Chinese Textual Entailment Recognition Enhanced with Word Embedding

Zhichang Zhang, Dongren Yao, Yali Pang, Xiaoyong Lu

School of Computer Science and Engineering, Northwest Normal University, Lanzhou, China
zzc@nwnu.edu.cn, wade330628704@163.com, {pyl, luxy}@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. By evaluating on NTCIR-11 RITE3 Simplified Chinese subtask data set, this paper firstly demonstrates and compares the performance of Chinese textual entailment recognition models that combine different lexical, syntactic, and semantic features. Then a word embedding based lexical entailment module is added to enhance classification ability of our system further. The experimental results show that the word embedding for lexical semantic relation reasoning is effective and efficient in Chinese textual entailment.

Keywords: Chinese Textual Entailment; RITE; Lexical Entailment; word embedding

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 [1]. A wide range of natural language applications which demand semantic inference need RTE to fulfill their tasks. For example, given the question "*Which team won the NBA championship in* 2013-2014?", a question answering system needs to identify whether the text "*Popvich led Spurs to their fifth straight title on June* 16,2014" entails a hypothesized answer form "*San Antonio Spurs win the championship in* 2013-2014" or not.

Many approaches have been proposed to solve this problem. Logic-based [2, 3] and decoding-based [4, 7] approaches require large-scale inference rules provided ahead to make entailment decision. However, it is a time consuming and labor intensive process to build enough number of entailment rules.

Machine Learning based recognition approaches [5, 6] therefore are used more widely where the task to detect whether a text *T* entails a hypothesis *H* is taken as a binary classification problem, and classification models can be trained automatically on a labeled textual entailment dataset. While this type of approach is easy to implement, the features demanded by the classification models should be carefully designed in order to achieving state-of-the-art recognition performance. Many features covering from lexical to semantic layer are studied and compared in existing research. But for Chinese textual entailment recognition, the analysis of different feature's contribution to recognition performance improvement is still not sufficient.

Based on our machine learning based textual entailment recognition system which took part in NTCIR-11 RITE-3 Chinese textual entailment Binary-class task, this paper presents the features used by the system and their effort to recognition performance. Particularly, a word embedding based lexical entailment module is added into the system and the system performance improved further.

The rest of the paper is organized as follows: Section 2 describes the features and algorithm employed in our system at NTCIR-11 evaluation. Section 3 details a word embedding based lexical entailment module added to improve our system's inference ability. 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 System description

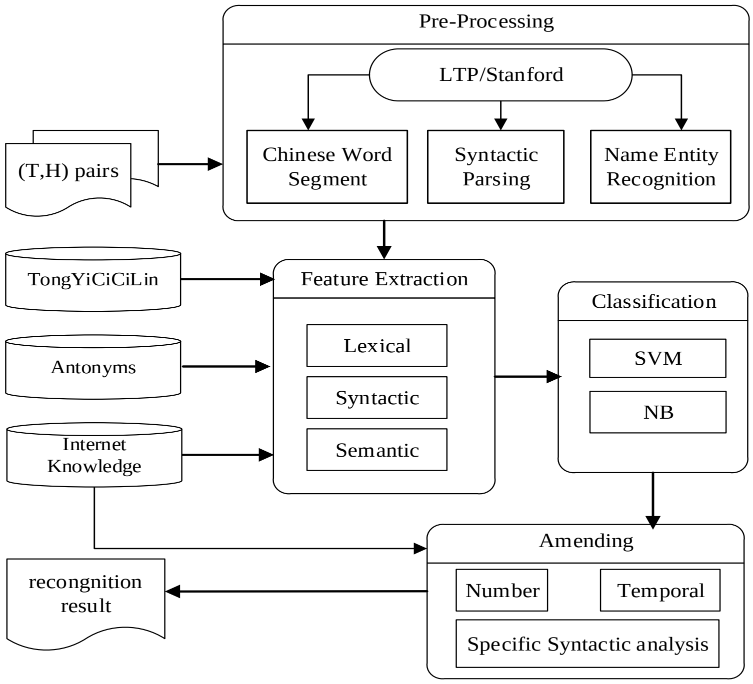

Figure1. System architecture

Our system that based on machine learning takes the Chinese textual entailment recognition task as a binary classification problem. The system uses the different matching similarities between *T* and *H* as important features for this task, on the understanding 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. The system contains four main modules including pre-processing, feature extraction, classification, and amending module. Figure 1 shows more details in our system.

