北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis
doi: 10.13209/j.0479-8023.2016.014

A New Ranking Method for Chinese Discourse Tree Building

WU Yunfang[1,†], WAN Fuqiang[1], XU Yifeng[1], LV Xueqiang[2]

1. Key Laboratory of Computational Linguistics (MOE), Peking University, Beijing 100871; 2. Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing Information Science and Technology University, Beijing 100192;
† E-mail: wuyf@pku.edu.cn

Abstract This paper proposes a novel method for sentence-level Chinese discourse tree building. The authors constrcut a Chinese discourse annotated corpus in the framework of Rhetorical Structure Theory, and propose a ranking-like SVM (SVM-R) model to automatically build the tree structure, which can capture the relative associated strength among three consecutive text spans rather than only two adjacent spans as most previous approaches do. The experimental results show that proposed SVM-R method significantly outperforms the state-of-the-art in discourse parsing accuracy. It is also demonstrated that the useful features for discourse tree building are consistent with Chinese language characteristics.

Key words discourse tree building; ranking method; Chinese discourse annotated corpus

基于排序方法的汉语句际关系树自动分析

吴云芳[1,†] 万富强[1] 徐艺峰[1] 吕学强[2]

1. 计算语言学教育部重点实验室，北京大学，北京 100871; 2. 网络文化与数字传播北京市重点实验室，北京信息科技大学，北京 100192; † E-mail: wuyf@pku.edu.cn

摘要 提出一种自动分析汉语小句级句际关系树的新方法。在修辞结构理论体系下，构建了一个汉语句际关系标注语料库。不同于传统方法只关心相邻两个单元的方法，提出一种类排序模型(SVM-R)，自动构建汉语句际关系的树结构，旨在把握相邻 3 个单元之间的关联强度。实验结果表明，所提出的 SVM-R 模型对句际关系树的分析显著优于传统方法。最后提出并验证了丰富的、适合于汉语句际关系分析的语言特征。

关键词 句际关系树构建；排序方法；汉语句际关系语料库
中图分类号 TP391

In recent years automatic discourse analysis has attracted many researchers' interest, and has been shown to be of great help in many natural language processing tasks such as summarization, question answering and machine translation.

Rhetorical Structure Theory (RST) [1] is one of the most influential theories of discourse, which posits a tree representation of a discourse. In the RST-framework, a complete discourse analysis system involves three subtasks: 1) breaking the texts into elementary discourse units (EDUs) (known as discourse segmentation); 2) linking all EDUs into a hierarchical tree structure (known as tree building); 3) assigning a relation label between two adjacent

国家自然科学基金(61371129)、国家重点基础研究发展计划(2014CB340504)、国家社会科学基金重大项目(12&ZD227)和网络文化与数字传播北京市重点实验室开放课题 (ICDD201302)资助
收稿日期: 2015–06–04; 修回日期: 2015–08–17; 网络出版时间: 2015-09-29 12:51:19

subtrees (known as relation labeling). The above three steps are often conducted in a pipeline fashion[2–4]. Discourse segmentation is a relatively easy task, and many researches neglect this task and use manual EDU segmentation. Tree building is a vital early step of relation labeling, and so plays a crucial role for discourse parsing. Relation labeling is a common task shared by the RST-framework researches and the researches based on the Penn Discourse Treebank (PDTB)[5], and so many efforts have been done on this task[2–3,6–7].

This paper aims to automatically build the hierarchical tree structure for discourse parsing. Previous studies on discourse tree building have been successful in identifying what machine learning approaches and what kind of features are more useful. However, these proposed solutions suffer from a key limitation. They train a binary classifier to evaluate whether a discourse relation is likely to hold between two consecutive units (S_i, S_{i+1}), and apply a greedy, bottom-up approach to build the structure of a discourse tree. Behind these approaches, they make strong independence assumptions on ((S_i, S_{i+1}), S_{i+2}).That is, the connected probability of two adjacent units (S_i, S_{i+1}) is independent with the third adjacent unit S_{i+2}. This is not consistent with our language intuition: whether two adjacent units (S_i, S_{i+1}) should be first merged into a discourse subtree not only depends on the connected probability between them but also depends on the relative connected probability among the three consecutive spans (S_i, S_{i+1}, S_{i+2}). In this paper, we propose a new ranking-like support vector machine model (SVM-R) to address this limitation, which can capture the relative associated strength among three consecutive text spans. In the experiment, our proposed SVM-R model obtains significant improvement over the state of the art for Chinese discourse tree construction.

