Effective Semantic Relationship Classification of Context-free Chinese Words with Simple Surface and Embedding Features

Yunxiao Zhou[1], Man Lan[1,2⋆], and Yuanbin Wu[1,2⋆]

[1] School of Computer Science and Software Engineering,
East China Normal University, Shanghai 200062, P.R.China
[2] Shanghai Key Laboratory of Multidimensional Information Processing
51164500061@stu.ecnu.edu.cn, mlan,ybwu@cs.ecnu.edu.cn

Abstract. This paper describes the system we submitted to Task 1, i.e., Chinese Word Semantic Relation Classification, in NLPCC 2017. Given a pair of context-free Chinese words, this task is to predict the semantic relationships of them among four categories: Synonym, Antonym, Hyponym and Meronym. We design and investigate several surface features and embedding features containing word level and character level embeddings together with supervised machine learning methods to address this task. Officially released results show that our system ranks above average.

Keywords: Semantic relation classification, Context-free Chinese words, Surface and embedding features, Supervised machine learning

1 Introduction

The Chinese Word Semantic Relation Classification task [1] in NLPCC 2017 is to provide a standard testbed for automatic classification of word semantic relations, which benefits many downstream applications in Natural Language Processing (NLP), such as the construction of semantic networks and recognizing textual entailment [2, 3]. Specifically, this task provides pairs of *context-free* Chinese words with different length, and participants are required to classify the semantic relationships of them into four categories: *Synonym*, *Antonym*, *Hyponym* and *Meronym*. These four categories of semantic relations are defined according to quite general ones in lexical semantics, and given two words A and B, their definitions and corresponding examples are shown in Table 1.

Clearly, the purpose of this shared task is to automatically identify semantic relationships of *context-free* Chinese word pairs, which is a bit different from previous studies which identified semantic relations between terms in given texts [3–5]. In many cases, the semantics of a word depends on its context in text. Without context between words, the identification of semantic relationships is

⋆ Corresponding authors.

Table 1. Definitions of semantic relations and corresponding examples.

Relation	Definition	A	B
Synonym	A is similar to B	消费	花
Hyponym	A is a kind of B	钢笔	笔
Meronym	A is a part of B	笔帽	钢笔
Antonym	A is contrast to B	骄傲	谦虚

more challenging. To address this task, we explore a supervised machine learning method which uses several surface features, e.g., character overlaps, length, positional overlaps, etc. In recent years, more and more studies have focused on word or character embeddings as an alternative to traditional hand-crafted features [6–10]. Therefore we examine several types of word level and character level embeddings. Besides, we perform a series of experiments to explore the effectiveness of feature types and supervised machine learning algorithms.

The rest of this paper is organized as follows. Section 2 describes our system framework including feature engineering and learning algorithms. The experiments on training and test data are reported in Senction 3. Finally, this work is concluded in Section 4.

2 System Description

To perform semantic relationship classification of context-free Chinese words, we adopt supervised learning algorithm with surface features extracted from given words and various embedding features. In next, we will introduce feature engineering and learning algorithms.

2.1 Surface Features

Without context, given a pair of Chinese words, we explore four types of surface features, i.e., length features, character overlaps, positional overlaps and sequential overlaps.

2.1.1 Length Features

Given two Chinese words A and B, they may contain different length of characters. Generally, the longer words contain more specific information than the shorter ones, which may imply a kind of *hyponym* relationship, for example, “花” and “菊花”. Therefore we design six features to capture this length information using the following six measure functions: $|A|$, $|B|$, $|A| - |B|$, $|B| - |A|$, $|A \cup B|$, $|A \cap B|$, where $|A|$ stands for the number of characters in word A, $|A \cup B|$ denotes the set size of non-repeated words found in either A or B and $|A \cap B|$ stands for the set size of shared characters found in both A and B.

2.1.2 Character Overlaps

Except for antonym relationship, the remaining three semantic relationships more or less indicate a certain degree of semantic similarity or relatedness between two words. Therefore, in light of our previous work addressing semantic relatedness and textual entailment [11], we adopt commonly used functions to calculate the similarity between word A and word B based on their character overlaps. Table 2 shows these three functions used in this work. As a result, we get four character overlap features.

