QLUT at SemEval-2017 Task 2: Word Similarity Based on Word Embedding and Knowledge Base

Fanqing Meng[1], **Wenpeng Lu**[*1], **Yuteng Zhang**[1], **Ping Jian**[2], **Shumin Shi**[2], **Heyan Huang**[2]
[1]School of Information, QiLu University of Technology, Jinan, Shandong, China
[2]School of Computer, Beijing Institute of Technology, Beijing, China
mengfanqing678@163.com, lwp@qlu.edu.cn, zhangyuteng1029@163.com,
pjian@bit.edu.cn, bjssm@bit.edu.cn, hhy63@bit.edu.cn

Abstract

This paper shows the details of our system submissions in the task 2 of SemEval 2017. We take part in the subtask 1 of this task, which is an English monolingual subtask. This task is designed to evaluate the semantic word similarity of two linguistic items. The results of runs are assessed by standard Pearson and Spearman correlation, contrast with official gold standard set. The best performance of our runs is 0.781 (Final). The techniques of our runs mainly make use of the word embeddings and the knowledge-based method. The results demonstrate that the combined method is effective for the computation of word similarity, while the word embeddings and the knowledge-based technique, respectively, needs more deeply improvement in details.

1 Introduction

Semantic word similarity aims at measuring the extent to which two words are similar (Camacho-Collados et al., 2017). Given two words, the runs in this competition should give a score which indicates the similarity between them, and it will be evaluated by the official gold standard set. This task doesn't offer any annotated corpus and the organizers encourage systems to utilize unlabeled corpus. With the development of word embeddings technique, more and more attentions are paid to it (Mikolov et al., 2013a; Mikolov et al., 2013b). We also adopt the word embeddings method in our runs.

Besides the word embeddings method, another knowledge-based method is proposed by us, which is based on BabelNet (Navigli and Ponzetto, 2012). Integrating Wikipedia and WordNet, BabelNet is a multilingual encyclopedic and lexicographic knowledge base, which builds an enormous semantic network linking concepts and named entities with the aid of a large semantic relations.

Based on the word embedding method and the knowledge-based method, a combined method is implemented, which achieves the best performance.

2 System Overview

In the subtask 1 (English monolingual word similarity) of this task, we have submitted two system runs, both of which are unsupervised. We mainly utilize the word embeddings method and the combined method.

The Figure 1 shows the framework of our system runs. In the top part of the figure, *word1* and *word2* are the input of our systems. Run1 utilizes the word embeddings method. Run2 utilizes the combined method, which is based on the word embeddings and knowledge-based method.

2.1 DataSet

Test Set: In this task, we submit our runs on the English monolingual word similarity dataset, which includes 500 word pairs. These word pairs

[*]Corresponding author

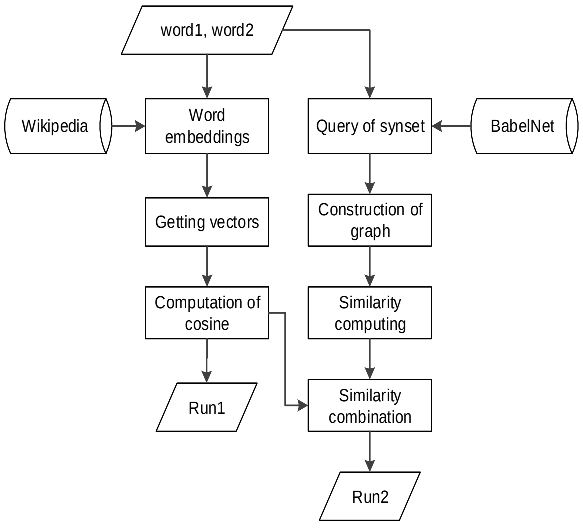

Figure 1: The framework of our system runs.

may be concepts or named entities, which are tab-separated.

