Recognizing Text Entailment via Syntactic Tree Matching

Zhewei Mai
Dept. of Computer & Info Science
Northeastern University
360 Huntington Ave
Boston, MA 02115, USA
zhewei@ccs.neu.edu

Yue Zhang
Dept. of Computer & Info Science
Northeastern University
360 Huntington Ave
Boston, MA 02115, USA
mnleod@ccs.neu.edu

Donghong Ji
School of Computer Science
Wuhan University
Luojia Hill
Wuhan, 430072, China
dhji@whu.edu.cn

ABSTRACT
In this paper, we present our approach for Chinese Binary-class (BC) subtask of Recognizing Inference in Text (RITE) task in NTCIR-9 [9]. Our system is to judge whether a given sentence can entail another or not. Firstly each sentence is parsed to a syntactic tree in which nodes represent words or phrase, and links represent syntactic relationships between nodes. Then, the entailment between two sentences is recognized by syntactic tree matching. In addition, to compute the similarity between two words or phrases, the external resources (Tongyici Cilin, HowNet[1], and Hudong Wiki[2]) are employed. The evaluation results show that our system can reach the accuracy of 87.1% in recognizing pairs with entailment relationship.

Categories and Subject Descriptors
1.2.7[**Artificial Intelligence**]: Natural Language Processing – Text Analysis

General Terms
Algorithms, Experimentation

Keywords
Tree Matching, Text Entailment, Semantic Matching.

1. INTRODUCTION
In present, the lack of accurate and robust semantic inference method blocks the development of Natural Language Processing. On the other hand, a persistent and robust one can bring a positive effect. For instance, for a question answering system, it will bring obvious improvement to recognize whether an answer can entail the query. For document summarization, the generated succinct sentences will express the same content as the original document. For an information extraction system, semantic inference method can help to recognize various equivalent linguistic expressions.

Recognizing Textual entailment is a task about how to capture a semantic inference. In detail, it's to recognize whether the information expressed in a textual hypothesis can be inferred from the information expressed in another text. There are already several popular approaches in this field. Andrew Hickl, et al. acquired linguistic information from the hypothesis-text pair, and casted the inference recognition as a classification problem [1]. Some researchers successfully tackled the inference by logic-based abductive approaches [2, 3]. Aria Haghighi, et al. utilized learned graph matching method to get good result in Recognizing Textual Entailment (RTE) Challenge [4].

Our idea is derived from syntactic tree matching. Intuitively, the similarity between two sentences is not only based on their edit distance, but also related to their syntactic structures. In our system, every sentence is firstly parsed to syntactic tree through the Stanford Parser [3](an example of a syntactic tree is shown in Figure 1). Then, the tree matching algorithm is applied to calculate the similarity between two syntactic trees. Besides, we consider not only lexical but also syntactic and semantic features into the matching model.

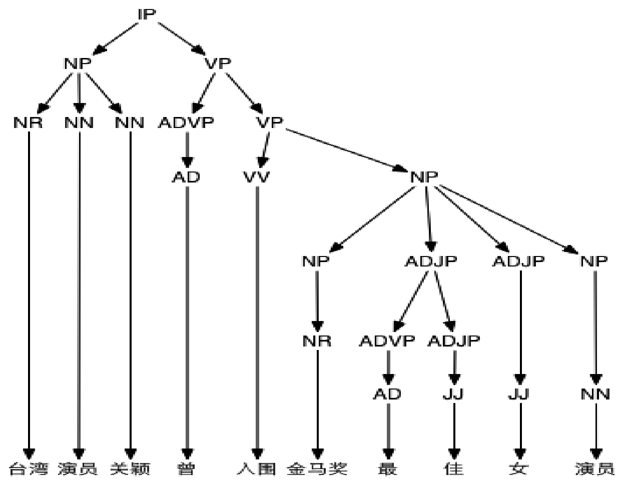

Figure 1. The syntactic tree for "台湾演员关颖曾入围金马奖最佳女配角" (Guanying, a Taiwan actress, has been nominated as the best supporting actress of Golden Horse Awards)

The remainder of the paper is organized as follows: Section 2 describes the linguistic processing for every hypothesis-text pair. Section 3 presents the algorithm about syntactic tree matching. Section 4 discusses the evaluation results. Finally, a conclusion is drawn on Section 5 with adjustment suggested for future works.