Table 2. Character overlaps similarity measures and their definitions used in our experiments.

Measure	Definition
Jaccard	$S_{jace} = \|A \cap B\|/\|A \cup B\|$
Dice	$S_{dice} = 2 * \|A \cap B\|/(\|A\| + \|B\|)$
Overlap	$S_{over} = \|A \cap B\|/\|A\|$ and $\|A \cap B\|/\|B\|$

2.1.3 Positional Overlaps

The above character overlap feature only records the degree of overlap between two words. In fact, the position of character overlap is crucial for hyponym and meronym relationships. Generally, the character overlaps between two words exist mostly on the head or tail of words. For example, hyponym relation may share the same last character, e.g., "植物油" and "花生油", "房间" and "卫生间". While meronym relation may share the same first character, e.g., " 鞋子" and "鞋跟". Therefore, we design the following features to record the positional information of overlaps. Given two words A and B, we implement four types of binary features: (1) whether the prefix of A is the same as B, (2) whether the suffix of A is the same as B, (3) whether the prefix of A is the same as the suffix of B and (4) whether the suffix of A is the same as the prefix of B. Considering these Chinese words with variant length, we set the length of prefix or suffix as 1 and 2, respectively. Totally, we collect eight positional overlap features.

2.1.4 Sequential Overlaps

Previous three features do not take the sequential information into account, while sequential overlaps are quite important for measuring the matching degree between two Chinese words. For example, synonym relationship may contain some instances that one word is sequentially contained in another word, e.g., "宁波" and "宁波市", " 法国" and "法兰西共和国". So we design the following features to record the sequential information of overlaps. First, we implement two types of binary features: given two words A and B, we record whether the word A is sequentially included in the word B and vice versa. What's more, we compute the *longest common prefix* and *longest common suffix* for each word pair. As a result, we get four sequential overlap features.

2.2 Embedding Features

The above surface features only capture semantic information between two words based on their surface forms, while word embedding is a continuous-valued vector representation for each word, which usually carries syntactic and semantic information. Therefore, the embedding features are designed to utilize embeddings to obtain the semantic relation between two words. In this work, we train word vectors by using Google *word2vec*[3] [6] with different dimensions, i.e., 50, 100, 200 and 300. The corpus that we used for training word vectors is Wikimedia dumps which will be described in Section 3.1. Moreover, as the most fine-grained representation of Chinese, character is the smallest meaningful form in Chinese language, i.e., morpheme. Since different combinations of characters may represent different meanings, we also adopt the fine-grained character level embeddings for this task.

2.2.1 Word Embedding Features

After acquiring the vectors of two words in the word pair, we explore five different ways of interaction in order to capture the semantic relation of the two words as much as possible. The operations between two vectors include *concatenation*, *multiplication*, *summation*, *subtraction* and the *min-max-mean pooling* operations. In our preliminary experiments, word vectors with dimensionality of 100 achieves the best performance, thus in the following experiments, we adopt 100 dimensional word vectors. Besides, preliminary experiments also show that the first two interactive operations have better performance, so we only adopt the *concatenation* and *multiplication* operations between two word vectors as word embedding features.

2.2.2 Character Embedding Features

Considering that character is the smallest meaningful form in Chinese language, we extract features from the character embeddings[4] which are provided by NLPCC 2017 Task 2 [12] with dimensions of 50, 100, 200 and 300. We simply adopt the *min*, *max* and *mean* pooling operations on all characters in a word to obtain word vectors. Similar to word embedding features, we also use *summation*, *subtraction*, *concatenation*, *multiplication*, and the *min-max-mean pooling* operations to get the interaction information of two words. In our preliminary experiments, the *summation* and *subtraction* operations with 300 dimensional character embeddings achieve the best performance, so we adopt these two operations and obtain $1,800$ dimensional vectors as character embedding features.

2.3 Learning Algorithm

We grant this task as a four-way classification task and explore six supervised machine learning algorithms: Logistic Regression (LR) implemented in *Liblin-*

[3] https://code.google.com/archive/p/word2vec
[4] https://pan.baidu.com/s/1mhPddpu

ear[5], Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), RandomForest and AdaBoost all implemented in *scikit-learn tools*[6], and XGBoost implemented in *xgboost*[7].

