International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC 2014)

A Hybrid Approach to Textual Entailment Recognition

Rongyue Mei
Beijing Institute of Technology
School of Computer Science
Beijing, China
meirongyue@bit.edu.cn

Xuejin Li
Beijing Institute of Technology
School of Computer Science
Beijing, China
xuejinli7@gmail.com

Hongping Fu
Beijing Institute of Technology
School of Computer Science
Beijing, China
fhongping@bit.edu.cn

Abstract—The task of textual entailment recognition is to determine whether a text entails a hypothesis. This paper proposes a hybrid technique to identify the entailment relation between texts and hypothesis. This technique includes an approach based on lexical similarities and an approach based on the classifier of support vector machine. The approach based on lexical similarities is to use the similarities between a set of words within a text and a set of words within a hypothesis. The approach based on the classifier means to treat this task as a classification problem. We propose two kinds of classification features which include features based on semantic roles, and ones based on dependency relations and WordNet. We use our hybrid technique to integrate the two sets of experimental results by the lexical similarities-based approach and the SVM classifier-based approach. The experimental results demonstrate that our technique is effective to solve the problem of textual entailment recognition.

Keywords-textual entailment; support vector machine; WordNet; dependency; semantic role labeling

I. INTRODUCTION

Recognizing textual entailment (RTE) has now become an important issue in the fields of natural language processing and text mining. The RTE task is to automatically identify the entailment relationship between a hypothesis and a text. Here, a hypothesis (H) is a piece of text, and a text (T) comprises a few sentences whose meaning may or may not entail the meaning of H. If the truth of H can be inferred from the evidence in T, then the entailment relationship between T and H is denoted as T→ H [1]. For example, the following text T entails the hypothesis H, i.e., T→H.

T: "*The wait time for a green card has risen from 21 months to 33 months in those same regions*."
H: "*It takes longer to get green card.*"

A lot of methods to deal with the textual entailment recognition problem have been proposed during recent years. Those approaches may use features of lexical, syntactic, or semantic levels of texts to represent texts and hypothesis. For instance, the works of Mehdad et al [2], Qiu et al[3], and Tsatsaronis et al.[4] extract syntactic and semantic characteristics of texts and hypothesis to determine the entailment relationship between a hypothesis and a text. Another works of Tsatsaronis [5, 6] utilize characters, lexical, syntactic and semantic features of texts and hypothesis to deal with the textual entailment recognition issue. Some textual entailment recognition systems [5,6,7,8] use classifiers whose include two class labels to distinguish whether a text entails a hypothesis. The textual entailment recognition methods can be used in many information processing applications such as question answering, information retrieval, information extraction, machine translation and automatic summarization and so on.

In this paper, a hybrid technique is proposed to identify the entailment relation between texts and hypothesis. This technique includes an approach based on lexical similarities and an approach based on the classifier of support vector machine. The approach based on lexical similarities is to use the similarities between a set of words within a text and a set of words within a hypothesis. The approach based on the classifier means to treat RTE task as a classification problem. We propose two kinds of classification features which are composed of features based on semantic roles, and ones based on dependency relations and WordNet. In addition, other lexical, syntactic and semantic features are also used in the approach based on the classifier. The experimental results demonstrate that our technique is effective to solve the problem of textual entailment recognition.

The rest of this paper is organized as follows. Section II presents the framework of our hybrid approach and section III describes lexical similarities-based entailment module. Section IV presents the SVM classifier-based entailment module. The experimental results are given in section V. The conclusions are drawn in section VI.

II. The Framework Of Our Approach

The framework of our textual entailment recognition approach is given in Fig .1. Our RTE approach includes the preprocessing module, the lexical similarities entailment module, the SVM classifier entailment module, and the integration module.

In the preprocessing module, texts and hypothesis terms are processed using Stanford CoreNLP [9] and ClearNLP [10] tools. Stanford CoreNLP is used for tokenization, stem, part of speech tagging, while ClearNLP is applied to identify semantic roles of sentences within texts and hypothesis. In the lexical similarities-based entailment module, five kinds of methods are used to compute the similarities between texts and hypothesis in order to determine the entailment relationships. In the SVM classifier-based entailment module, SVM with two kinds of class labels is applied to determine whether a text imply a hypothesis.

