Chinese Textual Entailment Recognition Based on Syntactic Tree Clipping

Zhichang Zhang, Dongren Yao, Songyi Chen, Huifang Ma

School of Computer Science and Engineering, Northwest Normal University, Lanzhou, China

zzc@nwnu.edu.cn; wade330628704@163.com;
snail200x@163.com; mahuifang@nwnu.edu.cn

Abstract. Textual entailment has been proposed as a unifying generic framework for modeling language variability and semantic inference in different Natural Language Processing (NLP) tasks. This paper presents a novel statistical method for recognizing Chinese textual entailment in which lexical, syntactic with semantic matching features are combined together. In order to solve the problems of syntactic tree matching difficulty and tree structure errors caused by Chinese word segmentation, the method firstly clips the syntactic trees into minimum information trees and then computes syntactic matching similarity on them. All features will be used in a voting style under different machine learning methods to predict whether the text sentence can entail the hypothesis sentence in a text-hypothesis pair. The experimental results show that the feature on changing structure of syntactic tree is effective and efficient in Chinese textual entailment.

Keywords: Textual Entailment; Minimum information tree; Syntactic tree clipping; Machine learning

1 Introduction

The Recognizing Textual Entailment (RTE) challenge focuses on detecting the directional entailment relationship between pairs of text expressions, denoted by *T* (the entailing “Text”) and *H* (the entailed “Hypothesis”). We say that *T* entails *H* if human reading *T* would typically infer that *H* is most likely true.

RTE is proposed as a generic task that captures the semantic inference demand with a wide range of natural language applications. For example, a question answering system needs to recognize the text whether entails a hypothesized answer, e.g., given the question “*Which team won the NBA championship in* 2012-2013?”, the text “*James led Miami Heat to their second straight title on June* 20,2013” entails the hypothesized answer form “*Miami Heat win the championship in* 2012-2013”.

RTE has attracted extensive attention ever since it was proposed, and researchers have developed many methods to solve this problem. These methods can be roughly classified into five categories.

1. ***Logic-based recognition Approaches*** [2, 3]**.** These approaches map the language

expression *T*(Text) and *H*(Hypothesis) to logical meaning representations Φ_T and Φ_H , then check if Φ_H can be inferred from Φ_T using many kinds of entailment rules and common sense knowledge *B* possibly by invoking theorem provers. But it is very difficult to convert the language expressions into logical forms concerning limited performance of current natural language processing tools.

2. ***Decoding based recognition approaches***[4,11]. Given many entailment rules like the following forms,

   X bought Y => X owned Y
   X loves Y => X likes Y
   X brought a lawsuit against Y => X sued Y

   These approaches are to search for a sequence of rule applications that turn *T* expression (or its syntactic or semantic representation) to *H* expression. If such a sequence is found, the two expressions constitute a positive textual entailment pair, depending on the rules used; otherwise, the pair is negative. Unfortunately, try to build these rules in a hierarchical way could not be easy no matter in theory or practice.
3. ***Transformation-based approaches***[23,24,25]. Transformation-based approaches use a set of rules to perturb the entailment pair with the goal of making the Text and Hypothesis identical. After the rule set has been exhausted (when either no more changes can be effected by apply rules, or some heuristic limited is reached), if the Text and Hypothesis match, the entailment pair is labeled as “entails”, and if they don’t, it is labeled as “not entailment”.
4. ***Alignment and similarity measures based recognition approaches***. These approaches are to measure a kind of similarity or distance between text *T* and hypothesis *H*, and then classify the pair into positive or negative example by comparing the similarity or distance with a threshold. The similarity measure used can be computed at different levels including surface lexical string [7, 22], syntactic tree structure [9, 10], or latent semantic representations [12, 13]. And quite a few successful approaches also treat RTE as an alignment problem [8, 21]. Thesaurus like WordNet, Hownet [20] can be useful in these approaches.
5. ***Machine learning based recognition approaches***[5, 6, 14, 15]. These approaches treat the entailment judgment problem between two texts *T* and *H* as a binary classification problem, then supervised machine learning methods can be used to make the textual entailment decision. Each pair of input language expressions <*T*, *H*> is represented by a feature vector <f_1, …, f_m> containing the scores of different similarity measures applied to the pair, and possibly other features. A supervised machine learning algorithm train a classifier on manually classified vectors corresponding to training input pairs. Once trained, the classifier can classify unseen pairs as correct or incorrect textual entailment pairs by examining their features.