The preprocessing module uses several different NLP resources and tools for basic processing like word segment, POS, and NE recognition. We use LTP[1] tool for word segment, POS tagging and dependency syntactic parsing and Stanford classifier for named entity recognition.

2.1 Features extraction

2.1.1 Lexical features

Existing researches show that lexical features are easy to obtain and effective for textual entailment recognition. We take over some traditional lexical features, and also supplement several new features according to the characteristic of Chinese textual relation. These lexical features can help our system solve some problems such as：

<t>《罗马假期》是 1953 年拍摄的浪漫爱情片。(“Roman Holiday” is a romantic love drama shot in 1953.) </t>

<h>《罗马假期》是浪漫爱情片，于 1953 年拍摄。(“Roman holiday” is a romantic love story, which is shot in 1953.) </h>

With a very high lexical similarity, our system will classify them as “entailment”, which is the right semantic relation between two texts.

Following table 1 illustrates the lexical features used in our system, where the symbol *T* stands for the text in text pair and *H* for the hypothesis.

Table 1. Lexical features

Feature Name	Comment	Formula
Word overlap	The overlap of words between two texts	$E_1=\lvert T \wedge H\rvert / \lvert H\rvert$ $E_2=\lvert T \wedge H\rvert / \lvert T\rvert$ $E=(2*E_1*E_2)/(E_1+E_2)$
Length difference	Using text length to distinguish entailment direction	$Lt(T,H)=\lvert Len(T)-Len(H)\rvert$
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

[1] http://www.ltp-cloud.com/demo/

Tongyicicilin semantic similarity [8]	Using Tongyicicilin to calculate the similarity between different words	$f_{CilinSim} = \frac{1}{2}(\frac{1}{m}\sum_{i=1}^{m}\max\{sim_w(w_{1i}, w_{2j}) \mid 1 \le j \le n\} + \frac{1}{n}\sum_{j=1}^{n}\max\{sim_w(w_{1i}, w_{2j}) \mid 1 \le i \le m\})$
Number of antonyms	Using the Web resource to count the number of antonyms	$f_a = (\#w_T - \#w_H) \bmod 2$
Number of negative words	Combining the number of antonyms to assist decision	$f_n = (\#w_T - \#w_H) \bmod 2$
Overlap of named entity	Named entities can show the text topics in a way	$T_{NE} = \lvert T \wedge H \rvert / \lvert H \rvert$ $H_{NE} = \lvert T \wedge H \rvert / \lvert T \rvert$ $L_{NE} = (2 * T_{NE} * H_{NE}) / (T_{NE} + H_{NE})$

These lexical features cannot capture the syntactic structures of two texts and view a text as a bag of words. But for some text pairs, their entailment relation could not be correctly detected if their syntactic structure is not considered and compared. Using the lexical features as a base, more complex features from syntactic to semantic aspects should be checked too.

