Automatic Identification of Chinese Event Descriptive Clause

Liou Chen
Department of Computer Science and technology
Tsinghua University
chouou@foxmail.com

Qiang Zhou
National Laboratory for Information Science and Technology,
Tsinghua University
zq-lxd@mail.tsinghua.edu.cn

Abstract

This paper gives a new definition of Chinese clause called "Event Descriptive Clause" and proposes an automatic method to identify these clauses in Chinese sentence. By analyzing the characteristics of the clause, the recognition task is formulated as a classification of Chinese punctuations. The maximum entropy classifier is trained and two kinds of useful features and their combinations are explored in the task. Meanwhile, a simple rule-based post processing phase is also proposed to improve the recognition performance. Ultimately, we obtain 81.32% F-score on the test set.

1 Introduction

An important task in natural language processing (NLP) is to identify the complete structure of a sentence. However, the ambiguities of the natural language make full parsing difficult to become a practical and effective tool for NLP applications. In order to solve this problem, "partial parsing" is proposed to divide complex sentences into simple units, and then the complex full-parsing task can be simplified to be the analysis of single units and relations among them. Ejerhed(1998) once found that a parser can benefit from automatically identified clause boundaries in discourse, and he showed the partial parsing method called "clause identification" is useful for full parsing.

For example, given a Chinese sentence as follows:

- 沿途，我们见到因为更新伐倒的树木，因为建筑伐倒的树木，都是有用之材；运送树木的货车、拖拉机，南来北往。
- Along the way, we see the trees have been cut down for regeneration, and the trees needed to be cut for building. All of them are useful building material. We also see several freight trucks and tractors going south and north.

The illustrative sentence is a long one that is difficult to parse with a one-step full parsing and will suffer from the error propagation from the previous wrong parsing results.

However, if the sentence is segmented into several independent clauses which can be parsed separately, the shortening of sentence length will make each sub-parsing much easier and the independent of each clause can also prevent the error-propagation. For example, the above sentence can be divided into four parts which are labeled with dashed borders shown in Figure 1. Each segment can be parsed solely as a sub tree and the whole parse tree can be easily built through analyzing the event relationships among them. Moreover, the parse errors occurring in each sub tree have little effect on the whole tree as they are parsed independently in each segment region.

The key issue is how to select a suitable segmentation unit. It is not a trivial question because it must be based on the characteristics of language itself. In English, a clause is a closely related group of words that include both a subject and a verb. The independent sentence is usually ended by punctuation and the dependent one is often introduced by either a subordinating conjunction or a relative pronoun. The structural

trait of English language is the basic to define English clause and clause recognition task, like CoNLL-2001 (Erik F et al., 2001).

However in Chinese, there is no obvious conjunction between two clauses, especially the dependent clauses. The separators used often are just punctuations, like commas and periods. Therefore the characteristics of Chinese sentence call for a new clause identification scheme to spit a sentence into clause segments.

To meet this need, we define a new clause unit called "Event Descriptive Clause (EDC)" in the Chinese sentence. It mainly considers the punctuation separators so as to skip the difficulty in identifying different subordination clauses without any obvious separating tags.