This paper aims to tackle with the Chinese discourse analysis. Although there is a large body of work on English discourse analysis, much less work has been done in Chinese. Chinese language is quite different from other western languages such as English. Chinese is a discourse-oriented language while English is a sentence-oriented language[8]. In this paper, we construct a Chinese discourse corpus in the RST-framework, and investigate what kind of features are helpful for Chinese discourse tree building.

Overall, in building the tree structure for discourse analysis, our contributions are in the following two aspects.

1) We propose a ranking-like SVM model to capture the relative associated strength among three consecutive text spans, and obtain significant improvement over the state-of-the-art in discourse parsing accuracy. Because the SVM-R model is simple to implement and is effective in performance, it is potentially useful for other tasks in the NLP field.

2) We construct a Chinese discourse annotated corpus in the RST framework, and propose a variety of linguistic features for Chinese discourse tree building. To the best of our knowledge, this is the first work to cope with Chinese discourse tree construction in the RST-framework.

1 Related Work

1.1 RST discourse tree

RST is one of the most widely used discourse theories in natural language processing. In RST, a coherent text can be represented as a discourse tree, whose leaves are non-overlapping text spans called EDUs. Adjacent nodes are related through particular discourse relations to form a discourse subtree, which can then be related to other adjacent nodes to form a discourse tree.

The RST Discourse Treebank (RST-DT)[9] is an English discourse annotated corpus in the framework of RST. It consists of 385 documents from the Wall Street Journal. RST-DT is widely used in English discourse parsing.

Another popular discourse corpus is the Penn Discourse Treebank (PDTB)[5]. It follows a lexically-grounded, predicate-argument structure. A discourse connective is regarded as a predicate that takes two text spans as its arguments (Arg1, Arg2). An

important difference between PDTB and RST-DT is that in PDTB, there does not necessarily form a tree structure covering the full text.

1.2 Discourse parsing

The researches of discourse parsing can be clustered into two groups: RST framework and PDTB framework. Here we only mention those researches in the framework of RST that are more related to our work.

In the RST framework, many different approaches[2–4,10–13] have been proposed, which extract different textual information and adopt various algorithms for discourse tree building. Here we briefly review some of them that are close to our work.

SPADE[10] is a sentence-level discourse parser, which extract syntactic and lexical information to parse a text. It demonstrates the close correlation between the syntactic structure and discourse information. HILDA[2] is the first fully-implemented English discourse parser. They employ two cascade classifiers to deal with tree construction and relation labeling respectively. In the tree building, a binary SVM classifier is trained to determine whether two adjacent text spans should be merged to form a subtree. In the relation labeling, a multi-class SVM classifier is trained to determine which kind of discourse relations should be labeled in the subtree. HILDA obtains 85.0% accuracy for tree construction and 66.8% accuracy for 18-class relation labeling. Later, Feng et al.[3] adopt HILDA parser as the basis, and incorporate more linguistic features to improve the performance of parsing accuracy. Recently, Feng et al.[4] train two linear-CRFs as local classifiers to deal with tree construction and relation labeling in a cascade way. They propose a post-editing method by taking account of the upper-level information in the tree structure, and obtain promising improvement.

As noted before, the above proposed approaches are designed to learn the related probability between two adjacent text spans, but they cannot capture the relative associated strength among three consecutive spans that is of great help for discourse tree building.

Joty et al.[13] propose a probabilistic discriminative approach for sentence-level discourse analysis. They employ a Dynamic Conditional Random Field (DCRF) model to jointly determine the tree structure and discourse relations. Joty et al.[14] propose a discriminative approach to re-rank discourse trees relying on tree kernels. Their model is quite complex and has a high order of time complexity, and thus cannot be applied in a real task.

1.3 Chinese discourse analysis

There is a large body of work on English dis-course analysis, but much less work has been done in Chinese discourse parsing.