2.1.2 Syntactic features

As we discussed above, lexical features can't capture the text syntactic information. For example:

<pair id="30">

<t>《瀛台泣血记》为清朝宫廷女官裕德龄所撰写的一本有关光绪皇帝一生的故事。(*Son of heaven* that talks about Guangxu emperor's whole life is written by DeLing yu who is a female official in Qing dynasty)</t>

<h>《瀛台泣血记》为清朝宫廷女官裕光绪所撰写的一本有关德龄皇帝一生的故事。(*Son of heaven* that talks about DeLing emperor's whole life is written by Guangxu yu who is a female official in Qing dynasty)</h>

</pair>

<pair id="237">

<t>日本于 2005 年发行上映的动画电影《蒸汽男孩》，其故事背景以英国 1851 年万国博览会为主。(An animated movie in Japan, *Steamboy*, which was on in 2005 shows a background about the World Exhibition held by England in 1851.) </t>

<h>日本于 2005 年发行上映的动画电影《蒸汽男孩》，其故事背景以 1851 年英国万国博览会为主。(An animated movie in Japan, *Steamboy*, which was on in 2005 shows a background about England held the World Exhibition in 1851.) </h>

</pair>

Due to 100% percent character similarity, machine learning algorithm predicates both of them as “entailment” if just utilizing lexical features. Apparently, we can see that in the text pair with id being “30”, *T* don’t entail *H*, *T* and *H* are contradicted on mentioned entities. When the most of words in the text *T* occur in hypothesis *H*, the order or syntactic structure of these words will determine semantic relation between two texts.

From this view, we put forward the method “Minimum syntactic trees” [13] to incorporate some import syntactic features into our system. The main idea of “Minimum syntactic trees” is that two minimum trees which are generated by clipping the complete syntactic structure trees, and without useless nodes (no effect on entailment) can provide semantic features in the simplest way.

2.2 Classifier

Two machine learning methods are achieved to predicate the entailment relation, and the open source tool scikit-learn [10] is employed for classification in this system. SVM which is regarded as the best supervised learning method in general text classification didn’t obtain the best result during the training phase, but got the best score on testing set. The Naïve Bayes method based on Gaussian distribution achieved the best performance on training set, but didn’t do as good as SVM during the testing phases.

2.3 Amending module

Numbers occurs in different forms (Arabic numerals, Chinese numerals, or numbers combing with Chinese character) among the sentences. For one thing, we need to transfer different number form into the same format. For another, some basis operation between these numbers should be taken into consideration. Such as:

<pair id="311">
<t>火地群岛总面积73, 753平方公里。(The total area of Tierra del Fuego is 73,753 square kilometers.)</t>
<h>火地群岛总面积超过六万平方公里。(The total area of Tierra del Fuego is over than sixty thousand square kilometers .)</h>
</pair>

Our system need to understand the word “超过(over)” and the fact that 73,753 is larger than “六万(sixty thousand)” to make a correct decision.

Furthermore, temporal recognition also faces the similar situation. So far, we made some rules solve this reasoning problem. Although these rules can improve our performance, they only work in a very limited circumstance.

3 Lexical entailment module

After participating NTCIR-11 RITE-3 evaluation, we added a lexical entailment module into our system to enhance the ability of reasoning on lexical level, and corresponding lexical knowledge were considered and used as follow.

3.1 Lexical Entailment vs. Semantic Relationship

When some researchers have applied semantic relation classification to lexical entailment, we think that semantic relation is different with lexical entailment in many respects. Lexical entailment is not just a superset of other known semantic relations, it is rather designed to select those sub-cases of other lexical relations that are needed for applied entailment inference [18]. Turney and Mohammad provide some further exploratory point of view and formulate a hypothesis [14]:

> ***Semantic relation subcategories hypothesis***: Lexical entailment is not a superset of high-level categories of semantic relations, but it's a superset of lower-level subcategories of semantic relations.

When many methods have been proposed to solve word semantic relation problem, we utilize word embedding method to reveal the entailment relationship between words in this paper.

3.2 Word Embedding Training

Mikolov et al. [15] proposed two log-linear models, namely the Skip-gram and CBOW model, to efficiently induce word embeddings. These two models can be trained on a large scale corpus. We employ the Skip-gram model for estimating word embeddings in identifying lexical entailment relationship. The Skip-gram model adopts log-linear classifiers to predict context words given the current word $w(t)$ as input, and $w(t)$ is projected to its embedding. Then, log-linear classifiers takes the embedding as input and predict $w(t)$'s context words within a certain range. After maximizing the log-likelihood over the entire dataset with stochastic gradient descent (SGD), the embeddings are learned.