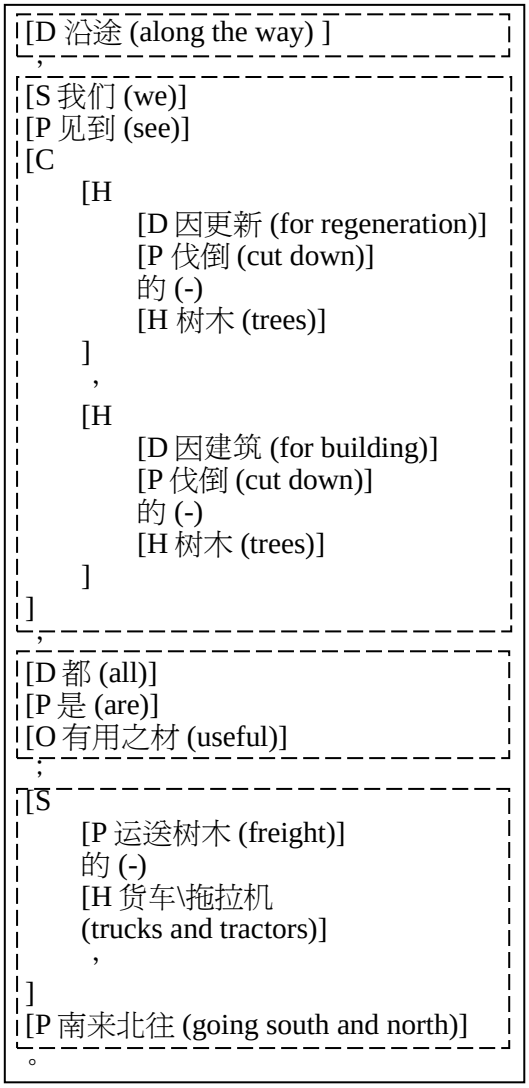

Figure 1. Parsing result of the example sentence.

According to the definition, we proposed an EDC recognition method based on punctuation classification. Experimental results show that the new definition of Chinese clause identification task is reasonable and our feature set is effective to build a feasible EDC recognition system.

2 EDC Recognition Task

2.1 Definition of Chinese Clause

As we discussed before, '*clause identification*' is a useful step in language processing as it can divide a long complex sentence into several short meaningful and independent segments. Therefore the definition of a clause should satisfy two basic requirements: 'meaningful' and 'independent'. The previous restriction requires each clause to make sense and express a full meaning, and the latter one insures that each clause can be parsed alone.

We firstly give the definition of '*Event*'. An event is expressed by several functional chunks (Zhou and Li, 2009) which are controlled by a certain predicate. The functional chunks are defined as the subject, predicate, object and adverbial parts of a clause. According to different event level, the complex components of a high level event may contain some low level events.

Let us take the second part of Figure 1 as an example. The high level event dominated by the verbal predicate '见到/see' is : "[S 我们/ We] [P 见到/ see] [C 因为更新伐倒的树木，因为建筑伐倒的树木/ the trees have been cut down for regeneration, and the trees needed to be cut for building]". The event is composed of three high level functional chunks.

The complement of above event also contains two nested events controlled by the predicate '伐倒/cut down'. Which are '[D 因为更新(for regeneration)] [P 伐倒(cut down)] 的 [H 树木(trees)]' and '[D 因为建筑(for building)] [P 伐倒(cut down)]的[H 树木(trees)]'. The chunks in these two events are low level ones.

Next, we consider the characteristics of Chinese sentences. Because the punctuations, like commas, semicolons, question marks, etc. are commonly-used obvious independent event separators. We can use them to segment a word sequence as a possible clause in a sentence.

Then based on the overall consideration of the definition of 'Event' and the characteristics of Chinese sentence, we define the Event Descriptive Clause (EDC) as a word sequence separated by punctuations, the sequence should contain either a simple high level event or a complex main event with its nested low level events.

Taking some special conditions into consideration, the adverbials to describe common time or space situations of several events, and the independent components to describe sentence-level parenthesis, can also be regarded as special EDCs though sometimes they do not contain any predicates.

In the Chinese language, many events can share subject and object with the adjacent events so that the subject or object can be omitted. We differentiated them with different tags in our EDC definition schemes.

In summary, three types of EDCs are considered as follows:

(1) E1: an EDC that includes at least one subject in the event it contains.

(2) E2: an EDC that has no subject.

(3) D/T: an EDC acted as sentence-level adverbial or independent composition.