2. Linguistic Processing
The first component in our system is Linguistic processing, which includes two parts: Chinese segmentation and combination of phrase which have lexical, syntactic, and semantic connection. It's the basic component in Chinese Natural Language Processing. The qualities of words or phrases after segmentation directly influence the effectiveness of the whole system.

1 http://www.keenage.com/html/e_index.html

2 http://www.hudong.com/

3 http://nlp.stanford.edu/software/lex-parser.shtml

4 http://nlp.stanford.edu/software/segmenter.shtml

2.1 Chinese Word Segmentation

Chinese word segmentation plays a significant part in our system. An effective segmenter can have a direct and considerable influence on building syntactic trees in next step.

During the investigation, we find there are a variety of segmentation tools in the field of Chinese Natural Language Processing. Almost every research group, like Stanford, Harbin Institute of Technology, and Fudan University, has its own segmentation tool. There is no documented standard discussing which segmenter is better than the others. The reason is the performances of Chinese segmenters have little difference. In our system, we choose Stanford Chinese Word Segmenter[4], because it has an advantage of utilizing lexicon features, and produce more consistent segmentation with external lexicon features.

Despite the obvious advantage of Stanford Chinese, the segmentation granularity is too little in some specific situations. For instance, using Segmentation, "一月" (January) is segmented to "1/CD, 月/NN" instead of "一月/NT". This kind of wrong segmentations will cause inconsistencies in context and even bring bad effect on later steps. To avoid this problem, optimizations are necessarily required in words or phrases to improve the performance in a certain extent.

Our approach to solving inconsistent segmentation in those words or phrases, especially numeral phrases, is to construct patterns to replace those words before segmentation. First, various kinds of numeral words are recognized and replaced based on constructed patterns, and then converted into the given format. After that, numeral phrases would be segmented without ambiguity and serve as a highly correct numeral matching metric in the matching process. However, the normalization only functions in dealing with the number phrases, and can hardly be applied in other words.

2.2 Combination

After segmentation, some phrases, such as "500 米" (500 meters) and "下午 2:45" (2:45 P.M.) are respectively segmented into "500/CD 米/M" and "下午/NT 2:45/CD". Intuitively, we think the combination of semantic-related words can provide more detailed description for the sentence.

Therefore, we combined words which have semantic relation by pattern match. We manually construct patterns. For example, a numeral word is followed by measure word, and a period is in front of a specific time. This combination and assemble powerfully abbreviate the size of the syntactic tree and prevent the unnecessary comparisons, which reduce the performance of syntactic tree matching algorithm.

3. Syntactic Tree Matching

In this component, we want to calculate the probability of the entailment existed in two syntactic trees. We employ the tree matching method from [5]. Firstly utilize a syntax parser to build a syntactic tree for every question, and retrieved answers were ranked based on the similarity to the syntactic tree of the query question. It performs well in QA system. The tree matching algorithm is as follows:

$$M(r_1,r_2)=\begin{cases}0, & if\ r_1\neq r_2\\ \delta_{r1}\delta_{r2}\lambda^{S_1+S_2}\mu^{D_1+D_2}, & if\ r_1,r_2\ are\ leaves\\ w\prod_{j=1}^{nc(r_1)}M\big(ch(n_1,j),ch(n_2,j)\big), & otherwise\end{cases} \quad (1)$$

Where r_1 and r_2 are two nodes which are matched, δ_i denotes the weight of node i in the parsing tree, λ and μ are two tuning parameter denoting the preference between size and depth, S_i is the number of nodes that tree fragment i contains, D_i is the level of the tree fragment root in the entire syntactic parsing tree, w is $\delta_{r1}^{\eta}\delta_{r2}^{\eta}\lambda^{2\eta}\mu^{\eta\left(2-(1+nc(r1))(D_{r1}+D_{r2})\right)}$. $nc(n)$ is the total number of children of the node n. $ch(n,j)$ is the *j-th* child of node n in the tree.

In our system, we initially set that:

- $\delta_i=1.2$, when node i is VV, NR, or NT.
- $\delta_i=1.1$, when node i is either VE or NN,
- $\delta_i=0$, when node i have no importance for a sentence, include: DEC, DEG, DER, DEV, AS, SP, ETC, MSP, PU, ON, IJ, or VC.
- $\delta_i=1$, for all other types of nodes.