3.3 Lexical Entailment Relationship Learning

Word embedding preserve interesting linguistic regularities like $v(king) - v(queen) \approx v(man) - v(woman)$. It indicates that the embedding offsets indeed represent the shared semantic relation between two word pairs. Fu's work [16] shows that word embedding can measure semantic relationship effectively. When we deal with lexical entailment, this method should also be considered.

So our newly added module is based on a similarity difference hypothesis with word embedding:

> ***Similarity difference hypothesis with word embedding:*** The tendency of

a entails *b* based on their word embedding is correlated with the difference in their similarities **sim**(a, r_i)-**sim**(b, r_j), referring to a set of entailment relation pairs $(r_i, r_j) \in \mathrm{R}$.

According to this hypothesis, we provide some reference pairs (r_1, r_2) which have kinds of entailment relationships to recognize whether two words (a, b) appearing in Text (T) and Hypothesis (H) have the same relationship as r_1 and r_2.

For brevity, we write a for both a word and its associated vector $<a_1, \ldots, a_n>$, and b, r_1, r_2 are in same way. The symbols $\mathbf{v}$ and $\mathbf{v}'$ are used to denote the vector calculation results $r_1 - r_2 + a$ and $r_2 - r_1 + b$ respectively.

Define the function $\mathbf{sim}_{w2v}$ to represent the following procedure:

(a) Based on word embedding vectors, search three words w_1, w_2, w_3 from all words in corpus, which have the largest cosine similarities with $\mathbf{v}$, and search also three words w'_1, w'_2, w'_3 for $\mathbf{v}'$ in similar way. We also use w_1, w_2, w_3 to represent the associated word embedding vectors of these words for convenience.

(b) Compose the matrix $\mathbf{M}$ for $\mathbf{v}$ taking the word vectors $\boldsymbol{w_1}$, $\boldsymbol{w_2}$, $\boldsymbol{w_3}$ as rows in the matrix, or $\mathbf{M} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3})^{\boldsymbol{T}}$. Similarly, compose the matrix $\mathbf{M}'$ for $\mathbf{v}'$ and $\mathbf{M}' = (\boldsymbol{w}'_1, \boldsymbol{w}'_2, \boldsymbol{w}'_3)^{\boldsymbol{T}}$.

Define the following model to show this procedure:

$$\begin{aligned} \mathbf{M} &= \mathrm{sim}_{w2v}(r_1 - r_2 + a) \\ \mathbf{M}' &= \mathrm{sim}_{w2v}(r_2 - r_1 + b) \end{aligned}, \tag{1}$$

So 3×300 matrix $\mathbf{M}$ contains three word vectors associating to the most possible three words that exist the relationship with a as which exists between r_1 and r_2, and 3×300 matrix $\mathbf{M}'$ also contains three vectors corresponding to the most possible three words that exist the relationship with b as which exists between r_2 and r_1. Define a measure Thr_k as following:

$$\boldsymbol{Thr_k} = \frac{\mathrm{Max}(\mathrm{sim}_{\cos}(\boldsymbol{w_i}, \boldsymbol{b})) + \mathrm{Max}(\mathrm{sim}_{\cos}(\boldsymbol{w'_j}, \boldsymbol{a}))}{2} \tag{2}$$

Where i, j = (1, 2, 3), k= (1, 2, 3, 4, 5), and $\mathbf{sim}_{\cos}$ is just cosine similarity function. Different k represents different lexical entailment relationship which will be considered.