Huang et al.[15] build a PDTB-style Chinese discourse corpus, based on Taiwan Sinica corpus. They train a SVM classifier to predict 4 classes of discourse relations, and obtain an f-score 63.69%. Yang et al.[16] cluster the Chinese comma into seven hierarchical categories and then train classifiers to resolve the com-ma disambiguation problem. Zhou et al.[17] describe a PDTB-style discourse annotation scheme for Chinese, but have to make many adaptations to suit for the linguistic characteristics of Chinese text. Li et al.[18] propose a connective-driven dependency tree (CDT) scheme to represent the discourse rhetorical structure in Chinese, but we think Chinese is a paratactic language and so it is very difficult to insert an explicit connective to some implicit discourse relations.

Different from most work on PDTB scheme, we follow the RST theory to construct Chinese discourse trees. Different from the work of Ref. [18], we do not insert an explicit connective to each implicit discourse relation, and let alone the lexical and syntactic information to carry the subtle discourse meanings.

2 Chinese Discourse Annotated Corpus

We aim to build a large-scale Chinese discourse annotated corpus in the RST-framework. Here we briefly describe the main ideas in building Chinese discourse trees.

As noted by previous studies[8,19], Chinese is topic-prominent and so is a discourse-oriented language, while English is subject-prominent and so is

a sentence-oriented language.

These two linguistic features result in a fact that there is not a clear distinction between Chinese discourses and sentences. Although marked explicitly by a full stop, a Chinese sentence can often consist of quite a few predicate-argument structures. Compared with English, many Chinese sentences have more tokens in length and are more complex in the tree structure.

It is difficult and confused for sentence-level Chinese discourse analysis in two aspects: how to determine a sentence and how to determine an EDU. In our work, we deal with these two issues in a straightforward and practical way: the text span explicitly marked with a full stop is a sentence; and the text span split by a comma is regarded as an EDU.

We follow the methodology of RST-DT to construct Chinese discourse trees. In a sentence, two adjacent spans are related with a particular discourse relation, which can then be related to another adjacent span to form a tree structure. We define 7 classes and 18 subclasses of discourse relations in our annotation scheme, as listed in Table 1. In order to deal with all discourse relations in an unified way, the multi-nuclear relation (conjunction) is transformed into a right-branching structure.

Table 1 Chinese discourse relation types

Class	Subclass
conjunction	coordinate, alternative, temporal, progressive, succession
comparison	contrast, concession
inference	cause, result, purpose
condition	hypothetical, condition
specification	explanation, list
summary	generalization
background	topic, attribute, marker

An example of Chinese discourse tree is shown in Fig. 1, where the leaf nodes e1, e2, e3, e4 are fourEDUs, and the non-leaf nodes are represented in the form of relation (span1, span2). For example, RESU(e2, e3-e4) denotes that there holds a result relation between e2 and e3-e4.

Up to now, we have built a Chinese discourse annotated corpus on *People's Daily News*, containing more than 8000 sentences. All the data was manually annotated by a retired professor major in linguistics. We selected 1,000 sentences for inter-annotator agreement evaluation, and all 1000 sentences are complex and consist of more than two EDUs. We got a Kappa value of 0.71 in the tree construction,

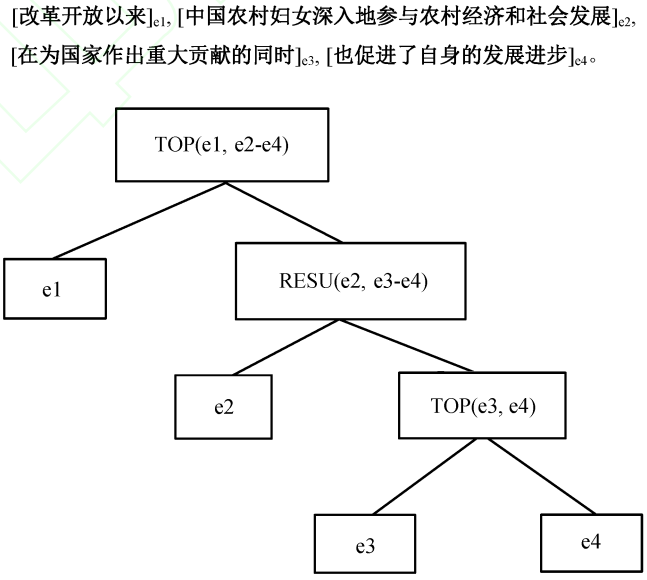

Fig. 1 An example of Chinese discourse tree

indicating that building the Chinese discourse tree is a non-trivial task.

