UAS	80.18	87.64	79.90

Table 1. Statistical analysis on the data. The CTB data is a subset extracted from CTB using Penn2Malt.

Our work can be described as following steps:

Step 1: Use MSTParser to parse the data. We name the result as ***"normal output"***.

Step 2: Split train and test data by all commas and semicolons. The delimiters are removed in the sub sentences. For train data, a word's dependency relation is kept if the word's head is under the cover of the sub sentence. Otherwise, its head will be set to root and its label will be set to *ROOT* (*ROOT* is the default label of dependency arcs whose head is root). We define the word as ***"sentence head"*** if its head is root. ***"Sub-sentence head"*** indicates the *sentence head* of a sub-sentence. After splitting, there may be more than one *sub-sentence heads* in a sub-sentence. Figure 1 shows an example of the split procedure.

Step 3: Use MSTParser to parse the data generated in step 2. We name the parsing result ***"split output"***. In *split output*, there may be more than one sub-sentences corresponding to a single sentence in *normal output*.

Step 4: Merge the *split output* and the *normal output*. The outputs of sub-sentences are merged with delimiters restored. Dependency relations are recovered for all punctuations and *sub-sentence heads* in *split output* with relations in *normal output*. The sentence head of *normal output* is kept in final output. The result is called ***"merged split output"***. This step need to be consummated because it may result in a dependency tree not well formed with several *sentence heads* or even circles.

The results of experiments on develop data and test data are showed in table 2. For develop data, an improvement of 0.85 could be obtained while 0.93 for test data, both on LAS.

In step 2, there is an alternative to split the sentences, i.e., using a classifier to determine which comma and semicolon can be split. This method is taken in the contrast system. When applying the classifier, all commas and semicolons in train data

are labeled with S-IN or S-STOP while other words with NULL. If the sub sentence before the comma or semicolon has only one *sub-sentence head*, it is labeled with S-STOP, otherwise with S-IN. A model is built from train data with CRF++ and test data is evaluated with it. Features used are listed in table 3. Only commas and semicolons with label S-STOP can be used to split the sentence in step 2. Other steps are the same as above. The result is also shown in table 2 as ***"merged split output with CRF++"***.

Data	Methods	LAS	UAS
Develop data	normal output	61.37	80.18
	merged split output	62.22	80.56
	merged split output with CRF++	61.97	80.73
	lemma output	61.64	80.47
	primary system output	**62.41**	80.96
	contrast system output	62.05	80.90
Test data	normal output	60.63	79.37
	merged split output	61.56	80.17
	merged split output with CRF++	61.42	80.20
	lemma output	60.88	79.42
	primary system output	61.63	80.35
	contrast system output	**61.64**	80.29

Table 2. Results of the experiments.

w-4,w-3,w-2,w-1,w,w+1,w+2,w+3,w+4
p-4,p-3,p-2,p-1,p,p+1,p+2,p+3,p+4
wp-4,wp-3,wp-2,wp-1,wp wp+1,wp+2,wp+3,wp+4
w-4|w-3,w-3|w-2,w-2|w-1,w-1|w, w|w+1,w+1|w+2,w+2|w+3,w+3|w+4
p-4|p-3,p-3|p-2,p-2|p-1,p-1|p, p|p+1,p+1|p+2,p+2|p+3,p+3|p+4
first word of sub-sentence before the delimiter

Table 3. Features used in CRF++. w represents for word and p for PosTag. +1 means the index after current while -1 means before.

3.2 Extract last character of word as lemma

In Chinese, the last character of a word usually contains main sense or semantic class, which indicates that it may represent the whole word. For example, "国"(country) can represent "中国"(China) and "恋"(love) can represent "热恋"(crazy love).

The last character is used as lemma in the experiment, with an improvement of 0.27 for LAS on develop data and 0.24 on test data. Details of the scores are listed in table 2 as ***"lemma output"***.

3.3 Multi-level labels experiment

A notable characteristic of SemEval-2012's data is multi-level labels. It introduces four kinds of multi-level labels which are s-X, d-X, j-X and r-X. The first level represents the basic semantic relation of the dependency while the second level shows the second import, except that s-X represents sub-sentence relation.

The r-X label means that a verb modifies a noun and the relation between them is reverse. For example, in phrase "贫户(poor) 出身(born) 的 明星(star)", "出身" is headed to "明星" with label r-agent. It means that "明星" is the agent of "出身".