This paper explores the methods for recognizing Chinese textual entailment. The difficult problems for this task include the lack of Chinese textual entailment rules, and word segmentation as well as other language processing errors, and it is also very hard to convert the language expressions T and H into logical forms to infer H from T. Therefore, we use a machine learning based Chinese textual entailment recognization method in which a new syntactic tree clipping and matching feature we presented is combined with other traditional different similarity features. With clipping and transforming the original syntactic tree structure into the “minimum information tree”, the matching between T and H can be more accuracy and tolerant of word segmentation error. The experimental result shows that the clipping on syntactic tree structure is effective for Chinese textual entailment recognization.

The remainder of this paper is composed as follows. In section 2 we introduce the rules to clip a syntactic tree structure to a “minimum information tree” and the similarity measure between different “minimum information tree”. In section 3 we present other features and machine learning methods used in our system. In section 4 we show the experimental results on the test data and give some analysis. Finally, we summarize our work and outline some ideas for future research.

2 Approach

Our approach also treat recognizing Chinese textual entailment as a binary classification problem. We believe that a Hypothesis *H* with “similar” content to the Text *T* is more likely to be entailed by that Text *T* than one with “less similar” content, therefore using matching similarity between *T* and *H* should be an important feature for entailment classification. In this paper we match *T* and *H* at different levels including lexical level, syntactic level, and shallow semantic level. At syntactic level, firstly we clip and transform the two original syntactic trees of *T* and *H* into “minimum syntactic trees”, then search for their common structure and compute matching similarity.

2.1 Clipping syntactic tree

The main idea of syntactic tree clipping is to delete meaningless nodes by aggregating those nodes of syntactic tree. Based on syntactic tree, the first operation is to aggregate the common subsequence into one node. Secondly, aggregate those strings which can be treated as “common similar subtrees”. Finally, we will get a tree with minimum information by saving related links of notes and deleting redundant information (nodes without any operation).

2.1.1 Common subsequence aggregation

In this step, we aggregate all common nodes by searching all subsequences. After this step, some entities can be extracted to reduce the Chinese word segmentation errors and the syntactic tree will be less complex. The following example (Marked as Example 1)is taken from NTCIR-10’s data:

T: 张艺谋执导的新作《十面埋伏》上映 4 天票房已突破 6300 万元人民币，超过同期《英雄》的票房记录.

H:《十面埋伏》上映 4 天票房突破 6300 万元人民币.

Two syntactic trees of *T* and *H* in the example is as following(ignore all punctuations):

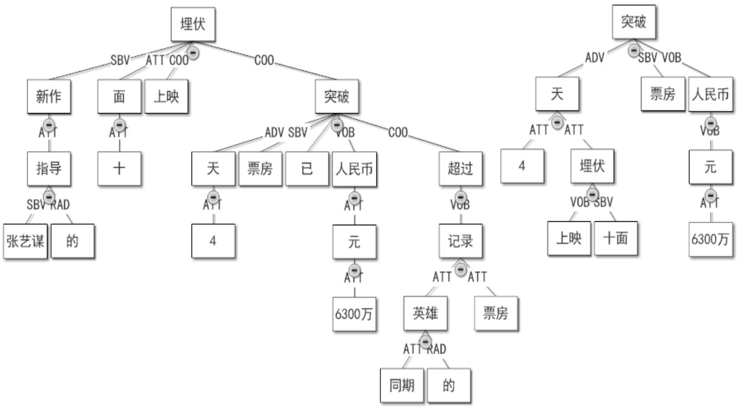


Fig. 1. Syntactic trees of T and H in Example 1

Common subsequence can fix the errors as Example 1 shows. In this example, "十面埋伏" should not be separated into three nodes, and we need to treat them as an entirety. But even two equal single nodes couldn't be treated as common subsequence. After aggregating the nodes, the "minimum information trees" would be as follows:

