Measuring Semantic Similarity between Words Based on Multiple Relational Information

Jianyong DUAN†,††a, Member, Yuwei WU†,††b, Mingli WU†,††c, and Hao WANG†,††d, Nonmembers

SUMMARY The similarity of words extracted from the rich text relation network is the main way to calculate the semantic similarity. Complex relational information and text content in Wikipedia website, Community Question Answering and social network, provide abundant corpus for semantic similarity calculation. However, most typical research only focused on single relationship. In this paper, we propose a semantic similarity calculation model which integrates multiple relational information, and map multiple relationship to the same semantic space through learning representing matrix and semantic matrix to improve the accuracy of semantic similarity calculation. In experiments, we confirm that the semantic calculation method which integrates many kinds of relationships can improve the accuracy of semantic calculation, compared with other semantic calculation methods.

key words: semantic similarity, representation learning, multiple-relation

1. Introduction

Semantic similarity of words is the base of many natural language processing (NLP) tasks [1], such as word sense disambiguation (WSD), named-entity recognition (NER), part-of-speech tagging (POST), and so on. The results of words semantic similarity calculation affect the progress of higher-level NLP tasks greatly. Establishing the appropriate semantic representation of natural language words and acquiring the similarity degree between words are the bases of the natural language processing task [2].

It is difficult to perfectly express the semantics of words in corpus through one method because of the diversity and complexity of natural language. Especially in the natural language processing tasks related to social networks, the results of the model application are difficult to reach the ideal state. Because texts are short and informal, which leads to the application of semantic models often encountering the problem of sparse features. Most semantic processing methods only use part content of the corpus information. In this paper we hope to improve the accuracy of semantic similarity calculation by proposing a method to fuse the results of multiple models [3]–[5].

2. Related Work

In the field of current similarity calculation of words, the main implementation methods can be divided into three categories.

Method based on text statistics

This method can obtain the semantic similarity information between words by directly measuring the statistical information of the words in the corpus [6]. The traditional method mainly uses co-occurrence [7] statistical features of the words in context [8] to measure the semantic similarity between words. The word representation obtained by this method are mostly sparse and high-dimensional.

Method based on relation networks

This method usually measures the semantic similarity of words by the relationship of words in relation network. The relation network can be built based on article hyperlinks and reposted comments. The common calculation strategies include counting the number of child nodes and the shortest distance between the two word nodes.

Method based on category relationships

This approach usually estimates the similarity between words by measuring some existing category relationships. The category relationships commonly used are: Wikipedia category relationship, WordNet category relationship, Cilin [9] and so on.

But in practice, these methods have some shortcomings in measuring the semantic similarity between words:

Sparse feature

Compared to the number of the relationship between words, the number of relationships in the corpus are very rare. In the actual process, there are usually a large number of words without co-occurrence corpus. In particular, when we use method based on relation network and the category, the information provided by the corpus is even more sparse.

Homogeneity

When calculating the semantic similarity of words, many corpora can be easily obtained, such as the context between the words, the link between the words, the category relationship of words. Most of the semantic similarity calculation methods only consider a single relationship, which makes their performance limited.

In this paper we propose a calculation method to address these problems, which can fuse multiple word similarity calculation method and consider a variety of...

Manuscript received March 25, 2019.
Manuscript revised August 6, 2019.
Manuscript publicized September 27, 2019.
†The authors are with College of Computer Science & Technology, North China University of Technology, Beijing, China.
††The authors are with the Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing, 100144 China.
a) E-mail: duan_jianyong@163.com
b) E-mail: wuyuwei1996@foxmail.com
c) E-mail: wuml@ncut.edu.cn
d) E-mail: wanghaioms@gmail.com
DOI: 10.1587/transinf.2019EDP7083
corpora [10]–[12]. In this paper, we learn the low-dimensional continuous vector representation of words, and use cosine similarity to measure the semantic similarity between words through various corpus relations and methods. The experimental results are evaluated by comparison with the manually annotated word similarities. In order to compare with a variety of methods, we use classic Wikipedia Chinese corpus in our experiment.

3. Fusion of Multiple Relationships

A rich text relation network, which consists of many association relationships between multiple entities, can be expressed as \( G = (\chi, \xi) \). \( \chi \) represents the set of all entities and \( \xi \) represents the set of all relationships. For Wikipedia, its entity part consists of three parts: the set of entries (denoted as \( E \)), the gather of categories (denoted as \( C \)) and the set of words (denoted as \( W \)), so \( \chi = E \cup C \cup W \). And its relationship also consists of three parts: the hyperlink relation between entries: \( L_E = \{ (e_i, e_j) | e_i, e_j \in E \} \), the classification relation that the entry belongs to \( L_C = \{ (c_i, e_j) | c_i \in C, e_j \in E \} \), the relationship that the word appears in the entry description text: \( L_W = \{ (w_i, e_j) | w_i \in W, e_j \in E \} \), so \( \xi = L_E \cup L_C \cup L_W \).

