Chinese Word Similarity Computation based on Automatically Acquired Knowledge

Yuteng Zhang[1], Wenpeng Lu[1] and Hao Wu[2]
[1] *School of Information, Qilu University of Technology*
[2] *School of Computer, Beijing Institute of Technology*
zhangyuteng1029@163.com, lwp@qlu.edu.cn, wuhao123@bit.edu.cn

Keywords: Similarity measures, word vectors, word similarity, PMI, Dice, Phi.

Abstract: This paper describes our methods for Chinese word similarity computation based on automatically acquired knowledge on NLPCC-ICCPOL 2016 Task 3. All of the methods utilize off-the-shelf tools and data, which makes them easy to be replicated. We use Sogou corpus to train word vector for Chinese words and utilize Baidu to get Web page counts for word pairs. Both word vector and Web page counts can be acquired automatically. All of our methods don't utilize any dictionary and manual-annotated knowledge, which avoids the huge human labor. Among the four submitted results, three systems achieve a similar Spearman correlation coefficient (0.327 by word vector, 0.328 by word vector and PMI, 0.314 by word vector and Dice). Besides, when all the English letters are converted to lowercase, the best performance of our methods is improved, which is 0.372 by word vector and Dice. All of the comparative methods and experiments are described in the paper.

1 INTRODUCTION

Word similarity computation is a fundamental technique for many applications in natural language processing(NLP), such as question answering (Surdeanu, Ciaramita and Zaragoza, 2011), textual entailment (Berant, Dagan and Goldberger, 2012), lexical simplification (Biran, Brody and Elhadad, 2011) and word sense disambiguation (Lu, Huang and Wu, 2014).

The traditional measures for word similarity can be divided into two categories: the methods based on dictionary and the methods based on corpus. The methods based on dictionary need to select a dictionary, for example WordNet or HowNet, as knowledge base. Then, there are multiple methods to measure the similarity based on the semantic taxonomy tree or multi-dimensional semantic description. The methods based on corpus utilize the cooccurrence statistical information to compute the similarity (Smadja, McKeown and Hatzivassiloglou, 1996) arity of word pairs, such as PMI (Church and Patrick, 2002), Dice, Phi (Gale and Church, 1991). Recently, with the development of deep learning in NLP, word vector has drawn more and more attention, which has been applied in many fields.

In shared task 3, we have submitted four system runs. One is based on word vector, one is based on word vector and PMI, one is based on word vector and Dice, the last is based on word vector and Phi. Besides, we have modified our systems to improve their performance. In the paper, we have shown another three system runs. The main difference with previous submitted four system runs lies that all of English capital letters are converted to lower case.

All of our system runs don't utilize any dictionary or other manual-annotated knowledge. Our original intention is to find an effective method to compute word similarity based on automatically acquired knowledge. Especially, the ability of word vector is focused by us.

2 RELATED WORKS

As a traditional problem in NLP, word similarity has attracted substantial interests in the research community. Many similarity measures have been proposed. The traditional measures for word similarity can be divided into two categories: the methods based on dictionary and the methods based on corpus.

Wu H., Lu W. and Zhang Y.
Chinese Word Similarity Computation based on Automatically Acquired Knowledge.
DOI: 10.5220/0006443500480052
In *ISME 2016 - Information Science and Management Engineering IV (ISME 2016)*, pages 48-52
ISBN: 978-989-758-208-0

Chinese Word Similarity Computation based on Automatically Acquired Knowledge

The methods based on dictionary measure the similarity with the semantic taxonomy tree or multi-dimensional semantic description. Based on WordNet, Pederson et al. have developed a similarity tools, which supports the measures of Resnik, Lin, Jiang-Conrath, Leacock-Chodorow, Hirst-St.Onge, Wu-Palmer, Banerjee-Pedersen, and Patwardhan-Pedersen (Pilevar, Jurgens and Navigli, 2013). Based on Tongyici Cilin, Wang has proposed a method to measure similarities between Chinese words on semantic taxonomy tree (Wang, 1999). Based on HowNet, Liu et al. have proposed to compute Chinese word similarity on the multi-dimensional knowledge description (Liu and Li, 2002) Based on WordNet, Pilehvar et al. have presented a unified approach to compute semantic similarity from words to texts, which utilizes personalized Pagerank on WordNet to get a semantic signature for each word and compares the similarity of semantic signatures of word pairs (Pilevar, Jurgens and Navigli, 2013). For the methods based on dictionary, a reliable dictionary with high quality is difficult to build, which is a hard work and needs lots of labors. With social development, many new words will emerge, which usually are missed in the dictionaries. This will affect the performance of word similarity computation based on dictionary.

