

Chinese Event Descriptive Clause Splitting with Structured SVMs

Junsheng Zhou^{1,2}, Yabing Zhang¹, Xinyu Dai¹, Jiajun Chen¹

¹(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, 210093)

²(Department of Computer Science, Nanjing Normal University, Nanjing, Jiangsu, 210097)

{Zhoujs, Zhangyb, Dxy, Chenjj}@nlp.nju.edu.cn

Abstract: Chinese event descriptive clause splitting is the task of splitting a complex Chinese sentence into several clauses. In this paper, we present a discriminative approach for Chinese event descriptive clause splitting task. By formulating the Chinese clause splitting task as a sequence labeling problem, we apply the structured SVMs model to Chinese clause splitting. Compared with other two baseline systems, our approach gives much better performance.

1 Introduction

Chinese event descriptive clause splitting is the task of splitting a complex Chinese sentence into several clauses. This task is important for various tasks such as syntactic parsing, machine translation, aligning parallel text and transformation from natural language sentences into logical forms.

In English, there is a similar clause splitting problem presented as a shared-task problem in CoNLL-2001 (Tjong Kim Sang and H. Dejean, 2001). The goal of English clause splitting problem is to identify embedded clauses in text. Considering the difficulty of English clause splitting, the shared task was divided into three parts: identifying clause starts, recognizing clause ends and finding complete clauses. Many machine learning approaches have been developed for English clause splitting. These methods include boosting decision trees and decision graph, neural networks, memory-based learning, statistical, and symbolic learning (Carrerasl and Marquez, 2001; Hammerton, 2001; Tjong Kim Sang, 2001). (Carrerasl and Marquez, 2002) applied the Adaboost algorithm and improved clause identification by using global inference on the top of the outcome clauses hierarchically learned by local classifiers. (Carrerasl and Marquez, 2005) used a discriminative model for it. They applied a global learning algorithm, FR-Perceptron to recognize the structure of clauses. The FR-Perceptron method shows the best result for English clause splitting now. (Nguyen et al. 2009) presented a CRFs-based framework approach to clause splitting, and achieved a result competitive with the state-of-the-art results of clause splitting.

The problem of Chinese event descriptive clause splitting is similar to the third part in the shared-task problem in CoNLL-2001, and Chinese event descriptive clause splitting aims at recognizing the high-level clauses (Qiang Zhou and Yumei Li, 2009). However, there is little work to date on Chinese event descriptive clause splitting problem.

We present a discriminative approach to Chinese event descriptive clause splitting problem.

We formulate Chinese clause splitting as a sequence tagging problem, and learn a discriminative tagger from labeled data using a structured support vector machine (SVM) (I. Tsochantaridis et al., 2005; T. Joachims et al., 2009).

2 Chinese Event Descriptive Clause Splitting Problem

The input to the event descriptive clause splitting splitter is a complete Chinese sentence that is correctly segmented and labeled the part-of-speech (pos) tags. Then the event descriptive clause splitting algorithm recognizes the left and right boundaries of every event descriptive clause to form a sequence of event descriptive clauses. Here is an example of a sentence and its event descriptive clauses obtained from Qinghua treebank:

[只有/c 自身/rNP 硬/a] , /wP [才/d 能/vM 对/p 不良/a 风气/n 、/wD 腐败/a 现象/n 敢/vM 抓/v 敢/vM 管/v] , /wP [不/dN 怕/v “/wLB 鬼/n ”/wRB] , /wP [不/dN 信/v 邪/a] , /wP [敢/vM 摸/v “/wLB 老虎/n ”/wRB 屁股/n] 。/wE
The brackets “[” and “]” in the sentence specify the left and right boundaries of each event descriptive clause respectively.

For English clause splitting problem, a clause splitter is intended to be used after a pos tagger and a chunk parser. Chunks are sequences of consecutive words in the sentence which form the basic syntactic phrases, subject to the constraints that chunks cannot overlap or have embedded chunks. In a correct syntactic tree, clause boundaries are always at some chunk boundaries. However, in Chinese clause splitting task, Chunk tags are not provided for Chinese clause splitter.

In general, an English clause is leaded by an antecedent, which is obviously a formal mark for clause. In contrast to English clauses, there are not any particular marks between Chinese clauses. In Chinese clause splitting, punctuators are often viewed as the separators between clauses. But the use of punctuator is very flexible in Chinese. For instance, the punctuators can be used for separating the functional chunks such as subject, predicate and object. It can also be applied to separating the conjuncts in a functional chunk. Chinese clause splitting is a difficult task.