For a word pair (a, b), if the Thr_k value between them is larger than *0.80*, we think that lexical entailment relationship k exists between a and b. For example, given a reference pair like (r_1= "中国(China)", r_2="北京(Beijing)") and a word pair a="日本(Japan)" and b="东京(Tokyo)", use Equation (1) to obtain $\mathbf{M}$=["东京(Tokyo)", "横滨(Yokohama)", "名古屋(Nagoya)"](use the word instead of their vectors to make the example easier to understand) and $\mathbf{M}'$=["日本(Japan)", "韩国(Korea)", "关西地区(Kansai region)"]. After filtering each matrix with Thr_k score using Equation 2, we will treat a="日本(Japan)" and

b="东京(Tokyo)" as entailment lexical pair.

Five relationships will be taken into considered (object-component, synonymy, contradictory, set-member, agent-recipient). These relationships are only based on experience and convenience. Consider cases of object-component instances "机翼(aerofoil)","飞机(plane)", "轮胎(tyre)", after simple operations (**add** and **minus**) on 300 dimensional word embedding vectors trained in Skip-gram model, the result is the same word "汽车(car)" which we wish. Although the relationships of synonymy and contradictory have been used as lexical features, we still keep these relationships. Another case about set-member also gets a good performance. Make "森林(forest)","树木(tree)", "湖泊(lakes)" as input, and the outcome is "河水(river water)". And agent-recipient relationship inputs as "医生(doctor)", "患者(patient)", "商户(merchant)", the result is "顾客(custom)". So we store some word-pairs having these relationship we told before. After calculation, some lexical entailment relationship can be reasoned. The experimental result shows that this kind of lexical entailment recognition increase the performance of textual entailment classification apparently.

4 Experimental result and analysis

4.1 Data and evaluation measures

Our textual entailment recognition system will be tested on NTCIR-11 RITE3 Simplified Chinese binary-class subtask data [9, 11], which consists of 1200 text pairs. The evaluation of the system is performed by applying it on recognizing entailment relation of test text pairs. The topics of these text pairs cover many domains including history, political, geography, sports etc. Many linguistic phenomena such as inference, paraphrase, and clause are contained in the evaluation data.

The performance of the system is evaluated using *precision*, *recall*, *F-score* and their variations as measures. *Precision* is defined as the ratio of the number of correctly made decisions to the total number of decisions, while *recall* is the number of pairs classified correctly over the total number of pairs. *F-score* is the harmonic mean of *precision* and *recall*.

There are other variations of *precision*, *recall* and *F-score*, concerning the focus on entailment (denote by $\boldsymbol{Y}$) or not (denote by $\boldsymbol{N}$). Let $\boldsymbol{C}=(c_{ij})_{2\times2}$ be a 2×2 confusion matrix, where c_{ij} represents the real category i and classified category j (hence $i, j\in\{\boldsymbol{Y}, \boldsymbol{N}\}$),

$$C=\begin{pmatrix} c_{YY} & c_{YN} \\ c_{NY} & c_{NN} \end{pmatrix}$$

Then all variations of *precision*, *recall* and *F-score* are defined as follows:

$$Pre_Y = \frac{c_{YY}}{c_{YY} + c_{NY}} \qquad Pre_N = \frac{c_{NN}}{c_{NN} + c_{YN}} \tag{3}$$

$$Rec_Y = \frac{c_{YY}}{c_{YY} + c_{YN}} \qquad Rec_N = \frac{c_{NN}}{c_{NN} + c_{NY}} \tag{4}$$

$$F_Y = \frac{2 \times Rec_Y \times Pre_Y}{Rec_Y + Pre_Y} \qquad F_N = \frac{2 \times Rec_N \times Pre_N}{Rec_N + Pre_N} \tag{5}$$

Furthermore, NTCIR-11 RITE3 competition used *Macro-F* and *Pre* as the two most important evaluation criteria, which are define as:

$$Macro\text{-}F = \frac{F_Y + F_N}{2} \qquad Pre = \frac{Pre_Y + Pre_N}{2} \tag{6}$$

4.2 Experimental result and analysis

We firstly compare the contributions to system performance by different features. Figure 2 demonstrates the effect of different features in SVM decision model. Overall, the performance of the system improves gradually when the number of features increasing. We can also see that the lexical entailment feature can improve the performance of the system from *Macro-F* 59.03 to 62.75, which demonstrates that word embedding is effective in textual entailment recognition.