The methods based on corpus utilize the co-occurrence statistical information to compute the similarity of word pairs. Lin et al. have presented an information-theoretic definition of similarity, which measures similarities between words based on their distribution in a database of dependency triples (Lin, 1998). Liu et al. have proposed to measure indirect association of bilingual words with four common methods, that is, PMI, Dice, Phi and LLR (Liu and Zhao, 2010). With the development of deep learning, a distributed representation for words, that is word vector, has been proposed by Bengio et al (Bengio et al., 2003). A word vector is trained on a large scale corpus. With word vector, it is easy to compute the similarity of words. For the methods based on corpus, though their performances are affected by the size and quality of corpus, the methods can save lots of human labor and can append new words at any time.

For the convenience of acquire knowledge automatically, we focus on the methods based on corpus, especially the method based on word vector. A series of experiments has been done to compare their performance.

3 SUBMITTED AND REVISED SYSTEMS OVERVIEW

In NLPCC-ICCPOL 2016 Task 3, we have submitted four system runs. All of them are unsupervised systems.

3.1 Submitted System

Run 1: Word Vector Method.
In this run, we use a word vector obtained by word2vec (Mikolov et al., 2013). Using these word vector representations, the similarity between two words can be computed with the cosine operation. It is advisable to keep all the given values.

We train word vector by running the word2vec toolkit (Mikolov et al., 2013). Sogou news corpus is selected as train corpus, which contains news data on Internet from June to July in 2012. The news corpus is formatted and cleaned. HTML marks are removed and only news text is reserved. The news text is done Chinese word segment by ICTCLAS 2016. Word2vec runs on the preprocessed news corpus to train an effective Chinese word vector. In the run, CBOW model is selected, window size is set to 5 and dimension of word vector is set to 200.

The similarity between a pair of words is computed with the cosine distance of their associated word vectors, as is shown in Equation (1).

$$Similar(w_1, w_2) = WordVector(w_1, w_2) = \cos(vector(w_1), vector(w_2)) \times 10 \quad (1)$$

In which, w_1 and w_2 are the target word pair, $vector(w_1)$ and $vector(w_2)$ are their word vectors.

Run2: Word Vector and PMI method.
In the experiment of Run1, we find that there are some missing words by word vector, such as GDP, GRE, WTO. The similarity of word pairs that involve the missing words are 0 in Run1. This is not reasonable. There should be a supplement for the missing words. Therefore, in Run2, we take PMI method as the backup of word vector. That is to say, the missing words by word vector would be processed by PMI.

For illustration purposes, supposing that two words that need to calculate the similarity are w_1 and w_2. We introduce the following relations for each word pair (w_1, w_2).

$a=freq(w_1,w_2)$: The number of web pages that contain both w_1 and w_2.

$b=freq(w_1)-freq(w_1,w_2)$: The number of web pages that contain the w_1 and don't contain the w_2.

49

$c=freq(w_2)-freq(w_1,w_2)$: The number of web pages that contain the w_2 and don't contain the w_1.

$d=N-a-c$: The number of web pages that don't contain the word w_1 and the w_2.

In which, N is the size of web pages on Internet, which is assumed to 10^{11}. a, b, c, d are obtained with Baidu search engine.

PMI is computed with Equation (2).

$$PMI(w_1,w_2)=\log\frac{N\times freq(w_1,w_2)}{freq(w_1)\times freq(w_2)}=\frac{N\times a}{(a+b)\times(a+c)} \quad (2)$$

The similarity computed by PMI method doesn't lies in the range of 0~10. We utilize two normalizations to map the similarity to the range of 0~10. In first normalization, a min-max normalization is done with Equation (3). Then, for the results of first normalization, we set a threshold value, which is 1. All the results which are higher than the threshold, are normalized to 10; the other values are normalized with Equation (3) again.

$$x^*=\frac{x-x_{\min}}{x_{\max}-x_{\min}}\times 10 \quad (3)$$

The similarity of word vector and PMI method in Run2, can be represented with Equation (4).

$$Similar(w_1,w_2)=\begin{cases}WordVector(w_1,w_2),\\(WordVector(w_1,w_2)>0)\\Normalize(PMI(w_1,w_2)),\\(WordVector(w_1,w_2)\le 0)\end{cases} \quad (4)$$

Run3: Word Vector and Dice method.