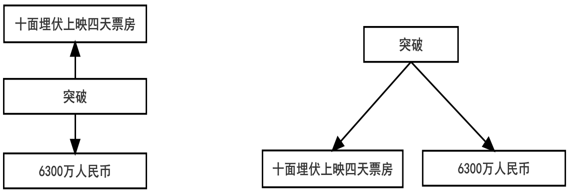


Fig. 2. "Minimum information trees" in Example

2.1.2 Common similar subtree aggregation

We define the similar subtrees as such kinds of trees that have similar format and generated from syntactic analysis. Those similar subtrees will be aggregated during the step of syntactic tree clipping. Our approach judges those similar subtrees by single word's overlap. With constraint in syntactic structure and core words' similarity we can determine whether these subsequences should be aggregated or not. Another example (Marked as Example 2) selected from NTCIR-10:

***T**:* 二次世界大战时日本广岛遭投原子弹

***H**:* 广岛在二次世界大战时遭原子弹轰炸

Two syntactic trees of *T* and *H* in this example are as following(ignore all punctuations):

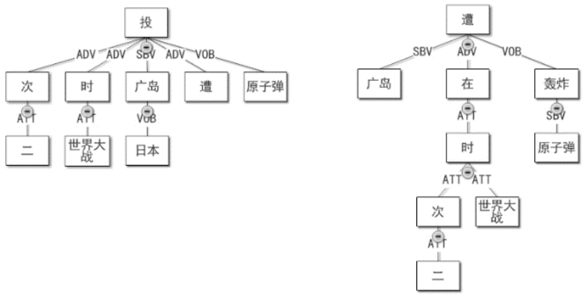


Fig. 3. Syntactic trees of T and H in Example 2

Here “二次世界大战时” is a common subsequence, “广岛” and “日本” would be name entities appearing in each text. The left one with “投” and “遭” and “原子弹” consist a subtree try to compare with the subtree which consists of “遭” and “轰炸” and “原子弹”. The score of similarity greater than threshold after calculation, then aggregate those nodes into one. The final “Minimum information trees” are:

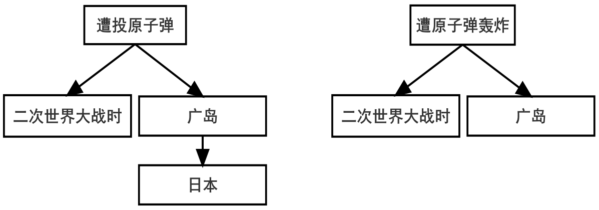


Fig. 4. “Minimum information trees” in example 1

What we had mentioned before, the equation for calculating the similarity between different subtrees is defined as:

$$\text{Sim}_{\text{subtree}}= \boldsymbol{\alpha}(\text{core vocabulary similarity}) + \boldsymbol{\beta}(\text{syntactic structure similarity}) \quad (1)$$

Here $0\leq\boldsymbol{\alpha}, \boldsymbol{\beta}\leq1, \boldsymbol{\alpha}+\boldsymbol{\beta}=1$. Via manual work, when $\boldsymbol{\alpha}=0.55$ and $\boldsymbol{\beta}=0.45$ this similarity work best in distinguishing the synonymy between two subtrees.

2.2 The syntactic tree clipping algorithm

The algorithm to clip the original syntactic trees of *T* and *H* into minimum information trees is as follows.

Table 1. Syntactic Tree Clipping algorithm

Input: Syntactic tree T_1 and T_2 with nodes $\{v_1, v_2,\ldots,v_n\}$ and $\{v_1', v_2',\ldots,v_n'\}$,

Output: Minimum information tree I_1, I_2 generated from T_1, T_2, which remain the entailment information contained in T_1, T_2. They also have deleted nodes that carry useless information about recognizing textual entailment.

Step1. Let D_1 and D_2 be the node sets to handle, D_i=∅. Then apply KMP algorithm to find all common subsequences in T_1, T_2, and put them into D_i (i =1,2) as independent subtree d_{ij}.