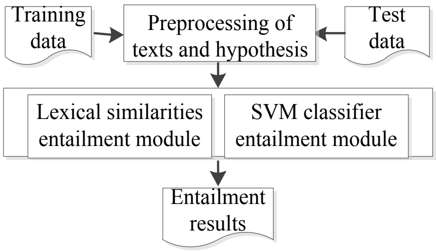
Training data
Preprocessing of texts and hypothesis
Test data
Lexical similarities entailment module
SVM classifier entailment module
Entailment results

Figure 1. The Framework of Our Hybrid Approach

III. Lexical Similarities-based Entailment Module

The architecture of the lexical similarities-based entailment module is given in Fig .2. In the following, we will present five kinds of methods of computing the similarities between texts and hypothesis.

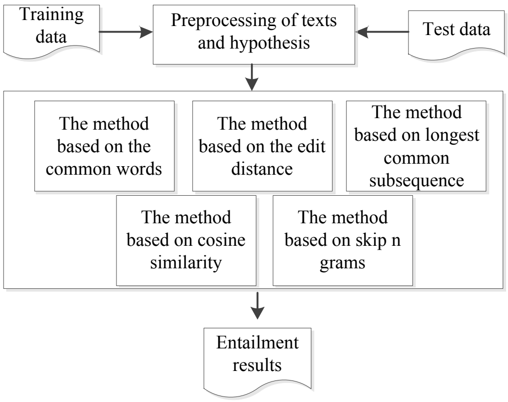
Training data
Preprocessing of texts and hypothesis
Test data
The method based on the common words
The method based on the edit distance
The method based on longest common subsequence
The method based on cosine similarity
The method based on skip n grams
Entailment results

Figure 2. Architecture of Lexical Similarities-based Entailment Module

The first method of computing similarities is founded on the number of common words within texts and hypothesis, as shown in Equation (1). Here, W_T is the set of words in a text, and W_H is the set of words in a hypothesis.

$$Similarity_O(T,H)=\frac{|W_T\,\mathbf{I}\ W_H|}{|W_T\,\mathbf{U}W_H|} \qquad (1)$$

The second method of measuring similarities is based on the longest common word subsequence of words within texts and hypothesis, as shown in Equation (2). Here, *length(LCS(T,H))* is the length of the longest common word subsequence *LCS(T,H)* of a text T and a hypothesis H, and *length(H)* is the length of H.

$$Similarity_L(T,H)=\frac{length(LCS(T,H))}{length(H)} \qquad (2)$$

The third method of calculating similarities is according to cosine similarities between a vector of the text and a vector of a hypothesis in an inner product space, as defined in Equation (3). Here, t_i and h_i are the elements within vectors of a text and a hypothesis, respectively. Dimensions of those vectors are the number of different words within the text and the hypothesis, while the elements of those vectors are the frequency of words in the text or the hypothesis.

$$Similarity_C(T,H)=\frac{\sum_{i=1}^{n}(t_i\times h_i)}{\sqrt{\sum_{i=1}^{n}t_i^2}\times\sqrt{\sum_{i=1}^{n}h_i^2}} \qquad (3)$$

The forth method of computing similarities is founded on the edit distance between a text and a hypothesis. It is the minimum number of operations needed to transform the hypothesis into the text, where an operation is an insertion, deletion, or substitution of a single word.

The last method of measuring similarities is based on the number of common skip-n-grams within a text and a hypothesis. A skip-n-gram is a subsequence of n words within sentences, where the words appear with arbitrary gaps, as shown in Equation (4). Here, *skip-gram(T,H)* is the number of common skip-grams extracted from the text and the hypothesis, and *m* is the number of skip-grams in the hypothesis. We use 2 and 3 gaps in our experiments.

$$Similarity_S(T,H)=\frac{skip-gram(T,H)}{m} \qquad (4)$$

In our lexical similarities-based entailment module, we use the five methods above to identify the entailment relation between texts and hypothesis. If one or more methods assign a pair of text and hypothesis as “YES”, we assign this pair as “YES”. Otherwise, we assign the pair as “NO”.

IV. SVM Classifier-based Entailment Module

The architecture of the SVM classifier-based entailment module is described in Fig .3. We use six lexical features, two syntactic features and five semantic features.