3.1 Similarity Metrics

For getting the similarity score between two syntactic trees T_1 and T_2, firstly the algorithm traverse two trees in post-order, and calculate the pair-vise node matching scores between the nodes in these two trees. This results in a $|T_1|$x$|T_2|$ matrix of $M(r_1,r_2)$. Then sum up all scores in the matrix to represent the similarity score between two parsing trees. Finally, normalize the score. The final equation is below:

$$\text{Score}_{\text{Final}}=SIM(T_1,T_2)/\sqrt{SIM(T_1,T_1)*SIM(T_2,T_2)} \quad (2)$$

$$SIM(T_1,T_2)=\sum_{r_1\in T_1}\sum_{r_2\in T_2}M(r_1,r_2) \quad (3)$$

3.2 Word and Phrase Match

How to measure the similarity or dissimilarity between two words or phrases is one of the most crucial parts for textual entailment recognition system. In order to improve the accuracy, we apply multiple external sources to implement it.

We have the matches in several ways. They are exact match, prefix match, antonymic match, date/time match, Tongyici Cilin similarity, HowNet similarity, and Hudong Similarity.

3.2.1 Exact Match

As the words "Exact Match" suggesting, the two given words or phrases are identical to each other.

3.2.2 Prefix Match

Though there is no such concept of prefix in Chinese, we regard a group of phrases that contains the same attribution in first few words as prefix in prefix matching. For example, the two geological-related phrases "武汉市" (the city of Wuhan) and "武汉人" (the residents of Wuhan), which in most situations, has the positive similar relationship.

3.2.3 Antonymic Match

It's indispensable part in our system, since, in the textual entailment recognition, the appearance of a pair of antonymous words or phrases may cause the entire dissimilarity between two sentences. However, it's very difficult to recognize whether one word or phrase is antonymic to another, mainly because there is not a standard antonymous dictionary in Chinese.

Within our system, antonymous relationship between words or phrases is determined by a pre-computed list of about 35, 000 antonymous pairs. However, the limited word list has a very small quantity.

3.2.4 Date/Time Match

As it has been mentioned in 2.1, by pre-defined pattern the date and time phrase can be recognized and then normalized. For example, “2011 年春节” (The Spring Festival in 2011) can be normalized to a specific date “2011/02/03”; “万历十五年” (a Chinese era, Wanli Fifteenth Year) to “1587”.

3.2.5 *Alias Match*

With Hudong Wikipedia, the world’s largest Chinese encyclopedias, aliases for given words can be effective identified. We can utilize the functionality of redirection in Hudong to find aliases for a given thing. For instance, when we type in “毛主席” (Chairman Mao), it will redirect to the web page of “毛泽东” (Zedong Mao). Therefore, what we do is to compare the two webpages for two words/phrases, if they are the same page, then each one is the alias to the other. In this way, the problem how to recognize whether one word/phrase is the alias to another can be efficiently tackled.

3.2.6 *Tongyici Cilin Similarity*

Tongyici Cilin is a thesaurus of synonyms and antonyms in Chinese. We employ Tian’s algorithm [6] to compute the similarity of words or phrases. In order to improve the matching accuracy, we set a confidence level of 0.80 on semantic distances between two words or phrases. Therefore, two words/phrases can be matched only when their semantic score passes the confidence level.

3.2.7 *HowNet Similarity*

HowNet is a common-sense knowledge base unveiling inter-conceptual relations and inter-attribute relations of Chinese words or phrases. We use two different algorithms in similarity computation in HowNet [7, 8] to prevent the unreasonable score from comparison of independent words

4. Evaluation

In the BC (Binary-class) subtask, for every text pair (t_1, t_2), we want to identify whether t_1 entails a hypothesis t_2 or not. For evaluating a system, the meeting uses accuracy of labels predicted. In formal run, we submitted three runs to NTCIR-9: RITE1-NSNG-CS-BC-01, RITE1-NSNG-CS-BC-02, RITE1-NSNG-CS-BC-03. RITE1-NSNG-CS-BC-01 is the result of the original algorithm described before. In RITE1-NSNG-CS-BC-02, we increase the weight of important nodes. For VV, NR, or NT, $\delta_i = 2.2$, and for VE or NN, $\delta_i = 2.1$. In RITE1-NSNG-CS-BC-03, we increase the weight of matched parent nodes.

Table 1. Results of the formal run

Run ID	Accuracy	Y_Accuracy	N_Accuracy
RITE1-NSNG-CS-BC-01	0.654	0.863	0.270
RITE1-NSNG-CS-BC-02	0.668	0.871	0.299
RITE1-NSNG-CS-BC-03	0.590	0.711	0.368

Table 1 lists the result of our 3 formal runs. In formal run, there are 407 text-hypothesis pairs which include 263 pairs which have entailment relationship and 144 without entailment. Column Y_Accuracy represents the accuracy of recognizing pairs with entailment relationship, and column N_Accuracy represents the accuracy of recognizing pairs without entailment relationship.