3 Different SVM Models

We employ two SVM models to build the Chinese discourse tree: SVM-H and SVM-R. SVM-H is similar with the previous approach of HILDA[2], which can be regarded as the baseline of our work. SVM-R is our proposed ranking-like model, which is designed to capture the dependence among three consecutive elements.

We adopt SVM classifiers to do the discourse tree building, since SVM model has been widely used in discourse analysis and has been shown to be effective[2–3,20].

Instance extraction is a crucial step for discourse tree construction. The main differences between SVM-H and SVM-R are: 1) how to extract positive and negative instances in training data; 2) how to incorporate different linguistic features into the model. In order to illustrate these differences in a unified way in the next subsections, we give the following two discourse trees as examples.

In Fig. 2, the leaf nodes *A*(1), *B*(2), *C*(3), *D*(4) represent four EDUs, and the letter A is just to denote the node; the non-leaf nodes are represented with the form *E*(1, 2), denoting that text span 1 and 2 constitute a subtree marked with *E*. Please note that the representation in Fig. 2 is a little different from Fig. 1, where we neglect the relation information, since our work is just to build the tree structure instead of doing the complete discourse parsing.

3.1 SVM-H

This method follows the methodology of HILDA. Initially, a binary structure classifier is trained to evaluate whether a discourse relation is likely to hold between two consecutive elements. Then, the two elements which are most probably connected by a discourse relation are merged into a discourse subtree. This method applies a greedy, bottom-up algorithm to build the Hierarchical structure of the discourse tree, so we name it SVM-H.

The instance extraction of SVM-H is similar with that of Ref. [3]. Each instance is of the form (S_i, S_{i+1}), which is a pair of adjacent text spans extracted from the discourse annotated corpus. The positive instances are extracted as those pairs of text spans that share the same parent node in the discourse tree, while the negative instances as those pairs of text spans that are not siblings of the same parent node. In SVM-H, the text spans S_i and S_{i+1} can be either an EDU or a constituent consisting of multiple consecutive EDUs.

In the above example shown in Fig. 2(a), SVM-H extracts three positive instances <*A*, *B*, +>, <*C*, *D*, +>, <*E*, *F*, +> and three negative instances <*B*, *C*, –>, <*E*, *C*, –>, <*B*, *F*, –>. We can see that <*E*, *F*, +> is extracted as a positive example while <*E*, *C*, –> is a negative one. The difference between <*E*, *F*> and <*E*, *C*> is only that *F* contains an additional EDU *D*(4), which actually contributes little in the feature extraction for the text span *F*. As a result, the features between <*E*, *F*, +> and <*E*, *C*, –> are almost the same but the former is positive and the latter is negative. Thus, the almost same features (the almost same vectors in SVM) are assigned to different clusters, and

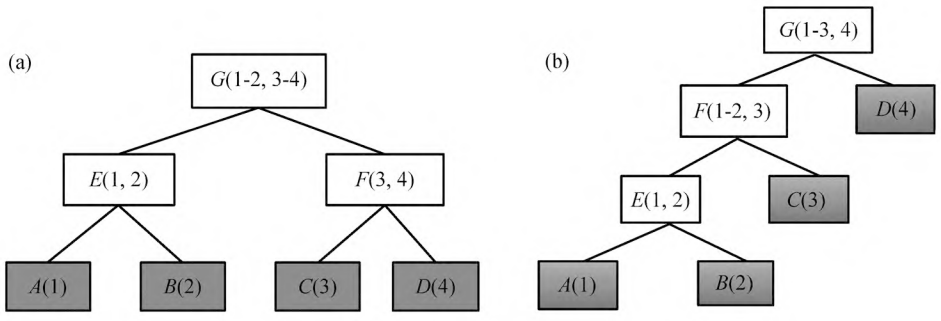

Fig. 2 Examples of discourse trees

will confuse the classifier. In Fig. 2(b), SVM-H extracts three positive instances <*A*, *B*, +>, <*E*, *C*, +>, <*F*, *D*, +> and two negative instances <*B*, *C*, –>, <*C*, *D*, –>.