Step2. Use the smaller one as beginning, find all common similar subtrees in T_1, T_2. The way to find all subtrees need two steps. First, use the degree of word overlap to find some pairs of subtrees, 0.76 will be the threshold to the first screen. Then those pairs through discriminant function $Sim_{subtree}$ will select the similar subtree. Let them join the set of D_i as the original way d_{ik}, until they traverse all tree. We won't deal the nodes already done in step1.

Step3. Transforming T_i, take nodes in D_i, aggregate them which in the same subtree. New nodes location depends on parent node of those aggregated node, after this delete it from D_i. Keep the syntactic structure between these nodes, the path to root and all nodes in this path， until D_i=∅. If two subtrees have the same type of named entity, we still retain those nodes even if they are not the same one.

Step4. Delete nodes in T_i that we do nothing about them, then Minimum information tree I_i presents.

2.3 The similarity computing between minimum information trees

Although using clipping method will decrease the number of nodes, we still can't just employ statistical features to make the entailment prediction correctly. The minimum information trees still keep semantic features. Therefore, the following Equation (2) for the similarity calculation is necessary:

$$Sim_{Tree} = (\frac{1}{3})^{SBV}(\frac{1}{3})^{NED}\frac{\sum Max\{Sim(Node_T, Node_H)\}}{Min(Node_T, Node_H)} \tag{2}$$

In the equation, SBV stands for the syntactic subject-predicate dependency relation while only considering those nodes around root one:

$$SBV=\begin{cases}0, & if\ the\ relation\ between\ nodes\ is\ different\\ 1, & else\end{cases} \tag{3}$$

NED stands for the result of named entity discrimination:

$$NED=\begin{cases}1, & if\ two\ texts\ contain\ the\ identical\ named\ entity\ or\ named\ entity\ type\\ 0, & others\end{cases} \tag{4}$$

3 Traditional features

The statistical machine learning models are trained to classify whether *T* in a given text-hypothesis pair entail *H*, and some traditional features including statistical and lexical semantic features are also used in these models. The following table 2 and table 3 illustrate these features and computational equations for them.

Table 2. Statistical features

Feature Name	Comment	Formula
Word overlap	The overlap of word between two texts	$E_1=\|T \wedge H\|/\|H\|$ $E_2=\|T \wedge H\|/\|T\|$ $E=(2*E_1*E_2)/(E_1+E_2)$
Length difference	Using text length to distinguish entailment direction	$Lt(T,H)=\| Len(T)-Len(H)\|$
Cosine similarity	Representing the text pair as vectors, then calculating their cosine similarity	$Sim_{cos}(T,H)=\frac{\sum_{i=1}^{n} t_i * h_i}{\sqrt{\sum_{i=1}^{n} t_i} * \sqrt{\sum_{i=1}^{n} h_i}}$ *n* is vector dimensions

Table 3. Lexical semantic features

Feature Name	Comment	Formula
HowNet semantic similarity	Using HowNet to calculate the similarity between different words	See Equation 5
Tongyicilin semantic similarity [16]	Using Tongyicilin to calculate the similarity between different words	See Equation 5
The number of antonyms	Using the Web resource to count the number of antonyms	None
The number of negative words	Combining the number of antonyms to assist the decision	None
The overlap of named entity	Named entities can show the text topics in a way	$T_{NE}=\|T \wedge H\|/\|H\|$ $H_{NE}=\|T \wedge H\|/\|T\|$ $L_{NE}=(2*T_{NE}*H_{NE})/(T_{NE}+H_{NE})$

$$Sim=\frac{1}{2}\left[\frac{\sum_{i=1}^{m} max\{sim_w(w_{1i},w_{2j})|1 \leq j \leq n\}+}{m}+\frac{\sum_{j=1}^{n} max\{sim_w(w_{1i},w_{2j})|1 \leq i \leq m\}}{n}\right] \tag{5}$$

What need to be illustrated is that although the equations of HowNet and Tongyicilin similarity are same, the same formula is only to sum the similarity which has already been calculated. The value of similarity in different features calculates in different ways.