Then the above example sentence can be divided into following four EDCs:

- [D 沿途]，[E1 我们见到因更新伐倒的树木，因建筑伐倒的树木]，[E2 都是有用之材]；[E1 运送树木的货车、拖拉机，南来北往] 。
- [D Along the way], [E1 we see the trees have been cut down for regeneration, and the trees needed to be cut for building]. [E2 All of them is useful building material]. [E1 We also see several freight trucks and tractors going south and north].

2.2 Task Analyses

According to the EDC definition, we define the Chinese clause identification as a task that recognizing all types of EDCs in an input sentence after word segmentation and POS tagging. Like the example in section 2.1, each EDC is recognized and enclosed between brackets. The task consists of two subtasks. One is to recognize suitable EDC boundaries in a sentence. The other is to assign suitable tags for each recognized EDCs. We only focus on the first subtask in the paper. Comparing with CoNLL-2010 task, our task only recognizes the EDCs that contain the highest level events without identifying its internal nested structures.

Since EDC is defined as a word sequence separated by certain punctuations. The identification problem can be regarded as a classification task to classify the punctuations as one of two classes: boundary of an EDC (Free Symbol), or not an EDC boundary (Non-Free Symbol). Then the words sequence between two Free Symbols is an EDC.

By analysis, we found only several types of punctuations could be used as EDC separator commonly, including period, question mark, exclamatory mark, ellipsis, comma, semicolon , colon and brackets. The previous four types of punctuations always appear at the end of a sentence so we simply name them as 'End Symbol'. The following four types are called 'Non-End Symbol' accordingly. The Free-Symbols are recognized from these special punctuations.

3 EDC Recognition System

3.1 Recognition Process

Statistical data from the EDC-annotated corpus provided by CIPS-ParsEval-2009 task (Zhou and Li, 2009) show that 99.87% End Symbols act as the boundaries of EDCs. So we can simply assume them as Free Symbol. But for Non-End Symbols, the linguistic phenomena are complex. If we present a baseline system that regards every Non-End Symbol as a Free Symbol rough, only 61% symbols can be correctly recognized and the remaining 39% are wrongly treated.

To solve this problem, we implement a classifier for Non-End Symbol specially. First of all, we propose several features that might be useful to determine whether a Non-End Symbol is free or not. Then, the performance of each feature is tested on a maximum entropy classifier to find the most effective features and form the final feature set. We will discuss them detailed in the following sections.

3.2 Features

Features are very important in implementing a classifier. We consider two types of features:

As EDC is a word sequence, the word and part of speech (POS) features are the most intui-

tional information for clause boundary recognition. We call the word level features 'basic features' as Table 1 shows.

However, the structural characteristics of a sentence cannot be completely reflected by words it contains. As the events in an EDC are expressed by functional chunks as section 2.1 presents, the functional chunk (FC) might be effective in recognition. They can provide more syntactic structure features than the word sequences. We consider four types of FC-related features as in Table 2.

Those two major types of features are tested and the final feature set will be selected through experiments

Feature	
Current POS	
$Word_n/POS_n$	
Adjacent Non-End Symbols	distance
	current word
	adjacent word
Left verb	
Left preposition	
Adjacent brackets	distance
	adjacent POS

Table 1. Basic Features

Feature	Description
Location	if current punctuation is in a functional chunk, the feature is 1, else is 0
$Chunk_n$	functional tags in different positions of local context windows
Chunk sequence	functional tags between current punctuation and first left Non-End Symbol
Predicate number	the number of predicates between current punctuation and first left Non-End Symbol

Table 2. Extended Features

3.3 Feature Selection Strategy

The features listed in Table 1 and Table 2 are considered to be useful but whether there are actually effective are unknown. Therefore we should select the most useful ones through experiments using certain strategy.

In the paper, we try a greedy strategy. Firstly, each feature is used alone to get its '*contribution*' to the classification system. Then after all features are tested, they are sorted by their contributions. At last, features are added one by one into classifier according to their contribution ranks and then pick out the features that can improve the performance and take out those features that have no effect on performance or even lead to the degradation. Eventually, we get a proper feature set.