3.2 SVM-R

The SVM-H method is designed to compute an absolute probability score between two adjacent spans (S_i, S_{i+1}) to determine whether they are likely to merge into a discourse subtree. This method (and most of the previous approaches) can be formalized with the following equation:

$$f(\boldsymbol{w}, e_{i-1}, e_i) = \boldsymbol{w}^{\mathrm{T}} \Phi(e_{i-1}, e_i), \quad (1)$$

where e denotes an element, Φ represents feature vectors and w denotes the learned weight. This method makes a strong independence assumption on ((e_{i-1}, e_i), e_{i+1})). However, in our task, according to our linguistic knowledge, whether (e_{i-1}, e_i) should be first merged into a discourse subtree depends on the relative associated strength among the three consecutive elements (e_{i-1}, e_i, e_{i+1}) rather than only two adjacent elements (e_{i-1}, e_i).

Therefore, we propose a new method to build the discourse tree structure. Our method is inspired by the Ranking-SVM[21]. Ranking-SVM is widely used in information retrieve to rank the returned documents ($d_i \in D$) for a given query q. It constructs a feature vector $F_i = f(d_i, q)$ for each pair of (d_i, q), and then employs the difference vector $f_i - F_j$ as a new feature to train a SVM classifier. The returned value (+1) of a SVM classifier means that the document d_i is more relative to the query q than the document d_j and vice versa. Thus, the ranking problem is transformed to a classification task. Ranking-SVM can be formalized with the following equation:

$$f(\boldsymbol{w}, d_i, d_j, q) = \boldsymbol{w}^{\mathrm{T}}[\Phi(d_i, q) - \Phi(d_j, q)]. \quad (2)$$

Inspired by the Ranking-SVM, our method SVM-R is designed to capture the relative associated strength among three elements. It can be formalized with the following Eq. (3):

$$f(\boldsymbol{w}, e_{i-1}, e_i, e_{i+1}) = \boldsymbol{w}^{\mathrm{T}} \Phi(e_{i-1}, e_i, e_{i+1}) \quad (3)$$

Different from Ranking-SVM in Eq. (2), SVM-R extracts features directly from three elements rather than using the vector difference between two pairs. Different from SVM-H in Eq. (1), SVM-R extracts features among three consecutive elements rather than only two adjacent elements.

The instance extraction of SVM-R is different from SVM-H. Each instance is of the form (S_i, S_{i+1}, S_{i+2}), which are three consecutive text spans in the discourse tree. If the latter two spans (S_{i+1}, S_{i+2}) constitute a node in the tree, (S_i, S_{i+1}, S_{i+2}) should be extracted as a positive instance; if the former two spans (S_i, S_{i+1}) constitute a node in the tree, (S_i, S_{i+1}, S_{i+2}) should be extracted as a negative instance.

In the above example shown in Fig. 2(a), SVM-R extracts two positive instances <*B*, *C*, *D*, +>, <*E*, *C*, *D*, +> and two negative instances <*A*, *B*, *C*, –>, <*A*, *B*, *F*, –>. In Fig. 2(b), SVM-R extracts two negative instances <*A*, *B*, *C*, –>, <*E*, *C*, *D*, –>.

Algorithm 1 gives the details of tree construction of SVM-R. It also conducts a greedy, bottom-up approach. In the 8th line, ($v[i]$–θ) is to measure the probability difference between two adjacent pairs (S_i, S_{i+1}) and (S_{i-1}, S_i). When $v[i]$–θ>0, it means that the pair (S_i, S_{i+1}) has a stronger association than the pair (S_{i-1}, S_i) and should be first merged, and vice versa. θ is a tuned parameter: when θ=0 the tree structure should be left-branch; and when θ=1 the tree structure should be right-branch. Since the probability of right-branching structure in Chinese discourse tree is above 0.5, θ ranges between (0, 0.5). We set it to 0.25 in our experiment.