4 Experimental result and analysis

4.1 Data and evaluation standards

The National Institute of Informatics (NII) of Japan organized the NTCIR [19] RITE (Recognizing Inference in TExt) competition [17] since 2011. RITE is to evaluate one system's ability about recognizing specific entailment relationship between two sentences. Therefore, we use NTCIR-10 RITE evaluation dataset as our experimental data, in which 814 text-hypothesis sentence pairs will employed as the training set, and other 781 pairs be the test set. The semantic relationship between every sentence pair has been already labeled as entailment or not entailment. We also use Precision, Recall and F-measure as system performance evaluation criterion. They are defined as follows.

$$Pre = \frac{\#\,right\ decisions}{\#\,all\ decisions} \tag{6}$$

$$Re = \frac{\#\,right\ decisions}{\#\,all\ \ pairs\ in\ one\ relation} \tag{7}$$

$$F\text{ - }measure = \frac{2 * Pre * Re}{Pre + Re} \tag{8}$$

4.2 Experimental result and analysis

We implemented two systems NLPWM-01 and NLPWM-02 for experimental evaluation. NLPWM-01 uses all features mentioned in section 3, and NLPWM-02 added further the minimum information tree similarity feature of text pair. We define those *T-H* test text pairs in which the entailment relationship exists as positive instances, and others as negative instances. Being similar to confusion matrix, Table 4 presents the prediction numbers of two systems for positive and negative test instances respectively.

Table 4. Experimental result

System	#correct predication for positive pairs	#incorrect prediction for positive pairs	#correct prediction for negative pairs	#incorrect prediction for negative pairs
NLPWM-01	371	51	171	188
NLPWM-02	393	29	**202**	157

From Table 4 we can see that the precision increases after adding the minimum information tree feature, especially for negative test text pairs. The reason for performance improvement can be explained as: 1) After clipping the syntactic tree, the noise in recognizing process decreased; 2) Equation 2 solved some problems such as the only difference in subject or object, reverse of subject and object, and so on.

Figure 5 shows the achieved F-measure values by different models including Decision Tree, SVM and Na ïve Bayes based on Gaussian distribution machine learning methods [18]. Different machine learning methods cause different results. SVM which is regarded as the best supervised learning method in general text classification didn't obtain the best result. Meanwhile, na ïve Bayes approach gets the highest score in both prediction. And apparently, when adding the minimum information tree feature, the performance is always improved under these different prediction models.

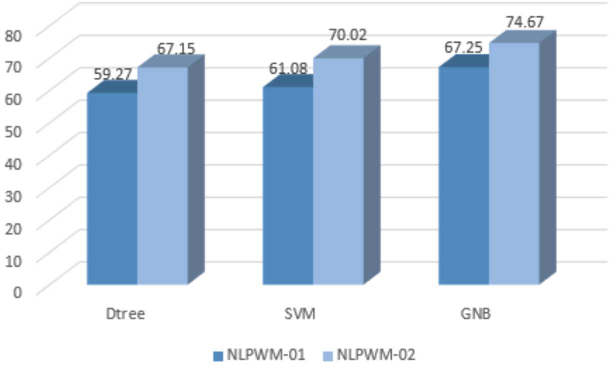
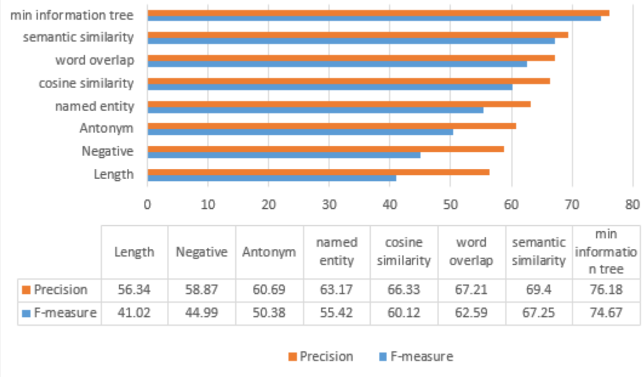


Fig. 5. Different machine learning methods achieve different F-measures

Figure 6 demonstrates the effect of different features in Na ïve Bayes decision model. Features are put into vector in an order from bottom to the top. The more features join the vector, the better accuracy would be.