As shown in Table 1 and Table 2, $Word_n/POS_n$ and $Chunk_n$ tags are used and their positions (n) are important. In this paper, we let the position window change from [0, 0] to [-5, 5] to select the proper position area.

4 Experimental results

All data we use in this paper are provided by CIPS-ParsEval-2009 task (Zhou and Li, 2009). They are automatically extracted from Tsinghua Chinese Treebank/TCT (Zhou et al., 1997), including 14,248 Chinese sentences as training material and 3,751 sentences as test data. We used the sentences annotated with Gold-standard word segmentation and POS tags as the input data for EDC recognition.

4.1 Feature Selection

We use the 14,248 training sentences to judge the contribution of each feature and get final feature set. The training corpus is divided into two parts with the ratio of 80% and 20%. 80% data is used to train classifiers and the remaining 20% for feature selection.

The maximum entropy toolbox1 is chosen for classification due to its training efficiency and better performance. A functional chunk parser (Yu, 2007) trained on the same CIPS-ParsEval-2009 FC bank (Zhou and Li, 2009) are used to provide extended features. Its F-score is 85%. The parser could only provide the lowest level functional chunks. For example, given the input sentence "运送树木的货车、拖拉机，南来北往/ the freight trucks and tractors going south

[1]http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

and north", the output functional chunk sequence are : '[P 运送树木 (freight)] 的 [H 货车、拖拉机 (trucks and tractors)]，[P 南来北往 (going south and north)]'.

The evaluation measure is defined as follows:

$$\text{Accuracy} = \frac{\text{Correctly classified Symbols}}{\text{Total Non-End Symbols}} \quad \text{(a)}$$

The performance of each feature is evaluated and ranked as Table 3 shows.

When selecting the proper position area of Chunk_n and $\text{Words}_n/\text{POS}_n$, the areas change from [0, 0] to [-5, 5] and the performance curves are shown in Figure 2 and Figure 3.

Then the feature in Table 3 is added one by one into classifier and the feature will be moved when it causes performance degradation. Table 4 presents the accuracy changes on 20% development data set.

Form above experimental figures and tables we can get several conclusions:

Figure 2 and Figure 3 display the performance changes under different window sizes (from [0, 0] to [-5, 5]). Then the abscissas of their highest points are chosen as best window sizes. We can find that when the window size is large enough, the performance change will be inconspicuous, which means the information far away from current punctuation has less help in judging whether it is free or not.

Table 3 gives the contribution of each single feature in identifying Non-End Symbols. Comparing with the baseline system proposed in section 3.1, each feature could achieve obvious increase. Therefore our attempt that building a classifier to identify Free Symbols from Non-End Symbols is feasible.

The results in Table 4 show that with features added into classifier the performance raises except for the fifth one (*Left preposition*). Therefore our final feature set will include nine features without the '*Left preposition*'.

At the same time, the top four features are all extended ones and they can achieve 81.83% accuracy while the basic features could only increase the performance less than 1% (0.95% g). This phenomenon indicates that the syntactic information can reflect the structural characteristics of Chinese clauses much better. Therefore we hypothesize that we can use extended features only to build the classifier.

Feature	Accuracy
Chunk_n (n∈[-4, 4])	80.07
Chunk sequence	76.51
Predicate number	75.40
Location	69.57
Left preposition	69.40
$\text{Words}_n/\text{POS}_n$ (n∈[-4, 3])	68.77
Left verb	68.77
Current POS	66.81
Adjacent Non-End Symbols	66.33
Adjacent brackets	66.19