Algorithm 1: SVM-R tree construction

Input: A complex sentence CS

1. CS ← <S_1, …, S_n>
2. while |CS|>1 do
3. for all (S_{i-1}, S_i, S_{i+1}) in CS do
4. $v[i]$←SVM-R(S_{i-1}, S_i, S_{i+1})
5. end for
6. scores[1] ← 0
7. for i←2 to n–1
8. scores[i] = scores[i–1]+ $v[i]$–θ
9. end for

10. i←argmax(scores)
11. S←merge(S_i, S_{i+1})
12. CS←<S_1, …, S_{i-1}, S, S_{i+2}, …, S_n>
13. delete(scores[i])
14. end while
15. CS^T ←1st element of CS

Output: A tree structure CS^T

4 Features

The experimental data was automatically word-segmented and POS-tagged using the open soft-ware ICTCLAS. In our experiment, we utilize only the surface lexical and POS features, but do not use syntactic tree features that are used in most of previous studies on English discourse parsing. We are dealing with the Chinese complex sentences that have many tokens in length and contain a few sub-sentences, and so the automatic Chinese parser cannot provide us a satisfying result in the syntactic tree. Actually, in our early experiments, we have implemented the syntactic tree features provided by Stanford Parser, but we obtain little performance gain.

Table 2 lists 9 sets of features we used. In this table, those features that are symmetrically extracted from both left and right candidate spans are denoted by S(pan); those features calculated as a function of the two spans as a pair are denoted by F(ull).

Discourse connectives: The discourse connectives occurring in each span S_i (or S_{i+1}). In our dictionary, we list 139 discourse connectives that are frequently used in Chinese real text.

Table 2 The used features

Features	Scope
discourse connectives	S
length in EDUs	S
first EDU	S
length difference in tokens	F
separated by a semicolon	F
beginning *n*-grams	S
the same verb	F
number word	S
auxiliary word *zhe*|着 or *le*|了 or *guo*|过	S

Length in EDUs: The number of EDU containing in each span S_i (or S_{i+1}).

First EDU: Whether S_i (or S_{i+1}) contains the first EDU. In a Chinese sentence, the first EDU often serves as a topic, and locates in a high level of the discourse tree.

Length difference in tokens: The length difference between two adjacent spans.

Separated by a semicolon: Whether two spans (S_i, S_{i+1}) is separated by a semicolon.

Beginning *n*-grams: The beginning lexical *n*-grams in each span S_i (or S_{i+1}), where $n \in \{1,2\}$. This feature is designed to find out the lexicalization discourse cues like "致使|resulting in".

The same verb: The same verb contained both in the two adjacent spans (S_i, S_{i+1}).

Number word: Whether S_i (or S_{i+1}) contains number word, which is denoted by the POS tag "m", such as "100 万|one million".

Auxiliary word *zhe*|着 or *le*|了 or *guo*|过: The auxiliary words *zhe*|着 or *le*|了 or *guo*|过 contained in each span S_i (or S_{i+1}). Due to the absence of morphological change, Chinese language uses tense auxiliary words *zhe*|着 or *le*|了 or *guo*|过 to represent the time information involved in events. If two text spans share the same tense auxiliary word, they are connected closely and should be first merged to form a discourse subtree.

5 Experiments

5.1 Experimental setup

As mentioned in Section 3, we have manually annotated more than 8000 sentences in People's Daily News. We removed those sentences containing 1) less than 3 EDUs because there is no need for them to build the tree structure and 2) more than 10 EDUs because they are rare in real texts and are too complicated to handle. Finally, we got 4747 sentences for our experiment, and all the data can be freely downloaded in our website. We use 4/5 of the sentences (3799) as training data and 1/5 (948) as test data. We also conduct a 5-fold cross-validation in the training data.

We extracted positive and negative instances from all the training data for SVM-H. However for SVM-R, from the training data, we removed those discourse trees that are all right-branching or are all left-branching in the tree structure, in order to keep a balance between positive and negative instances. But please note that in all situations the test data for both methods keeps the same. Table 3 lists the number of extracted instances for training by two methods in the held-out test set. The number of training instances of SVM-R is much smaller (about 1/6) than that of SVM-H.

Table 3 Training instances of two methods

Type	SVM-R	SVM-H
negative	2671	14661
positive	2671	15530
total	5342	30191

Macro Accuracy (mac-a) and Micro Accuracy (mic-a) are used to evaluate the performance of two methods SVM-H and SVM-R.

$$\text{mac-a} = \frac{\text{number of sentences wih correctly predicted disourse structure}}{\text{number of all sentences}}, \quad (4)$$

$$\text{mic-a} = \frac{\text{number of nodes with correctly predicted structure}}{\text{number of nodes in all sentences}}. \quad (5)$$

In training process, all classifiers are trained on the basis of individual instances. The test procedure is different for two evaluation metrics: mac-a tests on the tree structure of a full sentence, while mic-a tests on the basis of individual nodes of discourse subtrees. We adopt LibSVM package to do our experiments, with a linear kernel and all default values.