3 Experiments

3.1 Datasets

There are 200 training word pairs and 2000 test word pairs provided by task organizers. Table 3 shows the statistics and distribution of all these word pairs. We perform 3-fold cross-validation on training data set to build classification models.

Table 3. The statistics and distribution of training and test data sets.

Dataset	Synonym	Antonym	Hyponym	Meronym	Total
train	50	50	50	50	200
test	500	500	500	500	2,000

The Chinese corpus we used to train word and character vectors is Wikimedia dumps[8], which contains approximate $876,239$ Web pages. These Web page contents are extracted by Wikipedia Extractor[9] and a total of $3,736,800$ sentences are collected after preprocessing. In order to train simplified Chinese word vectors, we first convert traditional Chinese texts into simplified Chinese texts using OpenCC[10] and then tokenize them with jieba tokenizer[11].

3.2 Evaluation Metrics

The official performance evaluation criterion is *macro-averaged F1-score*, which is calculated among four classes (i.e., synonym, antonym, hyponym and meronym) as follows:

$$F_{macro} = \frac{F_{Syn} + F_{Ant} + F_{Hyp} + F_{Mer}}{4} \quad (1)$$

[5] https://www.csie.ntu.edu.tw/ cjlin/liblinear/
[6] http://scikit-learn.org/stable/
[7] https://github.com/dmlc/xgboost
[8] https://archive.org/details/zhwiki-20160501
[9] https://github.com/bwbaugh/wikipedia-extractor
[10] https://pypi.python.org/pypi/OpenCC
[11] https://github.com/fxsjy/jieba

3.3 Experiments on Training Data

Firstly, in order to explore the effectiveness of each feature type, we perform a series of experiments. Table 4 lists the comparison of different contributions made by different features on training set using 3-fold cross-validation with *Logistic Regression* algorithm. We observe the following findings.

(1) All feature types make contributions to semantic relation classification. And the combination of all types of features not only achieves the best performance for the overall classification but also for each semantic category.

(2) The first four surface features act as baseline and they perform better results on synonym and antonym relationships than on the other two semantic relationships. The possible reason is that hyponym and meronym are relatively abstract and surface features cannot adequately capture the semantic information between words.

(3) Word embedding features make a great contribution to semantic relation classification of four classes, especially for meronym and hyponym relationships. It maybe because the pre-trained word embedding usually carries syntactic and semantic information which is benefit for word pair semantic relation prediction.

Table 4. Performance of different features on training data in terms of F1-score(%). ".+" means to add current features to the previous feature set. The numbers in the brackets are the performance increments compared with the previous results.

Features	$\mathbf{F_{Syn}}$	$\mathbf{F_{Ant}}$	$\mathbf{F_{Hyp}}$	$\mathbf{F_{Mer}}$	$\mathbf{F_{macro}}$
Length	51.4	59.1	16.2	0.0	31.7
.+Character Overlaps	52.9	57.9	20.2	6.4	34.3 (+2.6)
.+Positional Overlaps	63.0	56.8	34.4	49.9	51.0 (+16.7)
.+Sequential Overlaps	64.7	56.0	38.2	49.9	52.2 (+1.2)
.+Word Embedding	76.0	83.8	75.3	90.2	81.3 (+29.1)
.+Character Embedding	**80.0**	**86.6**	**82.5**	**92.5**	**85.4** (+4.1)

Secondly, we also explore the performance of different supervised learning algorithms. Table 5 lists the comparison of different learning algorithms with all above features. Clearly, Logistic Regression algorithm outperforms other algorithms.

Therefore, the system configuration for our final submission is all features and LR algorithm.

3.4 Results on Test Data

Table 6 shows the results of our system and the top-ranked systems provided by organizers for this semantic relation classification task. Compared with the top ranked systems, there is much room for improvement in our work, especially for the classification performance of synonym relationship. There are several

Table 5. Performance of different learning algorithms on training data in terms of F1-score(%).