Gold Standard Set: This set is gold standard set, which is annotated by official annotators. Each line in this set is a similarity value according to the test set describe above in [0-4] rating scale. 4 shows that the two words are very similar, i.e., synonyms; 3 means that the two words are similar, but have slightly different details; 2 represents that the two words are slightly similar, having a topic/domain/function and ideas or related concepts in common; 1 shows that the two words are dissimilar, which only having some small details in common. 0 means that the two words are totally dissimilar.

2.2 Word Embeddings Method

In this competition, we use the word2vec toolkit[1] to train word embeddings on the English Wikipedia corpus[2]. Before training word embedding, we preprocess the text file of the corpus to change its character encoding form from Unicode to UTF-8, because it is the default set to run the word2vec toolkit. We set the training window size to 5 and default dimensions to 200, and choose the Skip-gram model. After training on the corpus, word2vec toolkit generates a word embeddings file, in which each word in the Wikipedia corpus can be mapped to a word embedding of 200 dimensions. Each dimension of the word embedding is a double value.

[1] https://code.google.com/p/word2vec/

[2] https://sites.google.com/site/rmyeid/projects/polyglot

Word Similarity: Mikolov has explained that the word embedding has semantic meaning (Mikolov et al., 2013a). Therefore, given two words, the semantic word similarity can be easily attained by the cosine of their word embeddings:

$$sim_{vec}(\mathrm{w}_1, w_2) = \frac{Vec(w_1)\,Vec(w_2)}{|Vec(w_1)||Vec(w_2)|}, \tag{1}$$

where $vec(w_1)$ is the word embedding of word w_1 and, $|vec(w_1)|$ and $|vec(w_2)|$ are the length of $vec(w_1)$ and $vec(w_2)$, respectively.

Phrase Similarity: As Mikolov has presented that phrase vector can be easily gotten by simple vector addition (Mikolov et al., 2013b), we can gain the phrase similarity between two phrases as follows:

$$sim_{vec}(p_1, p_2) = \frac{\sum_{i=1}^{|p_1|} vec(w_i) \sum_{j=1}^{|p_2|} vec(w_j)}{\left|\sum_{i=1}^{|p_1|} vec(w_i)\right|\left|\sum_{j=1}^{|p_2|} vec(w_j)\right|}, \tag{2}$$

where $|p_1|$ and $|p_2|$ are the number of the words, which phrase p_1 and p_2 contain respectively. Word w_i represents the word, which belongs to p_1.

2.3 Knowledge-based Method

Thanks to the BabelNet, which provides a large coverage of concepts and named entities connected in a large semantic relations, such as synonymy, hypernymy and hyponymy, we can get the semantic relations between the two given words (each being a concept or named entity) by the BabelNet API[3]. In order to easily compute the similarity of two words, we implement the following algorithm.

Algorithm 1:

```
Input: word1, word2
Output: the semantic similarity between word1 and word2
Procedure:
1: if word1 (or word2) isn't found
2:    then sim = 0.5, return sim;
3: if word1 and word2 are synset
4:    then sim = 1.0;
5: else{
6:    search their related words;
7:    if the search steps step > γ
8:       then sim = 0.0;
9:    else{
10:      construct a graph;
11:      get the shortest path path;
12:      get the similarity sim;
13:    }
14: }
15: return sim;
```

[3] http://babelnet.org/download

Lines 1-2, we make the similarity of 0.5 according to the official suggestion if the systems can't cover the words in the evaluation data. Lines 3-4, if the two items of input are synset, then we assign 1.0 as its similarity. Lines 5-6, if the two words do not have this relationship, the program will iteratively search the related synsets of *word1* and *word2*, respectively, until they have common related synset(s) or the search steps *step* beyond a set threshold γ beforehand. Due to the large cost of the subsequent graph computation, we simply set 10 steps as the maximum iterative steps (i.e., γ). Lines 7-8, If the steps *step* is beyond γ, we consider that it may cost more than 10 steps to get the common synset which connect them in the graph, or even not get anything. In other words, the two words may be weakly similar, then we just simply set 0.0 as their similarity. Lines 9-14, if the steps *step* do not reach the threshold γ, we begin to construct the graph with *word1*, *word2* and their related synset by means of JUNG toolkit[4] and then traverse the graph to get the Dijkstra shortest path *path* between the input *word1* and *word2*. And we make the reciprocal of the *path* power of μ as their similarity *sim*:

$$sim = 1/(\mu^{path}), \qquad (3)$$

where *path* is the Dijkstra shortest path described above, and μ is set to 1.4 manually, which is used to adjust the similarity *sim* to be in a proper range (see 2.1). At last (line 15), it return the similarity *sim*.