We hope to establish a unified K-dimensional semantic space \( H \) [13] and learn the vector representation of each entity in \( H \). Here we define \( E_H \in R^{(E)\times K} \), \( C_H \in R^{(C)\times K} \), and \( W_H \in R^{(W)\times K} \) as the vector representation matrix of entries, categories, and words. The corresponding vector to each entry \( e_i \in E \) is represented as the i line vector \( E_H \) of the matrix \( E_H \). Similarly, category \( c_i \in C \) corresponds to \( C_H \), word \( w_i \in W \) corresponds to \( W_H \). When we learn to get the vector representation \( E_H \) of all the entries, it can be measured by cosine similarity. The formula for measuring similarity between entries \( e_i, e_j \) is:

\[
{\text{sr}}(e_i, e_j) = \frac{\sum_{l=1}^{K} E_{Hl}^i \cdot E_{Hl}^j} {\sqrt{\sum_{l=1}^{K} E_{Hl}^2} \cdot \sqrt{\sum_{l=1}^{K} E_{Hl}^2}} \tag{1}
\]

Our goal is to estimate the value of \( E_H \) through the associative matrix \( X \in R^{(E)\times[E]} \) between entries. \( C_H \) and \( W_H \) will also be estimated at the same time during the learning process.

3.1 Semantic Representation Learning

In our model, \( E_H \) is obtained by estimating the associative matrices \( X \in R^{(E)\times[E]} \), \( Y \in R^{(C)\times[E]} \) and \( Z \in R^{(W)\times[E]} \). The associative matrix is derived from the associative relation \( L_E, L_C, L_W \), which is described in detail in the next section. The evaluation function of the estimated result uses the None Zero Square Loss (NZSL) with the regular term, the specific formula is:

\[
\sum_{(i,j) \in N_X} (X_{ij} - E_H E_L^T)^2 + \lambda(\|E_H\|^2_F + \|E_L\|^2_F) \tag{2}
\]

Among them, \( E_L \in R^{(E)\times K} \) is the semantic vector matrix of each entry, \( N_X = \{(i, j) : X_{ij} \neq 0\} \) is the nonzero subscript gather of the relational matrix \( X \), \( \lambda \) is regular factor, \( \| \cdot \|_F^2 \) is Frobenius norm.

In order to fuse the relationship between the entries and other entities, we use the dot product between the entity vectors to estimate \( C_H \) and \( W_H \). So, the overall evaluation function is:

\[
J = \sum_{(i,j) \in N_X} (X_{ij} - E_H E_L^T)^2 + \lambda(\|E_H\|^2_F + \|E_L\|^2_F) \\
+ \sum_{(i,j) \in N_Y} (Y_{ij} - C_H E_H^T)^2 + \gamma(\|C_H\|^2_F) \\
+ \sum_{(i,j) \in N_Z} (Z_{ij} - W_H E_H^T)^2 + \delta(\|W_H\|^2_F) \tag{3}
\]

\( N_Y = \{(i, j) : Y_{ij} \neq 0\} \), \( N_Z = \{(i, j) : Z_{ij} \neq 0\} \) are the nonzero subscript gather of the relational matrix \( Y, Z \) respectively. \( \lambda, \gamma, \delta \) are regular factors. During the experiment we take \( \lambda = 0.01, \gamma = \delta = 0.005 \).

In learning procedure, we only consider errors generated by one line in the matrix in each iteration. For example, when studying \( E_H \), for all nonzero elements \( (i, j) \in N_X \) in the matrix, we only update vector element \( E_{H_{i,j}} \) for each \( k \). And the local gradient of \( E_{H_{i,j}} \) is:

\[
\frac{\partial J}{\partial E_{H_{i,j}}} = -2(X_{ij} - E_H E_L^T)E_{H_{i,j}} + 2\frac{E_{H_{i,j}}}{N_{X_{ij}}} \tag{4}
\]

\( N_{X_{i,j}} \) refers to nonzero element in the \( i \)th line of matrix \( X \), and \( N_{X_{i,j}} \) refers to nonzero element in the \( j \)th column of matrix \( X \). For each selected element \( (i, j) \in N_X \), we do the following update to all available \( k \):

\[
\nabla E_{H_{i,j}} = E_{H_{i,j}} - \eta \cdot \frac{\partial J}{\partial E_{H_{i,j}}} \tag{5}
\]