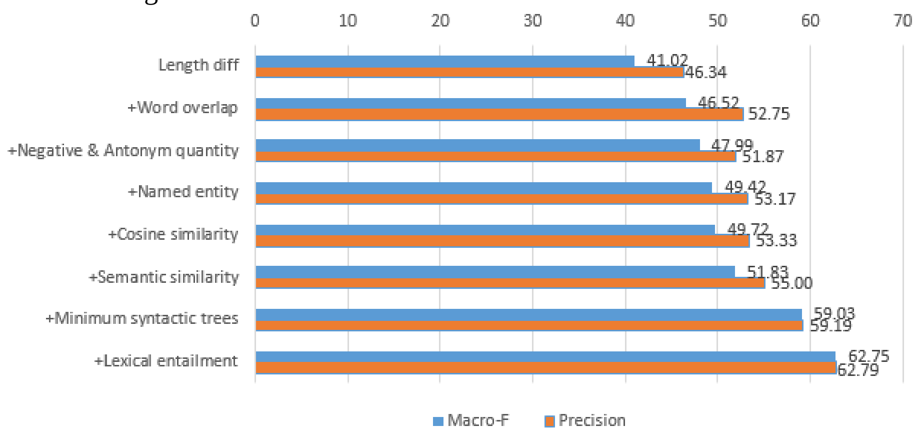

Figure2. System performance with different feature combinations

We also compare the performance of our system with other participators in NTCIR-11 RITE3 Simplified Chinese subtask. The formal run results of all systems on binary-class subtask are shown in Table 2. Our system **NWNW** achieves a second best result on all evaluation measures except F_y [11].

Table 2. The formula run result of all participated systems on BC

Participants	*Macro-F*	*Pre*	F_Y	F_N
BUPT	61.51	62.33	67.15	55.86
NWNU	59.71	59.75	60.95	58.47
III&CYUT	56.75	56.75	57.07	56.42
WHUTE	53.48	54.58	60.65	51.49
Yamraj	49.24	49.25	48.69	49.79
ASNLP	44.95	51.50	63.94	25.95
IMTKU	42.80	53.25	67.25	18.34
JAVN	42.32	51.17	64.91	19.73
WUST	39.14	52.25	67.39	10.89

We also analyze the effect of every module in our system to the overall performance of the system in terms of all evaluation measure when different classification algorithm is used. We got six run results shown in Table 3 in different ways, and compare them with those values achieved by the best system **BUPT** on NTCIR-11 RITE3 Simplified Chinese subtask.

Table 3. The formal run result of the system NWNU

System	*Macro-F*	*Pre*	F_Y	Pre_Y	Rec_Y	F_N	Pre_N	Rec_N
NWNU-01	45.82	51.75	63.74	51.05	84.83	27.90	55.17	18.67
NWNU-02	51.83	55.00	64.19	53.30	80.67	39.46	60.27	29.33
NWNU-03	58.03	59.00	64.40	56.91	74.17	51.67	62.92	43.83
NWNU-04	58.83	58.83	58.90	58.80	59.00	58.76	58.86	58.67
NWNU-05	59.71	59.75	60.95	59.18	62.83	58.47	60.39	56.67
NWNU-ADD	62.75	62.79	64.33	61.69	66.83	61.16	63.88	58.67
BUPT	**61.51**	**62.33**	**67.15**	**59.54**	**77.00**	**55.86**	**67.45**	**47.67**

Results achieved by NWNU-01 with simple character matching method can be considered as the baseline performance on this evaluation. NWNU-02 with more lexical features gain almost 6% improvement in *Macro-F* measure. The system NWNU-03 which taking "syntactic minimum information tree" into consideration to capture some syntactic information still improve 6% *Macro-F*. Both NWNU-04 and NWNU-05 employ an amending module to deal with some special situation.