5.2 Experimental results

Table 4 and 5 report the evaluation results of parsing accuracy. Table 4 reports the results of the held-out test set, and it shows that our proposed SVM-R model performs significantly better than the previous approach SVM-H (p-value <0.01), with 1.7% increase in mac-a and 2.2% increase in mic-a. Table 5 reports the results of 5-fold cross-validation in the training data, and it shows that our SVM-R model is consistently better than SVM-H.

Table 4 Results in the standard test data

Model	mac-a	mic-a
SVM-H	59.5	64.6
SVM-R	61.2	66.8

Table 5 Results in 5-fold cross-validation

Fold	mac-a		mic-a	
	SVM-H	SVM-R	SVM-H	SVM-R
1	58.9	60.9	64.1	67.1
2	57.9	60.1	63.8	66.2
3	58.9	60.3	63.1	66.3
4	58.8	60.1	64.7	66.4
5	57.7	60.4	64.2	66.3
Ave.	58.4	60.4	64.0	66.5

5.3 Different features

We further investigate what kind of features are more useful for Chinese discourse tree construction, by demonstrating the performance drop when subtracting each feature in our SVM-R model. Table 6 reports the results.

In Chinese discourse tree building, the top two important features are beginning n-grams and first EDU. To our surprise, these two features are well suited to two salient properties of Chinese language. First, Chinese is a paratactic language, and compared

Table 6 Effects of different features

Features	Performance drop	
	mic-a	mac-a
beginning n-grams	2.07	2.11
first EDU	1.85	1.71
separated by a semicolon	1.09	0.85
the same verb	0.37	0.32
auxiliary word *zhe*|着 or *le*|了 or *guo*|过	0.36	0.21
discourse connectives	0.32	0.74
number word	0.11	0.55
length difference in tokens	0.04	0.12
length in EDUs	0.01	0.01

with English, much less explicit connectives are used to connect two text spans, thus the lexicalization cues represented by beginning *n*-grams play a far more important role than discourse connectives. Second, Chinese is a topic-prominent language, and the first EDU often serves as a topic and is often in the highest level of the tree structure (for example in Fig. 1, "改革开放以来| since the reform and opening" is a topic), making first EDU a crucial feature for Chinese discourse tree construction.

5.4 Parameter

In our SVM-R model, as shown in Algorithm 1, there is a tuned parameter θ to measure the probability difference among three elements. We also investigate the effect of θ value on the performance, as illustrated in Fig. 3. When θ value is in between (0, 0.45], SVM-R outperformers SVM-H; when θ is above 0.45, the performance of SVM-R drops a little.

6 Conclusion

In this paper, we build a Chinese discourse annotated corpus in the RST-framework, and then we propose a ranking-like SVM-R model, which aims to capture the relative associated strength among three consecutive elements, to automatically construct the Chinese discourse tree. In the experiment, our SVM-R model significantly outperforms the previous approach. In addition, we present and analyze some useful features for Chinese discourse tree building.

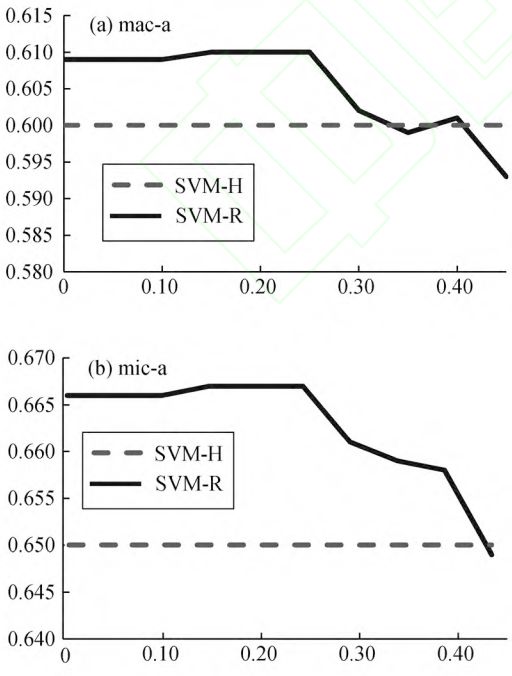

Fig. 3 Effect of θ value

There is still a lot of room to improve for Chinese discourse tree construction. In future work, we wish to extract more informative linguistic features, including the syntactic tree features learnt from the deep processing of texts and the word embedding features learnt from large unlabeled data. On the other hand, we wish to apply our SVM-R model to other languages and other data, such as the English RST-DT corpus. Also our SVM-R model can be applied to other similar tasks in the NLP field.