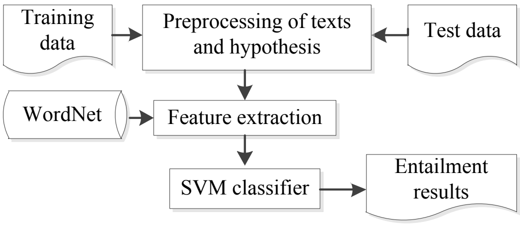

Figure 3. Architecture of the SVM Classifier-based Entailment Module

A. Syntactic Features

The syntactic features we used include the feature of the tree edit distance and the feature of dependency similarity. The first feature is the minimum number of operations needed to transform the dependency tree of a text into the dependency tree of a hypothesis (shown in (5)), where an operation is an insertion, deletion, or substitution of a tree node, and *costIns*, *costDel* and *costSub* represent the cost of insertions, deletions and substitutions.

$$TreeDistance\left(T,H\right)=costIns+costDel+costSub \quad (5)$$

The feature of dependency similarity is based on dependency relations of texts and hypothesis. First, we use Stanford Parser to identify the following syntactic components of sentences: subjects, objects, verbs, nouns, prepositions, numbers and determiners [9]. Subjects are extracted based on the dependency relations nsubj and nsubjpass while objects and determiners are acquired by the relation dobj and det. Verbs are extracted based on subjects and objects of sentences. Nouns, prepositions, and numbers are identified according to their respective part-of-speeches. For example, the parsing results of dependency relations of the sentence "Oracle released a confidential document" is as follows: nsubj(released-2, Oracle-1), root(ROOT-0, released-2), det(document-5, a-3), amod(document-5, confidential-4), dobj(released-2, document-5).

Secondly, we utilize the WordNet to extract the synonyms, hypernyms and hyponyms of the verbs which are included in the dependency relations. And then we compute the similarity between a text and a hypothesis on the aspect of dependency features, as shown in Equation (6).

$$\begin{aligned}Dependency_{Score}&=\alpha\times Sim_{sov}+\beta\times Sim_{num}+\gamma\times Sim_{nn}\\&+\delta\times Sim_{pre}+\varepsilon\times Sim_{det}\end{aligned} \quad (6)$$

Here, α, β, γ, δ and ε are parameters which are learned from the training data, and $\alpha+\beta+\gamma+\delta+\varepsilon=1$. (1) Sim_{sov} is the similarity between subjects, verbs and objects of a text and those of a hypothesis. It is assigned as 1 if subjects and verbs of the text are same as those of the hypothesis; otherwise, it is 0.5 if subjects or objects of the text are same as those of the hypothesis. (2) Sim_{num} is the similarity between numbers of a text and numbers of a hypothesis. If numbers of the text match to numbers of the hypothesis, it is assigned as 1; otherwise, it is 0. (3) Sim_{nn} is the similarity between nouns of a text and those of a hypothesis. It is assigned as 1 if nouns of the text are same as those of the hypothesis; otherwise, it is 0. (4) Sim_{pre} is the similarity between prepositional relations in a text and those in a hypothesis, while Sim_{det} is the similarity between the determiner of a text and determiners of a hypothesis. They are assigned as 1 if prepositional relations and determiners of the text match to those of the hypothesis; otherwise, they are 0.

B. Semantic Features

The semantic features that we utilize are extracted by using WordNet lexical ontology [11]. These features include Synonyms, Hypernyms-Hyponyms, Antonyms and Antonyms-Hyponyms.

The first semantic feature is based on the number of common synonyms of words within a text and a hypothesis, as shown in Equation (7), (8) and (9). Here, $t_i \in W_T$, W_T is a set of words within a text, and $h_i \in W_H$, W_{H} is a set of words within a hypothesis. n_t and n_h are the number of words in W_T and W_H, respectively. S_{t1} is a set of synonyms of words in W_T, and S_{h1} is a set of synonyms of words in W_H.

$$Synonyms(T,H)=\left|S_{t1}\cap S_{h1}\right| \quad (7)$$

$$s_{t1}=\bigcup_{i=1}^{n_t}synonyms(t_i) \quad (8)$$

$$s_{h1}=\bigcup_{i=1}^{n_h}synonyms(h_i) \quad (9)$$

The second semantic feature is designed to overcome the problem related to texts and hypothesis formulating concepts at different levels of conceptual abstraction. It calculates the number of common words between hypernyms of words within a text and hyponyms of words within a hypothesis, as shown in Equation (10), (11) and (12). Here, $t_i \in W_T$, W_T is a set of words within a text, and $h_i \in W_H$, W_{H} is a set of words within a hypothesis. n_t and n_h are the number of words in W_T and W_H, respectively. S_{t2} is a set of hyponyms of words in W_T, and S_{h2} is a set of hypernyms of words in W_H.