Table 3. Accuracy and rank of each feature

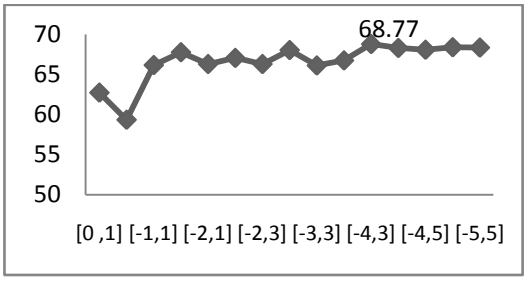

Figure 2. Performance of $\text{Words}_n/\text{POS}_n$

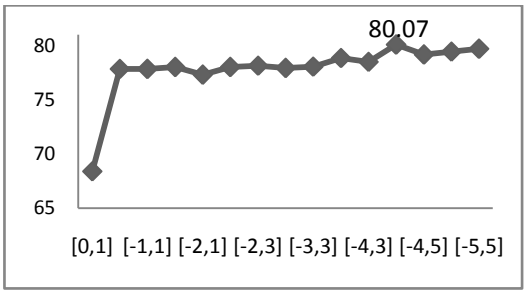

Figure 3. Performance of Chunk_n feature under different context windows

Feature	Accuracy
Chunk_n (n∈ [-4, 4])	80.07
(+)Chunk sequence	80.43
(+)Predicates number	80.87
(+)Location	81.83
(+)Left preposition	**81.67**
(+)$\text{Words}_n/\text{POS}_n$ (n∈ [-4, 3])	81.93
(+)Left verb	82.04
(+)Current POS	82.12
(+)Adjacent Non-End Symbols	82.43
(+)Adjacent bracket	82.78

Table 4. Accuracy with adding features on development data set.

4.2 Evaluating System Performance

With the feature set selected in section 4.1, the EDC identification system can be built. The total 14,248 sentences are included to train the classifier for classifying the Non-End Symbol and all test material is used for evaluating the performance of clause recognition.

We consider different modes to evaluate the clause recognition system. One is only using the extended features provided by automatic syntactic parser to validate our guess that the syntactic features are so effective that they will achieve satisfying result without other accessional features (mode_1). The second mode is adding basic word features along with syntactic ones to get the best performance that our current system can obtain (mode_2). Since the chunk features used in this classifier are from the automatic analyses. To clear the influence caused by automatic parsing, we use the lowest level correct chunks to provide syntactic features in the third method. The entirely correct chunks are provided by CIPS-ParsEval-2009 FC bank (Zhou and Li, 2009). As EDC is defined as the description of a high level event, we guess that the highest level chunks might provide more effective information. For example, for the same input sentence "运送树木的货车、拖拉机，南来北往/ the freight trucks and tractors going south and north", its high level chunk sequence will be '[S 运送树木的货车、拖拉机 (freight trucks and tractors)]，[P 南来北往 (going south and north)]'.Then model_4 will use the golden-standard high level chunk features extracted from relevant TCT (Zhou et al., 1997) to clear the upper bound of system performance.

The evaluation measure is defined as follows, and we only use the F-score.

$$\text{Recall} = \frac{\text{Correctly recognized clauses}}{\text{Total correct clauses}} \quad \text{(b)}$$

$$\text{Precision} = \frac{\text{Correctly recognized clauses}}{\text{Total recognized clauses}} \quad \text{(c)}$$

$$\text{F} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{(d)}$$

Recognition performances of the four modes are shown in Table 5.

In order to deal with some special conditions that our classifier cannot treat well to improve the performance of whole system, a simple rule-based post processing phase is designed which aims at rectifying wrong recognized sentence-level adverbial and independent composition, that is:

When there are only two EDCs are recognized in a sentence and one of which is an adverbial or independent composition, we simply assume that these two EDCs should be merged into a single big EDC.