References

[1] Mann W, Thompson S. Rhetorical structure theory: toward a functional theory of text organization. Text, 1988, 8(3): 243–281

[2] Hernault H, Prendinger H, duVerle D, et al. HILDA: a discourse parser using support vector machine classification. Dialogue and Discourse, 2010, 1(3): 1–33

[3] Feng V, Hirst G. Text-level discourse parsing with rich linguistic features // Proceedings of ACL-2012. Jeju: Association for Computational Linguistics, 2012: 60–68

[4] Feng V, Hirst G. A linear-time bottom-up discourse parser with constraints and post-editting // Proceedings of ACL-2014. Baltimore: Association for Computational Linguistics, 2014: 511–521

[5] Prasad R, Dinesh N, Lee A, et al. The Penn Discourse Treebank 2.0 // Proceedings of the 6th International Conference on Language Resources and Evaluation. Marrakech, 2008: 2961–2968

[6] Lin Z, Kan M Y, Ng H T. Recognizing implicit discourse relations in the Penn Discourse Treebank // Proceedings of EMNLP-2009. Singapore: Association for Computational Linguistics, 2009: 343–351

[7] Pitler E, Louis A, Nenkova A. Automatic sense prediction for implicit discourse relations in text // Proceedings of ACL-2009. Singapore: Association for Computational Linguistics, 2009: 683–691

[8] Cao F, Xie T. Functional study of topic in Chinese: the first step towards discourse analysis. Beijing: Language and Culture Press, 1995

[9] Carlson L, Marcu D, Okurowski M E. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory // Proceedings of Second SIGdial Workshop on Discourse and Dialogue. Enschede: Springer. 2003: 85–112

[10] Soricut R, Marcu D. Sentence level discourse parsing using syntactic and lexical information // Proceedings of NAACL-2003. Edmonton: Association for Computational Linguistics, 2003: 149–156

[11] Subba R, Eugenio B D. An effective discourse parser that uses rich linguistic information // Proceedings of NAACL-2009. Colorado: Association for Computational Linguistics, 2009: 566–574

[12] Sagae K. Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing // Proceedings of the 11th International Conference on Parsing Technologies. Paris: Association for Computational Linguistics, 2009: 81–84

[13] Joty S, Carenini G, Ng R T. A novel discriminative framework for sentence-level discourse analysis // Proceedings of EMNLP-2012. Jeju: Association for Computational Linguistics, 2012: 904–915

[14] Joty S, Moschitti A. Discriminative reranking of discourse parses using tree kernels // Proceedings of EMNLP-2014. Doha: Association for Computational Linguistics, 2014: 2049–2060

[15] Huang H, Chen H. Chinese discourse relation recognition // Proceedings of IJCNLP-2011. Chiang Mai: Association for Computational Linguistics, 2011: 1442–1446

[16] Yang Y, Xue N. Chinese comma disambiguation for discourse analysis // Proceedings of ACL-2012. Jeju: Association for Computational Linguistics, 2012: 786–794

[17] Zhou Y, Xue N. PDTB-style discourse annotation of Chinese text. In Proceedings of ACL-2012. Jeju: Association for Computational Linguistics, 2012: 69–77

[18] Li Y, Feng W, Sun J, et al. Building Chinese discourse corpus with connective-driven dependency tree structure // Proceedings of EMNLP-2014. Doha: Association for Computational Linguistics, 2014: 2105–2114

[19] Li N, Thompson A. Subject and topic: a new typology of languages // Subject and Topic. New York: Academic Press, 1976: 457–489

[20] duVerle D, Prendinger H. A novel discourse parser based on support vector machine classification //Proceedings of ACL-2009. Singapore: Association for Computational Linguistics, 2009: 665–673

[21] Joachims J. Optimizing search engines using click-through data // Proceedings of the ACM Conference on Knowledge Discovery and Data Mining. New York, 2002: 133–142