3 Structured SVMs

The structured support vector machine (SVM) generalizes the Support Vector Machine classifier that supports binary classification, multiclass classification and regression. the structured SVM allows training of a classifier for general structured output labels. Structured classification is the problem of predicting y from x in the case where y having a meaningful internal structure. Elements $y \in Y$ may be, for instance, sequences, trees, or graphs. The major problem for the structured SVMs is the modification of multiple classifications to the very large number of labels problem. To solve the problem, Tsochantaridis et al. (Tsochantaridis, et.al., 2005) presented a re-scaling method for the SVM optimization problem and viewed it as

discriminative classification by employing several loss function and maximization methods. As is typical of discriminative approaches, a feature vector $\Psi(x, y)$ needs to be created to represent a candidate y and its relationship to the input x . In the framework of structural SVMs (Tsochantaridis et al., 2005), training the parameters can be formulated as the following optimization problem (T. Joachims et al., 2009).

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \\ \text{s.t.} \quad & w \cdot \Psi(x_i, y_i) - w \cdot \psi(x_i, y) \geq L_{i,y} - \xi_i, \forall i, y \in Y \setminus y_i; \end{aligned} \quad (1)$$

The training objective is to weight the features using a vector w so that the correct tag sequence receives more weight than the incorrect sequences. The constraints state that the score $w \cdot \Psi(x_i, y_i)$ of the true output y_i must be greater than the score of all alternative alignments y by a difference of $L_{i,y}$. $L_{i,y}$ is a loss function that measures how different the two output y_i and y are. ξ_i is a slack variable shared among constraints from the same example, since in general the problem is not separable.

For most structured problems, the number of constraints in optimization problem (1) is huge and it is infeasible to solve the quadratic program directly. However, it has been shown that the cutting plane algorithm can be used to efficiently approximate the optimal solution of this type of optimization problem (Joachims et al., 2009). The cutting plane algorithm starts with an empty set of constraints, adds the most violated constraint among the exponentially many during each iteration, and repeats until the desired precision is reached.

4 Chinese Event Descriptive Clause Splitting with Structured SVMs

Due to the fact that Chinese clauses are often split at the locations of punctuators, we formulate the Chinese event descriptive clause splitting task as a sequence labeling problem.

4.1 Annotation Method

According to the definition of Chinese event descriptive clause, a clause is a sequence of words separated by a punctuator such as comma, semicolon and interrogation. But these punctuators have very flexible usage, as described in section 2, not limiting to the clause separator. By interpreting every sequence of words separated by a punctuator as a block, we can formulate the Chinese event descriptive clause splitting task as a sequence labeling problem where a block is similar to a token to be labeled in pos tagging problem. For example, the input is the following sentence:

只有/c 自身/rNP 硬/a , /wP 才/d 能/vM 对/p 不良/a 风气/n 、 /wD 腐败/a 现象
/n 敢/vM 抓/v 敢/vM 管/v , /wP 不/dN 怕/v “/wLB 鬼/n ”/wRB , /wP 不/dN 信
/v 邪/a , /wP 敢/vM 摸/v “/wLB 老虎/n ”/wRB 屁股/n 。 /wE

By finding specific punctuators occurring in the sentence, not including the double quotation marks, the complete sentence should be divided into six blocks. For each divided block, we need

to make a decision whether the block forms an independent clause. So, the Chinese clause splitting task is converted into a sequence labeling problem.

We employ a structured SVM that predicts tag sequences, called an SVM Hidden Markov Model, or SVM-HMM. This approach can be considered an hidden Markov model (HMM) because the Viterbi algorithm is used to find the highest scoring tag sequence for a given observation sequence. But it discriminatively trains models that are isomorphic to an k th-order HMM using the structured SVMs formulation. The scoring model employs a Markov assumption: each tag's score is modified only by the tag that came before it. In sequence tagging each input $x = (x_1, \dots, x_n)$ is a sequence of feature vectors, and $y = (y_1, \dots, y_n)$ is a sequence of labels $y_i \in \{1..k\}$ of matching length. Given the trained feature weight vector, the SVM-HMM tags new instances $x = (x_1, \dots, x_n)$ according to:

$$\arg \max_y \left\{ \sum_{i=1}^n \left[\sum_{j=1}^k (x_i \cdot w_{y_{i-j} \dots y_i}) + \varphi_{trans}(y_{i-j}, \dots, y_i) \cdot w_{trans} \right] \right\} \quad (2)$$

In SVM-HMM model, the feature should be divided into two types: emission features and transition features. SVM-HMM learns one emission weight vector $w_{y_{i-j} \dots y_i}$ for each different k th-order tag sequence $y_{i-k} \dots y_i$ and one transition weight vector w_{trans} for the transition weights between adjacent tags. When applying the SVM-HMM to Chinese clause splitting, the crux of the task is the design of suitable feature vectors and the loss function.