$$Hyponyms-Hypernyms(T,H)=\left|S_{t2}\cap S_{h2}\right| \quad (10)$$

$$s_{t2}=\bigcup_{i=1}^{n_t}hyponyms(t_i) \quad (11)$$

$$s_{h2}=\bigcup_{i=1}^{n_h}hypernyms(h_i) \quad (12)$$

The third semantic feature is based on the number of common words between antonyms of words within a text and the words within a hypothesis, as defined in Equation (13), (14) and (15). Here, $t_i \in W_T$, W_T is a set of words within a text, and $h_i \in W_H$, W_{H} is a set of words within a hypothesis. n_t and n_h are the number of words in W_T and W_H, respectively. S_{t3} is a set of antonyms of words in W_T, and S_{h3} is a set of words in W_H.

$$Antonyms\left(T,H\right)=\left|S_{t3}\cap S_{h3}\right| \quad (13)$$

$$s_{t3}=\bigcup_{i=1}^{n_t}antonyms(t_i) \quad (14)$$

$$s_{h3}=\bigcup_{i=1}^{n_h}(h_i) \quad (15)$$

The last semantic feature is according to the number of common words between antonyms of words within a text and hyponyms of words within a hypothesis, as shown in Equation (16), (17) and (18). Here, $t_i \in W_T$, W_T is a set of words within a text, and $h_i \in W_H$, W_{H} is a set of words within a hypothesis. n_t and n_h are the number of words in W_T and W_H, respectively. S_{t4} is a set of antonyms of words in W_T, and S_{h4} is a set of hyponyms of words in W_H.

$$Antonyms-Hyponyms=\left|S_{t4}\cap S_{h4}\right| \quad (16)$$

$$s_{t4}=\bigcup_{i=1}^{n_t}antonyms(t_i) \quad (17)$$

$$s_{h4}=\bigcup_{i=1}^{n_h}\mathrm{hyponyms}(h_i) \quad (18)$$

C. *Semantic Role Labeling*

The feature of semantic roles similarity is the similarity between texts and hypothesis on the aspect of semantic roles. Semantic roles are semantic arguments of a sentence which are associated with the predicates or verbs of a sentence. We use ClearNLP to label semantic roles of sentences. The semantic role similarity is calculates as Equation (19). Here, S_{t5} refers to semantic roles of a text while S_{h5} refers to semantic roles of a hypothesis.

$$Sim_{srl} = \left| S_{t5} \cap S_{h5} \right| \qquad (19)$$

We utilize six core semantic roles A0, A1, A2, A3, A4 and A_5, where A0 indicates an agent of the action of a predicate verb, A1 shows the effect of the action and A2-5 represent different semantic meaning according to different predicates [12].

D. *Our Hybrid Approach*

We use our hybrid technique to integrate the two sets of experimental results by the lexical similarities-based approach and the SVM classifier-based approach, as shown in the Equation (20). Here, α and β are parameters which are learned from the training data, *result*, $result_L$ and $result_S$ are the results of our hybrid approach, the lexical similarities-based approach and the SVM classifier-based approach, respectively. Their values are assigned as 0 or 1. If the value of *result* is 1, then we determine that the text entails the hypothesis; otherwise, the text does not imply the hypothesis.

$$result = \alpha \times result_L + \beta \times result_S \qquad (20)$$

V. Empirical Results

In our experiments, we use the labeled texts and hypothesis pairs provided by the Text Analysis Conference (TAC). The data sets RTE1_DEV, RTE2_DEV and RTE3_DEV are used as the training data and RTE1_TEST, RTE2_TEST, RTE3_TEST and RTE4_TEST are used as the testing data. Table 1 gives the comparison of the experimental results of our hybrid approach and the SVM classifier method to recognize the textual entailment on those seven datasets.

The results of TABLE 1 show that: (1) the precision, recall the F-measure of our hybrid approach are better than those of the best result of PASCAL RTE Challenge 2009 on the dataset RTE4_TEST, and are also better than those of the work of Ofoghi &Yearwood [1] on the datasets RTE1_TEST, RTE2_TEST, RTE3_TEST, and RTE4_TEST.The experimental results demonstrate that our technique is effective to solve the problem of textual entailment recognition.

