International Journal of Multimedia and Ubiquitous Engineering
Vol.15, No.2, 2020, pp.31-40
http://dx.doi.org/10.21742/ijmue.2020.15.2.03

A Chinese Word Similarity Model with Pronunciation, Radical and Semantic Embedding

Ruiming Xiao[1], Leilei Kong[2*], Zhongyuan Han[3], Xu Sun[4]

[1]*Harbin University of Science and Technology, Harbin, China*
China United Network Communications Group Co., Ltd., Huizhou, China
[2,3]*Foshan University, Foshan, China*
[4] *Heilongjiang Institute of Technology, Harbin, China*
[2*]*kongleilei@fosu.edu.cn*

Abstract

Aiming at the problem that the existing Chinese word similarity calculation research does not make full use of the three major factors of Chinese characters: pronunciation, radical and semantic, this paper proposes a Chinese Word Similarity Model with Pronunciation, Radical and Semantic Embedding. This model uses the distributed representation to learn the pronunciation embeddings, radical embeddings and semantic embeddings of Chinese characters or words, and then interactively measures the semantic similarity of these Chinese factors. Finally, it uses the ridge regression to fuse these similarities to obtain the similarity of the words. The experimental results on Word-Sim297 corpus show the effectiveness of the proposed model.

Keywords: *Chinese Word Similarity, Pronunciation, Radical, Semantic, Embedding*

1. Introduction

The semantics similarity of words indicates the similarity of words on the aspect of semantical contents(meaning) in a quantitative way [1]. Semantic similarity is the important basis for the tasks of natural language processing, such as digging word relations, text paraphrase and machine translation.

Early the words usually were represented by using discrete distributed methods. The most common way is to represent the words as a multi-dimensional vector. The vector has a dimensionality of the size of the vocabulary. Then, the dimension of the word that occurs is set to 1 and the rest dimensions are set to 0. The methods of distributed representation of language are not capable of modeling semantics for the problem known as vocabulary gap: any pair of words is independent and the semantic distance can not be measured. What's more, these types of methods have no abilities to represent the words with multi-meanings.

To represent the meaning of the words, researchers have constructed the dictionaries and knowledge bases such as WordNet and HowNet. By using route length and concept depth of the concept nodes in the dictionaries, measuring the semantic distances between words is made possible. However, there are two shortcomings in these methods. First, the resources of these dictionaries are limited and deficient. Not all of the languages obtain their semantic dictionaries or knowledge base. Second, the construction of the semantics dictionaries are commonly add

Article history:

Received (July 4, 2020), Review Result (August 12, 2020), Accepted (September 20, 2020)

Print ISSN: 1975-0080, eISSN: 2652-1954 IJMUE

the subjective tendency and timely updates would be unrealistic.

In recent years, deep learning made a great contribution on the explosive advance of semantic representation. Deep learning represents words as a vector on a continuous space by using unsupervised learning algorithms. This kind of representation is called word-embedding. The method of word-embedding results in a revolutionary change on the calculation and representation of words [2], allowing us to compute semantic relevance between words using word-embeddings. For example, a simple vector calculation can be used for word analogy. Its results reflects relations such as v(king) – v(man) + v(woman) ≈ v(queen), v(Paris) – v(France) + v(Italy) ≈ v(Rome), in which v(w) representing the word-embedding for word w. The validation of word-embedding has been tested in calculations for synonym, near-synonym, and semantic relations.

Methods based on word embeddings usually consider words(or characters) as its basic units. This processing method showed its priority not only on English word similarity tests, but also did well in Chinese information processing [3][4][5].

However, resent methods [6][7][8] ignores an important aspect, that is, Chinese and English have a fundamental difference on the form of words. In English, words are the basic semantic units. In the contrast, the formation of Chinese words has many features and more complicated semantic structures. In Chinese, words are made up of pronunciation, glyph, and meaning. Pronunciation act as the links between characters and language. The glyph of Chinese character is made up of the radical that indicates pronunciation and meaning. However, the information is not fully taken advantage of in today's method.

Taking pronunciation, radical and semantic into account, this paper proposes a Chinese Word Similarity Model with Pronunciation, Radical and Semantic Embedding(CWSM-PRSE). We represent the Chinese words using three forms, the pronunciation, the radical and the semantic and learn their embedding on continuous space individually. Then, interactively computes the semantic similarity between these forming factors. At last, the ridge regression model is used to integrate these similarities to get the final similarity. The experiments on WordSim347 shows the better performance of the proposed model.