To estimate the benefit of post-processing, we compare the performances before/after adding post-processing. The contrasts are shown in Table 6.

	mode1	mode2	mode3	mode4
Classifier Accuracy	79.64	80.60	83.46	93.34
System F-score	77.71	78.77	81.29	89.57
Model Size	181 KB	2.2 MB	/	/
Training Time	3.7s	12.6s	/	/

Table 5. Performances on four models

	mode1	mode2	mode3	mode4
F-score (Before)	77.71	78.77	81.29	89.57
F-score (After)	79.43 ⇧ 1.72	81.32 ⇧2.55	84.04 ⇧2.75	90.65 ⇧1.08

Table 6. The Performance changes caused by post-processing

The first line of Table 5 is the accuracy of Non-End Symbol classifier and the second one shows the F-score of whole EDC recognition system. From the two lines we can get this conclusion that the performance of whole system will increase along with the advancement of classifier. We also find that the system performance under automatic lowest level chunk feature does not drop too much comparing with the one under gold-standard chunks (less than 3%), which means existing syntactic parser is good enough to provide the low level chunk features. However, the recognition F-score will increase to nearly 91% when standard high level chunk features are used, which proves that the relationship between high level functional chunks and our defined EDCs are much closer that they are more efficient in recognition. Therefore we can try to build a good high level chunk parser in future. Results of mode_1 and mode_2 show that comparing with the classifier that uses all features, using only syntactic features can save

nearly three times of training time and occupy only 1/10 storage space without losing too much reorganization performance. It tells us that when time and storage space is limited we can just use syntactic features.

Table 6 presents the impact of our post-processing. We can find that the processing is effective though it is simple. This result also reflects that current classifier has difficulties to distinguish whether an adverbial or independent composition is at sentence-level or clause-level.

5 Discussions

5.1 EDC Error Types

Because different EDC recognition errors (too long or too short) might cause different problems, we define three error types according to the boundary differences between the recognized EDCs and the gold-standard ones.

(1) '1: N' error: The boundary of a recognized EDC is wider than the gold-standard one.

(2) 'N: 1' error: The boundary of a gold-standard EDC is wider than the recognized one.

(3) 'N: M' error: Several recognized EDCs and the gold-standard ones are crossed on their boundaries.

We do some statistical analysis on all 1584 wrongly recognized EDCs and Table 7 displays the distributional ratios of each error type.

Error type	1:N	N:1	N:M
Ratio (%)	59.2	38.9	1.9

Table 7. Distribution of different EDC recognition errors

5.2 Error Analysis

(1) 1:N Error

When this error happens, it will have no terrible effect on the final whole parse tree if the relations between this wrong recognized EDC and other EDCs remain the same. Like the example sentence in Figure 1, if the second and the third EDCs are wrong recognized as a single one, it will become a little troublesome to parse this EDC as its length is longer than it should be but the tree it builds with other two EDCs will not change. However, if the wrong EDC causes relationship changes, the parse errors might happen on the complete tree. In our system 1: N errors are mainly the following three types:

I. Several sentence-level adverbials are combined.

II. Adjacent EDCs are recognized as a subject or object that they are regarded as a single EDC.

III. Several adverbials at different levels are merged to be one adverbial incorrectly.

For the following sentence:

- [D 四十六亿年来]，[D 在地球表面形成过程中]，[E1 在陆地上，气候呈规律性变化] [E2 在中纬度表现最明显]，[E1 生物由海洋发展到陆地]。
- [D For 4.6 billion years], [D in the process of the formation of the earth's surface], [E1 the climate change regularly on land], [E2 the phenomenon presents clearly in the mid-latitude regions], [E1 organisms develop from ocean to land].

If the first two adverbials are recognized as a single one, error I happens. Then error II occurs when E1 and E2 are merged into one EDC. If the adverbial "在陆地上/on land" of E1 is wrongly recognized as sentence-level and is merged to its adjacent adverbial "在地球表面形成过程中/in the process of the formation of the earth's surface", the third error appears.

The previous two error conditions may not affect the final parser tree and could be regarded as 'tolerable' error. The third situation will change the relationships within EDCs that might affect following parser.