Algorithms	$\mathbf{F_{Syn}}$	$\mathbf{F_{Ant}}$	$\mathbf{F_{Hyp}}$	$\mathbf{F_{Mer}}$	$\mathbf{F_{macro}}$
LR	**80.0**	**86.6**	**82.5**	**92.5**	**85.4**
SVM	80.0	85.4	80.0	92.3	84.4
SGD	70.5	85.7	72.7	88.9	79.5
XGBoost	61.1	68.6	52.9	75.7	64.6
AdaBoost	50.4	42.0	52.7	38.1	45.8
RandomForest	78.4	80.4	73.3	88.9	80.2

possible reasons for this performance lag. First, the training set is too small to train a robust classification model with strong generalization. Building a large training data set with external resources is necessary. Second, we have not used extra semantic dictionary or coupus such as Tongyici Cilin [13] and Hownet [14] which may be effective for word pair semantic relation classification. Third, we only extract features from two words that need to be classified and have not used some extended resources like the sentences returned from search engines when retrieving the two words. Besides, compared with the results on training data, we see the performance on test data is much lower than that on training set. We observe and find that the Chinese words in the training data are formal, while some informal words exist in the test data, for example, "傻不拉几" and "SIM 卡", which may be the possible reason. Besides, the word pairs in test data are more diverse and are more difficult to identify. Moreover, although most of the word vectors can be trained from large corpora, almost 12% of the words in test data are missed in the word embedding dictionary, so word embedding features may not be as effective as in training data.

Table 6. Performance of our system and the top-ranked systems in terms of F1-score(%). The numbers in the brackets are the official rankings.

Team ID	$\mathbf{F_{Syn}}$	$\mathbf{F_{Ant}}$	$\mathbf{F_{Hyp}}$	$\mathbf{F_{Mer}}$	$\mathbf{F_{macro}}$
Ours	51.9	65.1	68.6	73.3	64.7 (3)
CASIA	90.3	88.3	81.7	83.1	85.9 (1)
Tongji_CU-KG	73.2	78.1	78.8	77.1	76.8 (2)

4 Conclusion

In this paper, we extract several surface features, word embedding and character embedding features from word pairs and adopt supervised machine learning algorithms to perform context-free Chinese word semantic relation classification.

The system performance ranks above average. In future work, we consider to collect more external semantic dictionary and web resources to expand the training set as well as capture semantic information between two Chinese words.

Acknowledgements

This research is supported by grants from NSFC (61402175), Science and Technology Commission of Shanghai Municipality (14DZ2260800 and 15ZR1410700), Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213) and Duty Collection Center (Shanghai) of the General Administration of Customs.

References

1. Yunfang Wu and Minghua Zhang. Overview of the nlpcc 2017 shared task: Chinese word semantic relation classification. In *the 6th Conference on Natural Language Processing and Chinese Computing*, Dalian, China, 8-12 November 2017.
2. Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics, 2009.
3. Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. Task-oriented learning of word embeddings for semantic relation classification. *arXiv preprint arXiv:1503.00095*, 2015.
4. Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*, 2015.
5. Vivian S Silva, Manuela Hürliman, Brian Davis, Siegfried Handschuh, and André Freitas. Semantic relation classification: Task formalisation and refinement. *COLING 2016*, page 30, 2016.
6. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.
7. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.
8. Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.
9. Shaoru Guo, Yong Guan, Ru Li, and Qi Zhang. Chinese word similarity computing based on combination strategy. In *NLPCC/ICCPOL*, pages 744–752, 2016.
10. Jiahuan Pei, Cong Zhang, Degen Huang, and Jianjun Ma. Combining word embedding and semantic lexicon for chinese word similarity computation. In *NLPCC/ICCPOL*, pages 766–777, 2016.

11. Jiang Zhao, Tiantian Zhu, and Man Lan. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *SemEval@ COLING*, pages 271–277, 2014.
12. Xipeng Qiu, Jingjing Gong, and Xuanjing Huang. Overview of the nlpcc 2017 shared task: Chinese news headline categorization. In *arXiv:1706.02883v1*, 2017.
13. Mei Jiaju, Zhu Yiming, Gao Yunqi, and Yin Hong-Xiang. Tongyici cilin. *ShangHai Dictionary Publication*, 1983.
14. Zhendong Dong, Qiang Dong, and Changling Hao. Hownet and the computation of meaning. 2006.