2. Word Similarity calculation model based on Pronunciation, Radical and Semantic

Given a training set $W_{train} = \{(w_i, w_j, s_{i,j})\}_{i=1..n, j=1..n}, s_{i,j} \in R$ to represent the similarity score between w_i and w_j. The purpose of the model is to learn the similarity model *f*. The goal is to train a model *f* that can predict the similarity score $s'_{i,j}$ for any word pair (w'_i, w'_j). In this paper, we use the similarity calculation model for Chinese using fused pronunciation, radical and semantic to learn the model *f*. The model has two parts. The first part is to get the vector from these three features. The second part is to learn a set of weights with ridge regression model. The model is described as [Figure 1].

2.1. The Semantic Features of the pronunciation, radical and semantic

For applying pronunciation, radical and semantic to compute the similarity of words, CWSM-PRSE firstly transforms the Chinese words into the corresponding pronunciation and the radical respectively. Then, for the same text datasets, there are three different datasets: the pronunciation datasets, the radical datasets and the original words datasets. Based on these three datasets, CWSM-PRSE learns the pronunciation embedding representation, the radical embedding representation and the word embedding representation for each given word or character using CBOW methods [9].

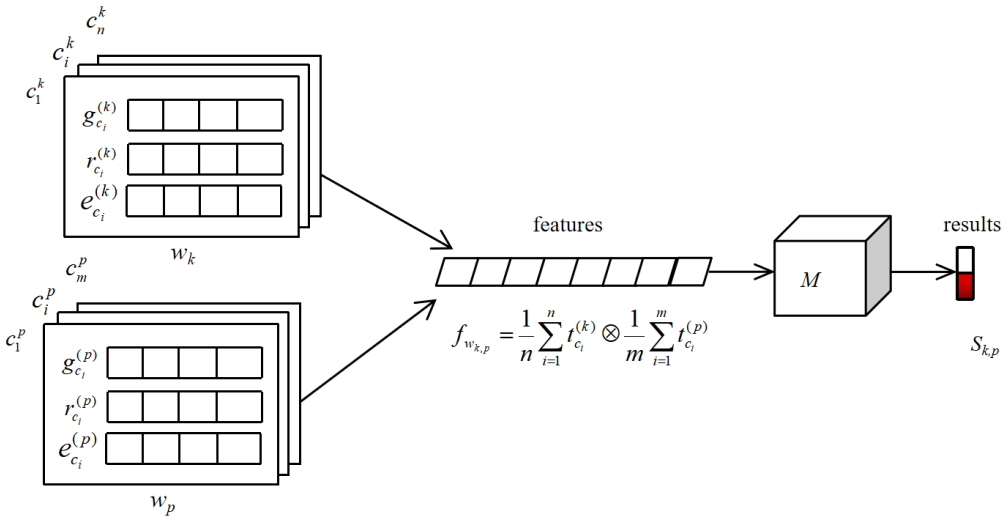

Figure 1. Model of word similarity based on pronunciation, radical and semantic

Composing these embeddings, we get six types of embeddings: words-segmented, character-segmented, radicals-segmented words, radicals-segmented character, pronunciation-segmented words and pronunciation-segmented character.

The above process of obtaining semantic similarity features by using the six vectors obtained from the three elements of Chinese characters can be formalized as follows. Assuming that the word w_k is expressed as: $w_k = \{c_1^k, \cdots c_i^k, \cdots c_n^k\}$, where c_i^k represents the i-th character of w_k. $g_{c_i}^{(k)}$ is the pronunciation character vector of c_i^k, $r_{c_i}^{(k)}$ is the radical character vector of c_i^k, $e_{c_i}^{(k)}$ represents the semantic character vectors. Then the words similarity feature $f_{w_{k,p}}$ can be expressed as:

$$f_{w_{k,p}} = \frac{1}{n}\sum_{i=1}^{n} t_{c_i}^{k} \otimes \frac{1}{n}\sum_{i=1}^{n} t_{c_i}^{p} \tag{1}$$