Different with Run2, we take Dice method as the backup of word vector in Run3. The detailed computational method is similar with Run2. Only the calculation of Dice is different, which is computed with Equation (5).

$$Dice(w_1,w_2)=\frac{2\,freq(w_1,w_2)}{freq(w_1)+freq(w_2)}=\frac{2a}{(a+b)+(a+c)}=\frac{2a}{2a+b+c} \quad (5)$$

We also use the normalization method in Run2 to deal with the results of Dice. The similarity of word vector and Dice method in Run3, can be represented with Equation (6).

$$Similar(w_1,w_2)=\begin{cases}WordVector(w_1,w_2),\\(WordVector(w_1,w_2)>0)\\Normalize(Dice(w_1,w_2)),\\(WordVector(w_1,w_2)\le 0)\end{cases} \quad (6)$$

Run4: Word Vector and Phi method.

Different with Run2 and Run3, we take Phi method as the backup of word vector in Run4. The detailed computational method is similar with Run2 and Run3. Only the calculation formula of Phi is different, which is computed with Equation (7).

$$Phi(w_1,w_2)=\frac{(ab-bc)^2}{(a+b)(a+c)(b+d)(c+d)} \quad (7)$$

We also use the normalization method in Run2 and Run3 to deal with the results of Phi. The similarity of word vector and Phi method in Run4, can be represented with Equation (8).

$$Similar(w_1,w_2)=\begin{cases}WordVector(w_1,w_2),\\(WordVector(w_1,w_2)>0)\\Normalize(Phi(w_1,w_2)),\\(WordVector(w_1,w_2)\le 0)\end{cases} \quad (8)$$

3.2 The Systems based on Web Page Counts

In order to compare the effectiveness of word vector and the methods based on web page counts, we separately utilize PMI, Dice and Phi methods to compute word similarity. PMI, Dice and Phi is computed with Equation (2), (5), (7). All of them are normalized with similar method in Run2~Run4. The normalized PMI, Dice and Phi are returned as word similarity.

3.3 Revised Systems

When analyzing the results of vector word in Run1, we find that some words, such as GDP, WTO and GRE, can be identified by vector word after the English letters are converted to lower case. Therefore, we revise the submitted system by converting the English words to lower case. After the little trick, we recomputed the similarity of word pair with word vector.

4 EVALUATION

The evaluation metric is Spearman's rank correlation coefficient (SRCC) between system output and the gold standard, which is shown as Equation (9).

$$r_R=1-\frac{6\sum_{i=1}^{n}(R_{Xi}-R_{Yi})^2}{n(n^2-1)} \quad (9)$$

Where n is the number of observations, R_{Xi} and R_{Yi} are the standard deviations of the rank variables.

Chinese Word Similarity Computation based on Automatically Acquired Knowledge

4.1 NLPCC2016 Results

Performances of all of our systems on NLPCC 2016 task 3 are showed in Table 1 and Fig.1. From them, we can make the following observations:

(1) Among the four submitted systems, the method of word vector and PMI(Run2) has achieved the best performance, which is 0.328. However, its advantage is very weak. There is only a little gap of 0.001 between Run2 and Run1.
(2) Among the systems based on Web page counts, the method of PMI is best, whose spearman correlation is 0.329. Though the method is simple, its performance has surpassed the method of word vector in Run1.
(3) Among the revised systems, the best performance is achieved by the method of word vector and Dice, which is 0.372.
(4) Comparing the submitted systems and revised systems, it is obvious that word vector is case-sensitive. Though only a little trick of converting capital English letters to lower case is utilized in the revised systems, the performance of each system is improved greatly, which is surprised. For the applications with different purposes, we should process the capital or lowercase letters carefully.
(5) Comparing the revised systems, when PMI, Dice or Phi is used as a backup strategy, the performances of all of them are improved. Since the vocabulary of word vector is limited, it is impossible for word vector to include new words. To take PMI as a backup is an effective method to solve the disadvantage of word vector.
(6) Comparing the submitted systems and systems based on Web page counts, we find that the performance of PMI is better than that of word vector, which is surprised. The reasons for this may be double. On one hand, word vector in Run1 fails to process the capital letters. As is shown in revised systems, once the capital letters are converted to lower case, the performance of word vector can be improved greatly. On the other hand, the corpus to train word vector is a Sogou news corpus. However, the evaluation data set is selected from news articles and Weibo text. The Sogou news corpus fails to meet with the evaluation data set. If another Weibo corpus is used to train word vector, its performance may be improved.