When a verbal noun is the head word and its child has indirect relation to it, the dependency is labeled with j-X. In phrase "学校(school) 建设(construction)", "建设" is the head of "学校" with label j-content. "学校" is the content of "建设".

The d-X label means that the child modifies the head with an additional relation. For example, in phrase "科技(technology) 企业(enterprise)", "科技" modifies "企业" and the domain of "企业" is "科技".

A heuristic method is tried in the experiment. The multi-level labels of d-X, j-X and r-X are separated into two parts for each level. For example, "d-content" will be separated to "d" and "content". For each part, MSTParser is used to train and test. We call the outputs ***"first-level output"*** and ***"second-level output"***. The outputs of each level and *normal output* are merged then.

In our experiments, only the word satisfies the following conditions need to be merged:

a) The dependency label in *normal output* is started with d-, j- or r-.
b) The dependency label in *first-level output* is d, j or r.
c) The heads in *first-level output* and *second-level output* are of the same.

Otherwise, the dependency relation in *normal output* will be kept. There are also three ways in merging outputs:

a) Label in *first-level output* and label in *second-level output* are merged.
b) First level label in *normal output* and label in *second-level output* are merged.
c) Label in *first-level output* and second level label in *normal output* are merged.

Experiment has been done on develop data. In the experiment, 24% of the labels are merged and 92% of the new merged labels are the same as original. The results of three ways are listed in table 4. All of them get decline compared to *normal output*.

outputs	LAS	UAS
normal output	61.37	80.18
way a)	61.18	80.18
way b)	61.25	80.18
way c)	61.25	80.18

Table 4. Results of multi-level labels experiment on develop data.

3.4 Combined experiment on split and lemma

Improvements are achieved by first two methods in the experiment while a further enhancement is made with a combination of them in the submitted systems. The split method and lemma method are combined as primary system. The split method with CRF++ and lemma method are combined as contrast system. When combining the two methods, last character of the word is firstly extracted as lemma for train data and test data. Then the split or split with CRF++ method is used.

The outputs of the primary system and contrast system are listed in table 2.

4 Analysis and Discussion

The contrast system presented in this paper finally got the second prize among nine systems. The primary system gets the third. There is an improvement of about one percent for both primary and contrast system. The following conclusions can be made from the experiments:

1) Parsing is more effective and accurate on short sentences. A word prefers to depend on another near to it. A sentence can be split to several sub sentences by commas and semicolons to get better parsing output. Result may be improved with a classifier to determine whether a comma or semicolon can be used to split the sentence.
2) Last character of word is a useful feature. In the experiment, the last character is coarsely used as lemma and a minor improvement is achieved. Much more language knowledge can be used in parsing.
3) The label set of the data is worthy to be reviewed. The meanings of the labels are not given in the task. Some of them are confusing especially the multi-level labels. The trying of training and testing multi-level labels separately by levels fails with a slightly decline of the score. Multi-level also causes too many labels: any single-level label can be prefixed to form a new multi-level label. It's a great problem for current parsers. Whether the label set is suitable to Chinese semantic dependency parsing should be discussed.

5 Conclusion and Future Work

Three methods applied in NJU-Parser are described in this paper: splitting sentences by commas and semicolons, taking last character of word as lemma and handling multi-level labels. The first two get improvements in the experiments. Our primary system is a combination of the first two methods. The contrast system is the same as primary system except that it has a classifier implemented in CRF++ to determine whether a comma or a semicolon should be used to split the sentence. Both of the systems get improvements for about one percent on LAS.

In the future, a better classifier should be developed to split the sentence. New method should be applied in merging split outputs to get a well formed dependency tree. And we hope there will be a better label set which are more capable of describing semantic dependency relations for Chinese.

Acknowledgments

This paper is supported in part by National Natural Science Fund of China under contract 61170181, Natural Science Fund of Jiangsu under contract BK2011192, and National Social Science Fund of China under contract 10CYY021.

References

Y.J. Chu and T.H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

MSTParser: http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html

J. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. COLING*.

J. Nivre. 2006. *Inductive Dependency Parsing*. Springer.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online Large-Margin Training of Dependency Parsers. *43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005).*

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective Dependency Parsing using Spanning Tree Algorithms. *Proceedings of HLT/EMNLP 2005*.

Zhenghua Li, Wanxiang Che, Ting Liu. 2010. Improving Dependency Parsing Using Punctuation. *International Conference on Asian Language Processing(IALP)* 2010.