where $t_{c_i}^k$ is one of three vectors corresponding to pronunciation character vector, radical character vector and semantic character for character c_i^k. And the operator $\otimes$ is the notation of cosine or inner product of a vector v_{w_k} and v_{w_p}. The formulas for $\cos(A, B)$ and inner product $f(A, B)$ are in Eq.(2) and (3):

$$\cos(A,B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{2}$$

$$f(A,B) = A \bullet B = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \tag{3}$$

where A and B are the same kind of embedding vectors corresponding to W_k and W_p, $A = (a_1, a_2, \cdots a_n)$, $B = (b_1, b_2, \cdots b_n)$. In this way, two kinds of words similarity features are obtained (cosine and inner product). Computing the cosine and inner product using three

character vectors gained above, the corresponding six semantic similarity features are obtained.

The way of using word vectors to obtain semantic similarity features of word pairs is the same as character vectors. The main difference is the way of getting word vectors. CWSM-PRSE implements the Chinese text segmentation words to obtain Chinese words. Using these Chinese words to learn the word embedding. Then the words similarity feature $f_{w_{k,p}}$ expressed as:

$$f_{w_{k,p}} = v_{w_k} \otimes v_{w_p} \tag{4}$$

Thus, the semantic similarity characteristics of six types of word pairs can be gained via the three word vectors obtained above. Ultimately, a total of 12-dimensional word pair semantic features are received.

2.2. Word Similarity Calculation Based on Ridge Regression Model

CWSM-PRSE exploits the Ridge Regression to learn the weights of features to predict the word similarity. Ridge Regression is a biased estimation regression method dedicated to collinearity data analysis, which is an improvement of the Least square estimation method. It discards the partial information and reduces the fitting accuracy through giving up the unbiased advantage of the ordinary least squares method.

A multilinear regression model can be represented as:

$$Y = X\beta + \varepsilon \tag{5}$$

where Y is the dependent variable, X is the independent variable, β is the regression coefficient and ε is the error. In Ridge Regression method, the regression coefficient β can be computed as:

$$\beta = \left(X^T X + kI\right)^{-1} X^T Y \tag{6}$$

where k is the parameter of Ridge Regression and I is an identity matrix.

3. Experiments

3.1. Dataset

In this experiment, the WordSim-347 [10] is chosen as the experimental dataset. WordSim-347 is a word similarity dataset, which was produced by WordSim-353 [11] as a raw dataset. The partial data of WordSim-297 are shown in [Table 1].

Table 1. Partial data of WordSim-297

Word 1	Word 2	Relativity
Admission ticket	Tickets	4.59
Street	Block	4.04
Environment	Ecology	3.64
Apoplexy	Hospital	3.37
Boxing	Round	2.97

Cup	Substance	2.05
Cock	Voyage	0.32
Noon	Cord	0.06

3.2. Evaluating Indicator

The Spearman correlation coefficient [12] was usually adopted to evaluate the correlation between the model prediction results with the human rating. The Spearman correlation coefficient is defined as:

$$\rho = 1 - \frac{6\sum_{i=1}^{n}(R_{X_i} - R_{Y_i})^2}{n \times (n^2 - 1)} \quad (7)$$

where n represents the number of word pairs predicted by the model, and for a word pair w_{z_i}, R_{X_i} and R_{Y_i} are the similarity score predicted by the model and by the human beings respectively.

3.3. Experimental setup

3.3.1. Preprocessing

For the dataset of training corpus, we use jieba word segmentation toolkit to process the original dataset to obtain the dataset segmented by words, and use the pypronunciation toolkit to process the words-segmented dataset and the word-segmented dataset obtained above. Use the Radical toolkit, the words-segmented dataset and the word-segmented dataset are used to obtain the radicals-words-segmented dataset and the radicals-word-segmented dataset. In this way, we get six datasets for the training vector.

3.3.2. Vector learning

The dataset used for the embedding vector learning is composed of the Chinese Wikipedia dataset on June 20, 2018, the news dataset released by Sogou Lab(SougouCA) on August 16, 2012 , and the dataset evaluated by CCIR 2018. The dataset covers almost all aspects of knowledge, which provides a guarantee for obtaining rich semantic information vectors. In this experiment, the CBOW and Skip-Gram models provided by the open-source third-party Python toolkit Gensim are used as the learning algorithm for getting the embedding vector, and the CBOW model is used to train the six preprocessed datasets to obtain six vectors, and Skip-Gram model trains the dataset by words-segmented to obtain word vectors.