Concerning Naïve Bayesian (NB) model has the best performance on RITE-1 and RITE-2 English textual entailment evaluations, all systems from NWNU-01 to NWNU-04 use it as classifier, but NWNU-05 uses SVM model to classify the text pair instead of NB, and this system shows the best performance among these five systems according to the final evaluation results.

Based on the system NWNU-05, a word embedding based lexical entailment module is appended into it and the new version of this system is then called as NWNU-ADD. The experiment result show that the newly added module works very well in recognizing those pairs with real entailment relationship, and the performance of the system enhanced with word embedding outperforms the best system BUPT on NTCIR-11 RITE3 Simplified Chinese subtask. This demonstrates again that word embedding is valuable for recognizing Chinese textual entailment.

The confusion matrices is Figure 3 presents the numbers of text pairs classified by NWNU-05 and NWNU-ADD. We can see that the lexical entailment model gives an effective supplement to NWNU-05.

$$\begin{pmatrix} 377 & 223 \\ 260 & 340 \end{pmatrix}_{\text{NWNU-05}} \quad \begin{pmatrix} 401 & 199 \\ 248 & 352 \end{pmatrix}_{\text{NWNU-ADD}}$$

Figure 3. Classified text pair number confusion matrices

We believe that there is much room left to improve in our system. For one thing, the lexical entailment reasoning ability is still very limited, and more general approach should be proposed to deal with more complex lexical entailment phenomena. For another, more powerful neural network models such as recursive autoencoder [17] can be employed and explored against traditional classifiers or linguistic features.

5 Conclusion

Recognizing textual entailment is an important subtask in many natural language processing applications, and the statistical learning approaches gradually become the mainstream on this task. Using our Chinese textual entailment recognition system that participated in NTCIR-11 RITE-3 task, this paper evaluated the contribution of different linguistic feature types to the quality of the supervised Chinese textual entailment recognition models. Especially, we analyzed the impact of a word embedding based lexical entailment module on system performance. Experimental result shows that word embedding can enhance effectively the ability of the system.

Our future work is two-fold. We should find new features or new methods to express entailment relationship better. Meanwhile we also need to focus on the recognizing multi-directional Chinese textual entailment relation, and this challenge requires entailment system developed in a more robust and reasonable way.

Acknowledgements. This work is supported by grants from 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, NWNU-LKQN-13-23).

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, Dagan I, Magnini B, et al. Machine Learning Challenges: evaluating predictive uncertainty, visual object classification and recognising textual entailment[C]//First Pascal Machine Learning Challenges Workshop (MLCW 2005). 2006.
6. 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.
7. 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.
8. 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
9. 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
10. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011
11. Suguru Matsuyoshi and Yusuke Miyao and Tomohide Shibata and Chuan-Jie Lin and Cheng-Wei Shih and Yotaro Watanabe and Teruko Mitamura, Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task, Proceedings of the 11th NTCIR Conference, Tokyo，Japan ,2014
12. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011
13. Zhang Z, Yao D, Chen S, et al. Chinese Textual Entailment Recognition Based on Syntactic Tree Clipping[M]//Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer International Publishing, 2014: 83-94.
14. Turney P D, Mohammad S M. Experiments with three approaches to recognizing lexical entailment [J]. Natural Language Engineering, 2013: 1-40.
15. Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
16. Fu R, Guo J, Qin B,et al. Learning semantic hierarchies via word embeddings[C]//Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics: Long Papers. 2014, 1.
17. Socher R, Huang E H, Pennin J, et al. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection[C]//Advances in Neural Information Processing Systems. 2011: 801-809.
18. Zhitomirsky-Geffet M, Dagan I. Bootstrapping distributional feature vector quality [J]. Computational Linguistics, 2009, 35(3): 435-461.