(2) N:1 Error

N: 1 error mainly includes three sub-types.

I. Complex coordinate structure/adverbial clause/attributive clause is wrong separated.

II. Complex subject/object clause is divided.

Conditions II is the reflections of sub-type II in 1: N error. Therefore it is 'tolerable' error. The first errors are caused by complex sentence-like component, like in Figure 1, when the comma in the second EDC is classified as End-Symbol, the error occurs. To solve this problem, one proper method is to consider some features of the relationship between two adjacent possible EDCs. Another way is trying to implement high level chunk parser that can provide sentence-level features instead of current bottom functional chunks.

(3) N:M Error

The proportion of this error is less than 2% that we will not pay much attention to it now.

6 Related works

There have already been some systems for clause identification. Abney (1990) used a clause filter in his CASS parser. The filter could recognize basic clauses and repair difficult cases. Leffa (1998) implemented an algorithm for finding clauses in English and Portuguese texts. He wrote a set of clause identification rules and applied them to a small corpus and achieved a good performance with recall rates above 90%. Orasan (1990) used a hybrid method for clause splitting in the Susanne corpus and obtained F-score of about 85% for this particular task. In the CoNLL-2001 shared task (Erik F et al., 2001), six systems had participated to identify English clauses. They used various machine learning techniques and connectionist methods. On all three parts of the shared task, the boosted decision tree system of Carreras and Marquez (2001) performed best. It obtained an F-score of 78.63.

However, as English and Chinese clauses have different characteristics, the researches on English sometimes ignore punctuation, especially the comma, or they just use a comma as one feature to detect the segmentation without fully using the information of punctuations.

In Chinese, Jin (2004) gave an analysis for the complete usages of the comma. Li (2005) tried to use punctuations to divide long sentence into suitable units to reduce the time consumption in parsing long Chinese sentences. Their processing based on simply rules. Yu (2007) proved that using clause recognition to divide a sentence into independent parts and parse them separately could achieve extremely significant increase on dependency accuracy compared with the deterministic parser which parsed a sentence in sequence. The CIPS-ParsEval-2009 (Zhou and Li, 2009) put forward a task to identify the Chinese EDC and six systems participated. Based on the idea of "HNC" (1998), Wei (2009) used a semantic knowledge corpus to identify EDCs and achieved the performance of F-score 80.84 (open track). Zhou (2009) formulated the task as a sequence labeling problem and applied the structured SVMs model. Their performance was 78.15. Wang (2009) also regarded the task as a sequence labeling problem and considered the CRFs to resolve this problem and got an F-score of 69.08. Chen and Zhou (2009) presented a classification method that identified the boundaries of EDCs using maximum entropy classifier, and the system obtained an F-score of 79.98.

Based on our previous work, some new features are introduced and the performance of each feature is evaluated, our identification system achieved an F-score of 81.32. At the same time, the comparison between two different chunk levels show that high level chunk features are much more powerful that we can devote ourselves to building a good high level parser in future to increase the performance farther.

7 Conclusions

In this paper, we compare the different characteristics between Chinese language and English, and define a new Chinese clause called "Event Descriptive Clause (EDC)". Then on the basis of this definition, we propose an effective method for Chinese EDC identification.

Our work focus on the commas which are usually useful in Chinese clause recognition but always ignored by researchers, and tries different types of features through experiments to clear their different effects in identifying EDC boundaries from commas. At the same time, our statistical model is combined with useful rules to deal with the recognition task better. Finally our automatic EDC recognition system achieved 81.32 of F-score, which is higher than other systems based on the same data set.

Meanwhile, error analyses show that the current identification system has some problems. Therefore we propose several possible methods, expecting to solve these problems and improve the recognition ability of EDC recognition system in future.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 60573185, 60873173), National High Technology Research and Development Projects 863 (No. 2007AA01Z173) and Tsinghua-Intel Joint Research Project.