4.2 Loss Function

For Chinese clause splitting tasks, there are two possible loss functions: whole-sentence loss and Hamming loss. Whole-sentence loss gives credit only when the entire output sentence is correct: there is no notion of partially correct solutions. Hamming loss is more forgiving: it gives credit on a per label basis. To better express the difference between two outputs, we choose the Hamming loss as the loss function. For a true output y of length N and hypothesized output \hat{y} (also of length N), the Hamming loss functions are given in Eq (3).

$$\ell^{Ham}(y, \hat{y}) = \sum_{n=1}^N \mathbb{1}[y_n \neq \hat{y}_n] \quad (3)$$

4.3 Emission Features

It is difficult to design a suitable set of features to capture the characteristic of Chinese clause, because we formulate a sequence of words (i.e. a block), not a single word, as a “token” to be tagged. According to definition of Chinese clause, a clause should include at least a predicate. But recognizing the predicates of Chinese sentence is difficult. Considering that the verb is a main type of pos that acts as predicate, we instead check whether every block includes a verb within it. By analyzing the instances in training data, we also find that

the first and the last word and pos in every block play an important role in Chinese clause splitting.

We use the lexical and pos information within a fixed window based on blocks. We also consider different combinations of them from the same block or different blocks. The features are listed as follows:

- Word features:
 - The first word in the block
 - The last word in the block
- pos features:
 - The first pos tag in the block
 - The last pos tag in the block
- Punctuator features: Punctuation mark located at the end of the block.
- Combinatorial features:
 - The first word and the last word in the block
 - The first pos tag and the last pos tag in the block
 - The last pos tag in the previous block and the first pos tag in the current block
- The features of inclusion of Verb:
 - Check if there exists a verb in the block.
 - Check if there exist multiple verbs in the block.
 - Check if there exists a noun occurring before the verbs in the block.
- The number of words appearing in the block.

4.4 Postprocessing with Rules

After interpreting a sequence of words separated by a punctuator as a block, we will make a decision whether the block forms an independent clause, to give the right boundary of every Chinese clause. That is, the left boundary of Chinese clause is implicitly decided. However, the double quotation marks is an exceptional punctuation mark acting as a separator of Chinese clauses, because the left double quotation mark sometimes leads to a left boundary of a clause, but it sometimes does not.

By analyzing the training data, we discover a rule: Either both of a pair of quotation marks or neither of them is in a clause. Based on that rule, postprocessing is done in detail as follows:

(1) If the text content between a pair of quotation marks is some simple words, the quotation marks should be contained in a single clause; If the content is a complete sentence, the quotation marks should not be contained in a single clause.

(2) If a left quotation mark is not in a clause, then the corresponding right quotation mark must not be in the clause.

(3) If a left quotation mark is in a clause, then the clause's end position must be behind the corresponding right quotation mark.

(4) If a right quotation mark is in a clause, then the clause's start position must be before the

corresponding left quotation mark.

5 Experiments

This section will evaluate the effectiveness of our approach for Chinese event descriptive clause splitting through experiments.

5.1 Experimental Setting

We used the Qinghua treebank that is adopted by Chinese Information Processing Society of China for ParsEval-2009 as data for training and testing the Chinese event descriptive clause splitting (Qiang Zhou and Yumei Li, 2009). The Chinese event descriptive clause splitting task of ParsEval-2009 included two types of test: open test and closed test. In the open tests participants could use any external data in addition to the training corpus to train their system. In closed tests, participants were only allowed to use information found in the training data. Absolutely no other data or information could be used beyond that in the training document. The data sets contain sentences with the words, the clause split solution, and tagged pos tags. To verify the effectiveness of our approach itself, we only participated in the closed test. That is, the set of features used by our system, as described in section 4.3, only exploited the information from the training data.