Table 1 shows that our system has a good performance on recognizing pairs with entailment relationship. For example, although the edit distance between text and hypothesis in Pair 22 (shown in Figure 2) is big, our system still can recognize it correctly. In addition, our system is robust in recognizing entailment in different syntactic structures. The system can correctly recognize the entailment in pairs like Pair 54 in Figure 2.

In spite of matching accuracy, unmatched node pairs are not given enough penalties. For instance, in Pair 24, shown in Figure 2, the edit distance between two sentences is very small, however the time difference between two pair, “一九九六” (1996) and “1998”, leads to the independence between the pair. In our system few penalties are given to this type of pairs, which as a consequence, results in incorrect recognition.

<pair id="22" label="Y">

<t1>

关颖新书自爆与蒋家第四代的蒋友柏情史，还指这是她今生唯一最接近“门当户对”的恋情

</t1>

<t2>

关颖传出与蒋家第四代蒋友柏的恋情

</t2>

</pair>

<pair id="24" label="Y">

<t1>

何大一在一九九六年被美国《时代》杂志评为年度风云人物

</t1>

<t2>

何大一在 1998 年首次被美国《时代》杂志评为年度风云人物

</t2>

</pair>

<pair id="54" label="Y">

<t1>阿拉法特于 2004 年 11 月 11 日过世</t1>

<t2>2004 年 11 月 11 日 ,阿拉法特去世</t2>

</pair>

Figure 2. Examples of the test data

Moreover, RITE1-NSNG-CS-BC-02 has a better result than RITE1-NSNG-CS-BC-01. It fully illustrates that increasing the weight of more important nodes can decrease the impact of less important nodes.

In RITE1-NSNG-CS-BC-03, we gain the weight of the depth of the matching sub-tree. It suggests that the higher the depth of the matched sub-tree is, the larger the matching score we can get. It increases the accuracy to recognize the pair without entailment relationship, that’s because it is more sensitive to structures of trees. Therefore, it decreases the accuracy of recognizing pairs with entailment relationship.

5. Conclusion and Future

In this paper, we demonstrate how to use syntactic tree to recognize the entailment relationship between two sentences. Also, functional methods are introduced to accurately compare the similarity or dissimilarity between two words or phrases. The evaluation results show that our method has a good score in recognizing pairs with entailment. However, because unmatched node pairs cannot be given enough penalties, it causes the high error rate in recognizing pair without entailment relationship.

In the future, there are still many aspects which we need to enhance, such as antonym relationship identification, how to compare similarities between words/phrases which are not in the Tongyici Cilin and HowNet, and proper penalties on unmatched tree node pairs.

6. ACKNOWLEDGMENTS

Thanks to NTCIR and LDC for supporting experiment dataset. The authors are grateful to Ms. Feng Hui in preparing the manuscript.

7. REFERENCES

[1] Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing Textual Entailment with LCC's Groundhog System. In Proceedings of the Second PASCAL Challenges Workshop.

[2] Dan I. Moldovan, Christine Clark, Sanda M. Harabagiu, and Steven J. Maiorano. 2003. Cogex: A logic prover for question answering. In *HLT-NAACL*.

[3] Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of AAAI 2005*. AAAI Press.

[4] Aria D. Haghighi, Andrew Y. Ng, Christopher D. Manning. Robust textual inference via graph matching, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p.387-394, October 06-08, 2005, Vancouver, British Columbia, Canada.

[5] Kai Wang , Zhaoyan Ming , Tat-Seng Chua, A syntactic tree matching approach to finding similar questions in community-based QA services, Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, July 19-23, 2009, Boston, MA, USA.

[6] 田久乐. 2010. 基于同义词词林的词语相似度计算方法. 吉林大学学报: 信息科学版.

[7] Tian Xia, An Edit Distance Algorithm with Block Swap. (2008). An Edit Distance Algorithm with Block Swap (pp. 54–59). doi:10.1109/ICYCS.2008.14

[8] Qun Liu, Word similarity computing based on How-Net. (2002). Word similarity computing based on How-Net. Computational Linguistics and Chinese Language

[9] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda. Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In NTCIR-9 Proceedings, to appear, 2011.