Table 1. Performances of our systems on NLPCC 2016 share task 3.

	Method	***SRCC***
Submitted Systems	*Word Vector*	0.327
	Word Vector + PMI	**0.328**
	Word Vector + Dice	0.314
	Word Vector + Phi	0.234
Systems Based on Web Page Counts	*PMI*	**0.329**
	Dice	0.221
	Phi	0.280
Revised Systems	*Word Vector*	0.359
	Word Vector + PMI	0.361
	Word Vector + Dice	**0.372**
	Word Vector + Phi	0.368

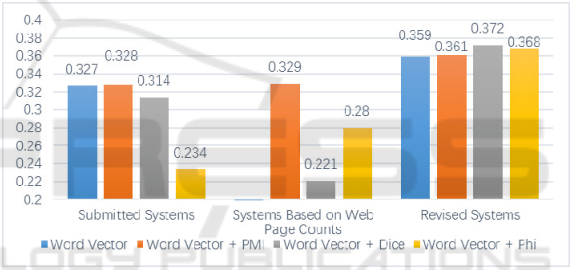

Fig. 1. Comparison of all of our systems

5 CONCLUSIONS AND FUTURE WORK

Besides the four system submitted in NLPCC 2016 task 3, the paper describes the methods based on Web page counts and the revised methods in detail. The performances of word vector, PMI, Dice and Phi are compared carefully. Among all of the systems, the revised method of word vector and Phi has achieved best performance. We observe that it is important to process the capital and lowercase letters for word vector. Besides, to take PMI method as a backup of word vector is an effective way to improve the performance.

Future works are twofold. On one hand, the method of normalizing the output of PMI, Dice and Phi is simple and stiff. We would consider to normalize them with Gauss regression. On the other hand, because current corpus is Sogou news corpus,

51

which is mismatched with the evaluation dataset, we would try to supply some Weibo corpus to train word vector. This may improve the performance of word vector.

ACKNOWLEDGEMENTS

The research work is partially supported by National Natural Science Foundation of China (61502259, 61202244), Natural Science Foundation of Shandong Province (ZR2011FQ038) and the Project of Shandong Province Higher Educational Science and Technology Program (J12LN09, J10LG20). Wenpeng Lu is the corresponding author.

REFERENCES

Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3, pp.1137-1155.

Berant, J., Dagan, I. and Goldberger, J. (2012). Learning Entailment Relations by Global Graph Structure Optimization. *Computational Linguistics*, 38(1), pp.73-111.

Biran, O., Brody, s. and Elhadad, N. (2011). Putting it simply: a context-aware approach to lexical simplification. In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. pp.496-501.

Church, K. and Patrick, H. (2002). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1), pp.22-29.

Gale, W. and Church, K. (1991). Identifying Word Correspondences in Parallel Texts. In: *Speech and Natural Language, Proceedings of a Workshop held at Pacific Grove*. pp.19-22.

Lin, D. (1998). An Information-Theoretic Definition of Similarity. In: *Fifteenth International Conference on Machine Learning*. pp.296-304.

Liu, P. and Zhao, T. (2010). Unsupervised Translation Disambiguation Based on Web Indirect Association of Bilingual Word. *Journal of Software*, 21(4), pp.575-585.

Liu, Q. and Li, S. (2002). Word Similarity Computing Based on How-net. *Computational linguistics in Chinese*.

Lu, W., Huang, H. and Wu, H. (2014). Word sense disambiguation with graph model based on domain knowledge. *Acta Automatica Sinica*, 40(12), pp.2836-2850.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv：1301.3781*.

Pilevar, M., Jurgens, D. and Navigli, R. (2013). Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In: *Association for Computational Linguistics*. pp.1341-1351.

Smadja, F., McKeown, K. and Hatzivassiloglou, V. (1996). Translating collocations for bilingual lexicons: a statistical approach. *Computational Linguistics*, 22(1), pp.1-38.

Surdeanu, M., Ciaramita, M. and Zaragoza, H. (2011). Learning to Rank Answers to Non-Factoid Questions from Web Collections. *Computational Linguistics*, 37(2), pp.351-383.

Wang, B. (1999). *Research on automatic alignment for Chinese-English bilingual corpus*. Doctor. Graduate University of Chinese Academy of Sciences(Institute of Computer Technology).