2.4 Combined Method

This method is directly generated by combining the two methods described above, i.e., the word embeddings method and the knowledge-based method. We make this method, in order to leverage the performance of the two methods. More specially, we use the following equation to get the final similarity.

$$sim_{final} = \alpha * sim_{vec} + (1 - \alpha) * sim_{kb}, \qquad (4)$$

where sim_{kb} represents the semantic similarity of the knowledge-based method and sim_{vec} stands for the semantic similarity of the word embedding method. The parameter α is the manually factor for balancing the results of the two methods. And it is set to 0.6 manually. sim_{final} is the final result.

[4] http://jrtom.github.io/jung/

Runs	Pearson	Spearman	Final
Run1	0.669	0.673	0.671
Run2	**0.774**	**0.780**	**0.777**
NASARI	0.683	0.681	0.682

Table 1: Results of our runs and baseline.

μ	Pearson	Spearman	Final
1.0	-0.025	-0.020	-0.022
1.2	**0.653**	0.652	0.652
1.4	**0.653**	**0.656**	**0.654**
1.6	0.644	**0.656**	0.650
1.8	0.633	**0.656**	0.644
2.0	0.621	0655	0.637

Table 2: Results of Run_{kb} with various parameters.

α	Pearson	Spearman	Final
0.0	0.653	0.656	0.654
0.2	0.731	0.747	0.739
0.4	**0.777**	**0.786**	**0.781**
0.6	0.774	0.780	0.777
0.8	0.731	0.735	0.733
1.0	0.669	0.673	0.671

Table 3: Results of Run2 with various parameters.

3 Evaluation

Run1: This run uses the word embeddings method described in Section 2.2. Given two words or phrases, it can get the semantic similarity by computing the cosine between their word vectors.

Run2: This run use the combined method described in Section 2.4. It can leverage the word embeddings method and knowledge-based method.

Run_{kb}: This run use the knowledge-based method which is described in Section 2.3.

The runs are evaluated according to the measures of standard Pearson and Spearman correlation. The final score (see the last column in Table 1) is the harmonic mean of Pearson and Spearman correlations. NASARI in Table 1 (the last row) is the baseline system which is created by the official of this task.

As we can see in Table 1 that the system Run2 make a 9.5% (Final) improvement in contrast with the baseline system (NASARI), and achieves the best performance. The performance of the system Run1 does not exceed the baseline system. Table 2 shows that the system Run_{kb} get its best performance when μ is set to 1.4 (see 2.3). Table 3 shows that Run2 get its best performance when α is set to 0.4 instead of 0.6 (see 2.4). These results show that the word embeddings method and the knowledge-based method, respectively, are not enough effective while the combined method of them makes the best performance of 0.781 in all our runs.

4 Conclusions and Future Work

Our best run achieves the performance of 0.781 (Final). It shows that the combined method is more effective for the computation of word similarity than the word embeddings method and the knowledge-based method, respectively. There are a large room to improve the performance of the word embeddings method and the knowledge-based method. In the future, we will refine the various relations among words to improve knowledge-based method.

Acknowledgments

The work described in this paper is mainly supported by National Programs for Fundamental Research and Development of China (973 Program) under Grant 2013CB329303, and National Natural Science Foundation of China under Grant 61502259, 61671064 and 61202244.

References

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier and Roberto Navigli. 2017. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada, pages 15--26. http://www.aclweb.org/anthology/S17-2002.

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. https://arxiv.org/abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*. https://arxiv.org/abs/1310.4546.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217-250. http://dx.doi.org/10.1016/j.artint.2012.07.001.