References

Abney Steven, “Rapid Incremental Parsing with Repair”. In "Proceedings of the 8th New OED Conference: Electronic Text Research", University of Waterloo, Ontario, 1990.

Carreras, X. and Marquez, L. “Boosting Trees for Clause Splitting”. In “Proceedings of CoNLL-2001”, Toulouse, France, pp 73-75, 2001.

Chen Liou, Zhou Qiang. “Recognition of Event Descriptive Clause”. In “Proceedings of the 1st CIPS-ParsEval”, Tsinghua University, Beijing, pp.65-72. 2009.

Ejerhed Eva I., “Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods,” In “Proceedings of ANLP ’88”, pp.219-227, 1998.

Erik F. Tjong Lim Sang and Dejean H. “Introduction to the CoNLL-2001 Shared Task: Clause Identification [A]”. In Proc. of CoNLL-2001 [C], Toulouse, France, p53-57, 2001.

Huang Zengyang. “Theory of Hierarchical Network of Concepts”. Tsinghua University Press, Beijing, 1998.

Jin Meixun, Mi-Yong Kim, Dongil Kim and Jong-Hyeok Lee. “Segmentation of Chinese Long Sentences Using Commas”. Proc. SIGHAN, Barcelona, Spain, pp. 1-8, 2004.

Leffa, Vilson J. “Clause processing in complex sentences, In “Proceedings of LREC'98”, Granada, Espanha, 1998.

Li Xing and Chengqing Zong. “A Hierarchical Parsing Approach with Punctuation Processing for Long Complex Chinese Sentences.” In Companion Volume to the Proceedings of Conference including Posters/Demos and Tutorial Abstracts, IJCNLP2005, Jeju Island, Korea, pp.9-14, 2005.

Orasan Constantin. “A hybrid method for clause splitting in unrestricted English texts”. In “Proceedings of ACIDCA'2000”, Monastir, Tunisia, 2000.

Wei Xiangfeng, “Labeling Functional Chunk and Event Sentence Based on the Analysis of Sentence Category”. In “Proceedings of the 1st CIPS-ParsEval”, Tsinghua University, Beijing, pp.57-64, 2009.

Wang Xi, Wang Jinyong, Liu Chunyang, Wang Qi, and Fu Chunyuan. “CRF-based Chinese Chunking and Event Recognition”. In “Proceedings of the 1st CIPS-ParsEval”, Tsinghua University, Beijing, pp.53-56. 2009.

Yu Hang. “Automatic Analysis of Chinese Chunks”, Graduation thesis of computer science, Tsinghua University，2007.

Yu Kun, Sadao Kurohashi and Hao Liu. “A Three-Step Deterministic Parser for Chinese Dependency Parsing”. In “Proceedings of the Human Language Technologies 2007 (HLT2007-NAACL2007)”, Rochester, pp.201-204, 2007.

Zhou Junsheng, Yabing Zhang, Xinyu Dai, Jiajun Chen. “Chinese Event Descriptive Clause Splitting with Structured SVMs”. In “Proceedings of the 1st CIPS-ParsEval”, Tsinghua University, Beijing, pp.73-80, 2009.

Zhou Qiang, Yumei Li. “The Testing Report of CIPS-ParsEval-2009 Workshop”. In “Proceedings of the 1st CIPS-ParsEval”, Tsinghua University, Beijing, 2009.

Zhou Qiang. “Annotation Scheme for Chinese Treebank”. Journal of Chinese Information Processing, pp 18-21, 2004.

Zhou Qiang, Yume Li. “The Design of Chinese Chunk Parsing Task”， The Tenth Chinese National Conference on Computational Linguistics (CNCCL-2009)，Tsinghua University Press, Beijing, pp.130-135, 2009

Zhou Qiang, Wei Zhang, Shiwen Yu, “Chinese Treebank Construction”, Journal of Chinese Information Processing, pp42-51, 1997.