	Length	Negative	Antonym	named entity	cosine similarity	word overlap	semantic similarity	min informatio n tree
■ Precision	56.34	58.87	60.69	63.17	66.33	67.21	69.4	76.18
■ F-measure	41.02	44.99	50.38	55.42	60.12	62.59	67.25	74.67

Fig. 6. Each feature effect the Na ïve Bayes decision

For all features as showed in figure 6, minimum information tree feature makes the biggest change in F-measure except the first change. These results show the minimum information tree feature is effective in textual entailment classification.

Table 5 compare the performance of our systems with those participated NTCIR-10 challenge.

Table 5. NTCIR-10 RITE Result (without percent)

System	MacroF1	Acc.	Y-F1	Y-Prec.	Y-Rec.	N-F1	N-Prec.	N-Rec
bcNLP-CS-BC-03	**73.84**	74.65	78.43	72.58	85.31	69.25	78.25	62.12
MIG-CS-BC-02	68.09	68.50	71.72	69.64	73.93	64.45	66.97	62.12
CYUT-CS-BC-03	**67.86**	68.12	70.74	70.16	71.33	64.98	65.63	64.35
bcNLP-CS-BC-01	67.04	69.65	76.32	65.98	90.52	57.75	80.20	45.13
bcNLP-CS-BC-02	66.89	69.91	76.89	65.71	92.65	56.88	83.33	43.18
MIG-CS-BC-01	65.71	65.81	67.56	69.33	65.88	63.87	62.11	65.74
CYUT-CS-BC-02	63.11	63.12	62.50	69.36	65.88	63.87	62.11	65.74
WHUTE-CS-BC-02	61.65	66.58	75.40	62.60	94.79	47.90	84.51	33.43
CYUT-CS-BC-01	61.17	61.59	57.14	71.94	47.39	65.20	55.86	78.27
IASL-CS-BC-02	60.45	63.25	70.98	61.90	83.18	49.91	66.82	39.83
NLPWM-01	**67.25**	69.40	75.64	66.37	87.91	58.86	77.03	47.63
NLPWM-02	**74.67**	**76.18**	80.86	71.45	93.13	68.48	87.45	**56.27**

The performance results of top-10 systems are listed in Table 5. In these performance criterions, MacroF1 is the average value of Y-F1 and N-F1. Items starting with "Y" show the prediction performances of different systems for positive instances (i.e., text pairs being entailment relationship), the other items are for negative instances.

Compared with bcNLP-CS-BC-03, NLPWM-02 should increase the value of recall in not entailment pairs. And 56.27% shows the weakness our system copes with not entailment pairs. This points our research direction and inspires us to find more reasonable expression on semantic features.

5 Conclusion

This paper proposed a machine learning based method for Chinese textual entailment recognization task. This method integrated lexical, syntactic, and shallow semantic levels of language matching features together. To construct syntactic matching feature, a syntactic tree clipping algorithm is presented to form minimum information trees of *T* and *H* for matching. The experimental result shows the minimum information tree feature is effective.

The approach still has two deficiencies: First, the time complexity in clipping algorithm isn't good enough; Second, the lack of resource make the system weak in dealing with not entailment pairs

Our future work is two-fold. The judgment on not entailment pairs is not successful, we therefore should find new feature or new method to express entailment relationship better. Meanwhile we also need to focus on the multi-direction in Chinese textual entailment recognizing, and this challenge requires entailment system developed in a more robust and reasonable way.

Acknowledgments

This work was sponsored by the National Natural Science Foundation of China (No. 61163039, No. 61163036, No. 61363058) and the Young Teacher Research Ability Enhancement Project of Northwest Normal University of China (NWNU-LKQN-10-2). We thank the anonymous reviewers for their insightful comments.