3.3.3. Feature normalization

To eliminate the influence of the large feature value on the prediction model, we use the feature normalization on the feature data. The normalization method used in this article is min-max standardization:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (8)$$

3.3.4. Parameter settings

In this test, three models are used, namely CBOW, Skip-Gram and Ridge Regression model. The parameter settings of these three models are shown in [Table 2].

Table 2. Parameter settings

No.	Model	Parameter settings
1	CBOW	sg=0, size=300, window=5, negative=5, iter=5, alpha=0.015, min_count=7
2	Skip-Gram	sg=1, size=300, window=5, negative=4, iter=5, alpha=0.015, min_count=5
3	Ridge Regression model	alphas=0.167, fit_intercept='True', normalize='False', copy_X='False', max_iter=910, solver='sag', tol=2.015

3.3.5. Baselines

To prove the validity of the model, several baseline methods are chosen to compare with the proposed model. They are Chinese words-segmented (CWS-CBOW), Chinese words-segmented based on Skip-Gram (CWS-Skip-Gram), Pronunciation words-segmented based on CBOW (PWS-CBOW), Radicals words-segmented based on CBOW (RWS-CBOW), Chinese character-segmented based on CBOW (CCS-CBOW), Pronunciation character-segmented based on CBOW (PCS-CBOW) and Radical character-segmented based on CBOW (RCS-CBOW). [Table 3] lists the details of these baselines.

For the baselines, we compute the cosine similarity of the two vectors to get the semantics similarity of word pairs.

Table 3. Parameter settings

Model	Pronunciation	Radicals	Semantics	words-segmented	character-segmented	CBOW	Skip-Gram
CWS-CBOW			√	√		√	
CWS-Skip-Gram			√	√			√
PWS-CBOW	√			√		√	
RWS-CBOW		√		√		√	
CCS-CBOW			√		√	√	

PCS-CBOW	√				√	√	
RCS-CBOW		√			√	√	

3.4. Experimental results and analysis

[Table 4] shows the main comparison results of our experiments on on WordSim-297.

Table 4. WordSim-297's evaluation results

Model	Spearman coefficient
CWS-CBOW	0.6523
CWS-Skip-Gram	0.6070
PWS-CBOW	0.6051
RWS-CBOW	0.5195
CCS-CBOW	0.3954
PCS-CBOW	0.2793
RCS-CBOW	0.1832
CWSM-PRSE (our model)	**0.6635**

We observe that the proposed model CWSM-PRSE outperforms the baselines on Spearman coefficient. The experimental results show that CWSM-PRSE has a significant improvement over the approaches only relying on pronunciation, radical or semantic embeddings.

At Comparing pronunciation, radical or semantic embeddings approach, the experimental results show that semantic embedding performs better than the method using pronunciation or radical features. The performance achieved by the method based on radical embedding is the lowest than the other two approaches.

We also note that the performance of Chinese word segmentation based on CBOW is significantly higher than the one based on Skip-Gram and the performance based on words is better than that based on characters.

[Table 5] gives some examples to show the similarity score of each model on some word pairs predicted in WordSim-297.

Table 5. The similarity score of each model on some word pairs predicted in WordSim-297

Word pairs		CWSM-PRSE	CWS-CBOW	CCS-CBOW	RWS-CBOW	RCS-CBOW	PWS-CBOW	PCS-CBOW	Human score
美元(dollar)	日元(yen)	3.44	4.38	3.94	3.02	4.25	4.38	3.80	3.25
行星(planet)	星系(galaxy)	3.50	4.52	3.74	4.03	3.39	4.54	3.80	3.59
旅程(journey)	汽车(car)	1.75	2.63	2.91	2.46	3.26	2.64	2.68	2.64
和平(peace)	保险(insurance)	1.43	2.42	2.51	2.68	3.94	2.37	2.55	1.90
奖章(medal)	英勇(brave)	2.38	3.53	2.96	3.13	3.18	3.54	2.67	2.77
杯子(cup)	咖啡(coffee)	2.61	3.73	3.00	3.37	3.77	3.68	2.80	2.71