5.2 Experimental Results

For evaluating the task in a set of N sentences, the usual precision, recall and F1 measures are used:

$$\text{Precision} = \frac{\text{num of correctly recognized clauses}}{\text{num of recognized clauses}}$$

$$\text{Recall} = \frac{\text{num of correctly recognized clauses}}{\text{num of total clauses}}$$

$$\text{F1} = \frac{\text{Precision} \times \text{Recall} \times 2}{\text{Precision} + \text{Recall}}$$

To compare the effectiveness of our approach with other approaches, we first developed two baseline systems. The first baseline system is based on a decision tree classifier, and the second one uses the CRFs model (Lafferty J, et al. 2001) to split the Chinese clauses. Then we implemented the third system with the method proposed in this paper. In the first baseline system, the decision on whether the current block forms a complete clause is made independently with a decision tree classifier, and we use j48 algorithm in Weka as the decision tree classifier (IH Witten and E. Frank, 2000). In the second baseline system, we also formulate the Chinese clause splitting task as a sequence labeling problem, and choose the CRFs model as the sequence tagging model, because a good result achieved by CRFs model for English clause splitting was reported in (Vinh Van Nguyen et al. 2009). The results of the first baseline system and the second

baseline system are shown in table 1. The second baseline system with CRFs model achieved better performance than the first baseline system with decision tree classifier.

Table 1 The results of two baseline systems

	Precision	Recall	F1
Decision tree	68.36	74.59	71.34
CRFs	73.37	76.64	74.97

When applying the SVM-HMM model to Chinese clause splitting, we can train different order models to express different length dependencies for both the transitions and the emissions. The results in table 2 show the performance of the applying 1th-order, 2th-order and 3th-order models to Chinese clause splitting task, respectively. We can observe that the higher order of model leads to a better performance. The fact also indicates that it is reasonable to treat the Chinese clause splitting task as a sequence tagging problem. Compared with other two baseline systems, our approach gives much better performance.

Table 2 The results of Our systems with different orders

	Precision	Recall	F1
1th-order model	75.15	76.99	76.06
2th-order model	75.95	78.82	77.36
3th-order model	76.36	80.02	78.15

6 Conclusion

In this paper, we explore a discriminative approach for Chinese event descriptive clause splitting task. By formulating the Chinese clause splitting task as a sequence labeling problem, we apply the structured SVMs model to Chinese clause splitting. We compare the approach proposed in this paper with two baseline systems and the experimental results show that our approach achieves a much better result. We will try to select more useful feature functions into the existing sequence tagging model in future work.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant No.60673043, 60773173; and the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant No. 07KJB520057.

References

Y. Altun, I. Tsochantaridis, T. Hofmann. 2003. Hidden Markov Support Vector Machines. In Proceedings of International Conference on Machine Learning (ICML).

- Xavier Carreras and Lluís Màrquez. 2001. Boosting Trees for Clause Splitting, In Proceedings of the CoNLL-2001 Shared Task.
- Xavier Carreras, Lluís Màrquez, Vasin Punyakanok and Dan Roth. 2002. Learning and Inference for Clause Identification. In Proceedings of 13th European Conference on Machine Learning (ECML). Helsinki, Finland.
- Xavier Carreras, Lluís Màrquez and Jorge Castro. 2005. Filtering-Ranking Perceptron Learning for Partial Parsing. Machine Learning Journal, Special Issue on Learning in Speech and Language Technologies, Volume 60, Issue 1-3, pages 41-71.
- Lafferty J, McCallum A, Pereira F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the International Conference on Machine Learning (ICML). San Francisco: Morgan Kaufmann Publishers, 282–289.
- Hammerton, James. 2001. Clause identification with Long Short-Term Memory. In Proceedings of CoNLL-2001, pages 61-63. Toulouse, France.
- T. Joachims, T. Finley, Chun-Nam Yu. 2009. Cutting-Plane Training of Structural SVMs, Machine Learning, 77(1):27-59.
- Vinh Van Nguyen, Minh Le Nguyen and Akira Shimazu. 2009. Clause Splitting with Conditional Random Fields. Information and Media Technologies. Vol. 4, No. 1 pp.57-75.
- E. F. Tjong Kim Sang and H. Dejean. 2001. Introduction to the CoNLL-2001 shared task: Clause identification. In W. Daelemans and R. Zajac, editors, In Proceedings of CoNLL-2001, pages 53-57.
- I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables, Journal of Machine Learning Research (JMLR), 6(Sep):1453-1484.
- IH Witten, E. Frank. 2000. Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann, San Francisco.
- Qiang Zhou, Yumei Li. 2009. Chinese Chunk Parsing Evaluation Tasks. Advances of computational Linguistics in China. Tsinghua University Press.
- Erik F. Tjong, Kim Sang. 2001. Memory-based clause identification. In Proceedings of CoNLL-2001, pages 67-69. Toulouse, France.