TABLE I. Recognition Textual entailment results

Data	Method	Precision	Recall	F-Measure
RTE1_DEV	SVM	0.547	0.547	0.546
	Our approach	0.739	0.740	0.739
RTE2_DEV	SVM	0.559	0.559	0.558
	Our approach	0.719	0.719	0.719
RTE3_DEV	SVM	0.545	0.546	0.545
	Our approach	0.724	0.724	0.724
RTE1_TEST	Ofoghi(2010)	0.520	0.521	0.520
	SVM	0.520	0.521	0.520
	Our approach	0.731	0.731	0.731
RTE2_TEST	Ofoghi(2010)	0.517	0.518	0.518
	SVM	0.542	0.542	0.541
	Our approach	0.689	0.689	0.689
RTE3_TEST	Ofoghi(2010)	0.551	0.551	0.551
	SVM	0.570	0.570	0.570
	Our approach	0.708	0.708	0.708
RTE4_TEST	PASCAL RTE Challenge	0.746	0.745	0.746
	Ofoghi(2010)	0.556	0.556	0.556
	SVM	0.567	0.567	0.567
	Our approach	0.753	0.753	0.753

VI. Conclusions

Recently, more and more efforts have been paid on solve the recognizing textual entailment problem in fields of intelligent information processing and knowledge acquisition. In this paper, we propose a hybrid technique which includes two methods for the recognizing textual entailment task. First method is based on lexical similarities, and the second method is founded on the classifier of support vector machine. We propose two kinds of classification features which include features based on semantic roles, and ones based on dependency relations and WordNet. Our experimental results demonstrate that our system is feasible to recognizing textual entailment. In the future, we will use other semantic analysis approaches to solve textual entailment recognition problem.

References

[1] Bahadorreza Ofoghi, John Yearwood. Learning Parse-Free Event-Based Features for Textual Entailment Recognition [J]. AI 2010, LNAI 6464, pp.184-193,2010.

[2] Mehdad, Y. (2009). Automatic cost estimation for tree edit distance using particle swarm optimization. In Proc. of the 47th Annual Meeting of ACL and the 4th Int. Joint Conf. on Nat. Lang. Processing of AFNLP, pp. 289-292, Singapore.

[3] Qiu, L., Kan, M. Y., & Chua, T. (2006). Paraphrase recognition via dissimilarity significance classification. In Proc. of the Conf. on EMNLP, pp. 18-26, Sydney, Australia.

[4] Tsatsaronis, G., Varlamis, I., & Vazirgiannis, M. (2010). Text relatedness based on a word thesaurus. Artificial Intelligence Research, 37, 1-39.

[5] Malakasiotis, P. (2009). Paraphrase recognition using machine learning to combine similarity measures. In Proc. of the 47th Annual Meeting of ACL and the 4th Int. Joint Conf. on Nat. on Nat. Lang. Processing of AFNLP, Singapore.

[6] Malakasiotis, P., & Androutsopoulos, I. (2007). Learning textual entailment using SVMs and string similarity measures. In Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 42-47, Prague.

[7] Nielsen, R., Ward, W., & Martin, J. (2009). Recognizing entailment in intelligent tutoring systems. Nat. Lang. Engineering, 15(4), 479-501.

[8] Zanzotto, F. M., Pennacchiotti, M., & Moschitti, A. (2009). A machine-learning approach to textual entailment recognition. Nat. Lang. Engineering, 15(4), 551-582.

[9] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In Proceedings of the CoNLL-2011 Shared Task, 2011.

[10] Jinho D. Choi, Ph.D. Thesis. Optimization of Natural Language Processing Components for Robustness and Scalability. University of Colorado Boulder, Computer Science and Cognitive Science, 2012.

[11] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, K. Miller. 1990. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235–244.

[12] Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. Comput. Linguist. **31**(1), 71–106 (2005). doi:10.1162/0891201053630264

[13] Dagan I, Dolan B, Magnini B, et al. Recognizing textual entailment: Rational, evaluation and approaches [J]. Natural Language Engineering, 2009, 15 (4): 1-17.

[14] Icon, A & Prodromos, M. A Survey of Paraphrasing and Textual Entailment Methods [J]. Journal of Artificial Intelligence Research 38, 2010.

[15] Asher Stern, Eyal Shnarch, Amnon Lotan, Shachar Mirkin, Lili Kotlerman, Naomi Zeichner, Jonathan Berant, and Ido Dagan. Rule Chaining and Approximate Match in textual inference. In the proceedings of the 2nd Text Analysis Conference, 2010.

[16] Milen Kouylekov,Alessio Bosca,Luca Dini. EDITS 3.0 at RTE-7. In the proceedings of the 3rd Text Analysis Conference, 2011.

[17] Milen Kouylekov, Yashar Mehdad, Matteo Negri, Elena Cabrio. FBK Participation in RTE6:Main and KBP Validation Task. In the proceedings of the 2nd Text Analysis Conference, 2010.

[18] Partha Pakray,Santanu Pal,Soujanya Poria,Sivaji Bandyopadhyay, Alexander Gelbukh. A Textual Entailment System using Anaphora Resolution. In the proceedings of the 3rd Text Analysis Conference, 2011.