Reference

1. Dagan I. and Glickman O. Probabilistic textual entailment: generic applied modeling of language variability[C]//In PASCAL Workshop on Learning Methods for Text Understanding and Mining, Grenoble, France. (2004).
2. Tatu M.& Moldovan D. COGEX at RTE 3[C]//.In Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing ,Prague,Czech Republic,pp.22–27. (2007).
3. Bos J. Is there a place for logic in recognizing textual entailment?[J]. Linguistic Issues in Language Technology, 2013, 9.
4. Harmeling S. Inferring textual entailment with a probabilistically sound calculus[J]. Nat.Lang. Engineering, (2009). 15(4), 459–477.
5. Quinonero Candela, J., et al. "Machine Learning Challenges: evaluating predictive uncertainty, visual object classification and recognising textual entailment}." Proceedings of the First Pascal Machine Learning Challenges Workshop on Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment (MLCW 2005)}. Eds. L. De Raedt, et al. Vol. 6. Springer}, 2012.
6. Pham, Quang Nhat Minh, Le Minh Nguyen, and Akira Shimazu. "A machine learning based textual entailment recognition system of jaist team for ntcir9 rite." Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR'11). 2011.
7. Malakasiotis P, Androutsopoulos I. Learning textual entailment using SVMs and string similarity measures[C]//Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Association for Computational Linguistics, 2007: 42-47.
8. Xiao - Lin Wang, Hai Zhao, and Bao - Liang Lu, BCMI - NLP Labeled - Alignment - Based Entailment System for NTCIR - 10 RITE - 2 Task[C]// /Proceeding of the 10th NTCIR Conference, Tokyo, Japan. 2013.
9. Kouylekov M, Magnini B. Recognizing textual entailment with tree edit distance algorithms[C]//Proceedings of the First Challenge Workshop Recognising Textual Entailment. 2005: 17-20.
10. Maytham Alabbas, Allan Ramsay, Natural Language Inference for Arabic Using Extended Tree Edit Distance with Subtrees[J],Journal of Artificial Intelligence Research, 2013,volume 48, pages 1-22,

11. Bar-Haim R, Berant J, Dagan I. A compact forest for scalable inference over entailment and paraphrase rules[C]//Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. Association for Computational Linguistics, 2009: 1056-1065.
12. Xiaofeng Wu, Chengqing Zong， An Approach to News Paraphrase Recognition Based on SRL [J].Journal of Chinese Information Processing, 2010,24(5),3-9
13. Burchardt A.Ph.D. Dissertation. Modeling Textual Entailment with Role-Semantic Information. (2008)
14. Burrows S, Potthast M, Stein B. Paraphrase acquisition via crowdsourcing and machine learning[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2013, 4(3): 43.
15. Galitsky B. Machine learning of syntactic parse trees for search and classification of text[J]. Engineering Applications of Artificial Intelligence, 2013, 26(3): 1072-1091.
16. Tian jiu-le, Zhao Wei, Words Similarity Algorithm Based on Tongyici CiLin in Semantic Web Adaptive Learning System [J].Journal of Jilin University， 2010,28(6),602-608
17. Liu Maofu, Li yan, Ji Donghong, Event Semantic Feature Based Chinese Textual Entailment Recognition [J]. Journal of Chinese Information Processing， 2013,27(5),129-136
18. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011
19. Watanabe Y, Miyao Y, Mizuno J, et al. Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10[C]//Proceedings of the 10th NTCIR Conference. 2013.
20. Dong, Zhendong, and Qiang Dong. HowNet and the Computation of Meaning. Singapore: World Scientific, 2006.
21. Turchi, Marco, and Matteo Negri. "ALTN: Word Alignment Features for Cross-Lingual Textual Entailment." *Atlanta, Georgia, USA* (2013): 128.
22. Graham, Yvette, Bahar Salehi, and Timothy Baldwin. "Umelb: Cross-lingual Textual Entailment with Word Alignment and String Similarity Features."Atlanta, Georgia, USA (2013): 133.
23. Asher Stern and Ido Dagan. A confidence model for syntactically-motivated entailment proofs. In Proceedings of the International Conference on Recent Advances in Natural Language Processing, 2011. 96
24. Rodrigo de Salvo Braz, Roxana Girju, Vasin Punyakanok, Dan Roth, and Mark Sammons. An inference model for semantic entailment in natural language. In Proceedings of the National Conference on Artificial Intelligence (AAAI), pages 1678–1679, 2005. DOI: 10.1007/11736790_15. 41, 90, 93, 162
25. Ido Dagan, Dan Roth, Mark Sammons, Fabio Massimo Zanzotto: Recognizing Textual Entailment: Models and Applications. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers 2013, ISBN 9781598298345, pp. 1-220