老虎(tiger)	哺乳动物(mammal)	2.37	3.41	2.31	3.47	2.58	3.46	2.39	3.46
中风(stroke)	医院(hospital)	2.20	3.19	2.48	3.23	3.21	3.12	2.40	3.38
计算机(computer)	软件(software)	2.96	4.01	2.92	3.78	3.60	3.95	2.77	3.53
钱(money)	财产(property)	2.59	3.51	2.71	3.07	3.49	3.09	2.37	3.40

From the results in [Table 5], the predicted similarity scores given by CWSM-PRSE are more close to the score given by human on “dollar” and “yen”, and “planet” and “galaxy”. The predicted similarity scores given by PWS-CBOW are more close to the human score on “journey” and “car” and “peace” and “insurance”. The predicted similarity scores given by PCS-CBOW are more close to the human score on “medal” and “brave” and “cup” and “coffee”. The predicted similarity scores given by RWS-CBOW are more close to the human score on “tiger” and “mammal” and “stroke” and “hospital”. The predicted similarity scores given by RCS-CBOW are more close to the human score on “computer” and “software” and “mone” and “property”.

The experimental results indicate that integrating pronunciation, radical and semantic are helpful for Chinese word similarity calculation.

4. Summary

Aiming at the problem that the existing Chinese word similarity calculation research does not make full use of the three elements of Chinese words pronunciation, radical and semantic, this paper proposes a similarity calculation model for Chinese words that combines pronunciation, radical and semantic, named A Chinese Word Similarity Model with Pronunciation, Radical and Semantic Embedding. Exploiting the distributed representation of words, CWSM-PRSE learns the embeddings of Chinese characters, radicals, and pronunciation, computes the semantic similarity of these Chinese character component, and finally use the ridge regression model to fuse these similarities to obtain the similarity of words. Through the analysis of the experimental part, we can see that the inherent characteristics of Chinese characters are fully utilized. At the same time, in the experiment of computing word similarity task, the proposed CWSM-PRSE is better than other baseline methods. This shows that it is necessary to consider using the three major elements of Chinese characters to compute the similarity of words.

Acknowledgements

This work is supported by National Social Science Fund of China (No.18BYY125).

References

[1] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky, “Montmain. semantic similarity from natural language and ontology analysis. synthesis lectures on human language technologies,” Morgan & Claypool Publishers, **(2015)**

[2] LeCun Y, Bengio Y, Hinton G, “Deep Learning,” Nature, vol. 521, no.7553, pp.436-444, **(2015)**

[3] Xu R, Chen T, Xia Y, et al., “Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification,” Cognitive Computation, vol.7, no.2, pp.226-240, **(2015)**

[4] Yu M, Gormley M R, Dredze M, and et al., “Combining Word Embeddings and Feature Embeddings for Fine-

grained Relation Extraction," Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.1374-1379, **(2015)**

[5] Zhou G, He T, Zhao J, et al., "Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering," Proceedings of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, pp.250-259, **(2015)**

[6] YanSong S S and JingLi Tencent A I. "Joint learning embeddings for Chinese words and their components via ladder structured networks," Proceedings of the Twenty-Seventh International Joint Conference on Artifificial Intelligence (IJCAI-18), pp.4375-4381, **(2018)**

[7] Kang R, Zhang H, Hao W, and et al. "Learning Chinese Word Embeddings With Words and Subcharacter N-Grams," vol.7, pp. 42987-42992, **(2019)**

[8] Cao S, Lu W, Zhou J, and et al. "cw2vec: Learning Chinese Word Embeddings with Stroke n-gram Information," Proceedings of AAAI, pp.5053-5061, **(2018)**

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space", ICLR Workshop, **(2013)**

[10] Jin P and Wu Y., "SemEval-2012 Task 4: Evaluating Chinese Word Similarity," Proceedings of Joint Conference on Lexical and Computational Semantics, pp.374-377, **(2012)**

[11] Finkelstein L, Gabrilovich E, Matias Y, and et al., "Placing search in context: The concept revisited," ACM Transactions on information systems, vol.20, no.1, pp.116-131, **(2002)**

[12] Hauke J, Kossowski T, "Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data," Quaestiones geographicae, vol.30, no.2, pp. 87-93, **(2011)**

This page is empty by intention.