\( \eta \) is the learning rate. We set the initial learning rate \( \eta \) to 0.01 in the experiment, and decrease it in each iteration, which stabilizes the learning result. The learning process of entire model can be expressed as the following steps:

1. Fill the matrix \( E_H \) with random initial values.
2. Take a pair of entities from the training data \( E_i \) and \( E_j \).
3. The entities \( E_i \) and \( E_j \) are mapped to vectors \( E_{H_i} \) and \( E_{H_j} \) by \( E_H \).
4. The weight prediction between entity \( E_i \) and \( E_j \) is estimated by \( E_{H_i} \) dot product \( E_{H_j} \).
5. The error \( J \) is obtained by using prediction weight subtract actual weight obtained from the relationship network.
6. The gradient of the parameters in the matrix \( E_H \) is calculated by the formula (1), and the parameters in \( E_H \) are corrected according to the gradient and the learning rate.
7. Repeat step 2 until the number of iterations is reached or the error is less than the expected value.

Similarly, we adopted a similar method to learning $E_H$, $E_L$, $C_H$, $W_H$. First, we give the random initial value to $E_H$, $E_L$, $C_H$, $W_H$, then $E_H$, $E_L$, $C_H$, $W_H$ are modified by the gradient times the learning rate, until the number of iterations is reached or the error is less than the expected value.

3.2 Relational Matrix Acquisition

When the model we need to combine has the ability of vectoring words, we can treat the model as a coding matrix directly, and fix the parameters, which can greatly reduce the complexity of the model, reduce the computing resources required for training and improve the training speed of the model. When we need to combine a model which does not have vector representation of words, we can obtain the relation matrix $X$ between words in the following way, so that the knowledge of the combined model can be used as part of a priori knowledge of fusion model training.

The relation matrix $X$ between words can be obtained by fusing the basis of the term relevance algorithm. First, the basic semantic similarity calculation method is converted into the similarity matrix form:

$$X_{ij} = f(E_i, E_j), \quad (i, j) \in N$$

$N$ is the word table, and $X$ is the matrix representing the similarity between words. Due to different similarity range of different methods, the similarity matrix needs to be standardized before fusion:

$$\bar{X}_{ij} = \frac{X_{ij}}{\sum_k X_{ik}}, \quad (i, j) \in N$$

When we need to fuse multiple basic methods, we take the geometric average of nonzero entries of multiple word correlation matrices.

$$\bar{X}_{ij} = \frac{\bar{X}_{ij}}{\sum_k \bar{X}_{ik}^2}, \quad (i, j) \in N_X, N_X = \{(i, j) | X_{ij} \neq 0\}$$

The relation matrix $Y$ between entry and category can be obtained through classified corpus. For category corpus:

$$L_C = \{(c_i, e_j) | c_i \in C, e_j \in E\}$$

$$Y_{ij} = \begin{cases} 1, & (c_i, e_j) \in L_C \\ 0, & (c_i, e_j) \notin L_C \end{cases}$$

The relation matrix $Z$ between word and entry can be obtained through classified corpus. For category corpus:

$$L_W = \{(w_i, e_j) | w_i \in C, e_j \in E\}$$

$$Y_{ij} = \begin{cases} 1, & (w_i, e_j) \in L_W \\ 0, & (w_i, e_j) \notin L_W \end{cases}$$

4. Experiments

In this paper, the following methods are used as the basic semantic relevancy network, and fusion experiments are performed on different combinations of them. The basic semantic network is used as baseline.

4.1 The Control Method

We use several classical models as baselines, including method based on text statistics, method based on relation networks, and method based on word category.

- **Information Content [14]**
  The semantic similarity between words should be judged by the number of common subcategories in the category of words, and the similarity of words was expressed as:

$$sr(w_1, w_2) = \max_{c_1, c_2} \left[-\log p(c)\right]$$

$c_1$, $c_2$ are the types of word $w_1$, $w_2$, $S(c_1, c_2)$ are common subcategories gather of $c_1$, $c_2$, and $p(c) = \frac{\sigma(c)}{\sigma}$.  

- **Shortest Path [15]**
  Semantic similarity between the words was estimated by measuring the shortest distance between the words in the relation graph, and the semantic similarity of the words was expressed as:

$$sr(w_1, w_2) = -\log \frac{\text{len}(w_1, w_2)}{2 \text{dep}}$$

$\text{len}(w_1, w_2)$ represents the shortest path length between words in the relation graph, and $\text{dep} = \max(\text{len}(w_1, w_j))$. Because the bidirectional path between words may not exist, or not the same in directed graph, the use of this formula does not have exchangeability, so $sr(w_1, w_2) \neq sr(w_2, w_1)$.

- **Information Distance [16]**
  It is similar to the shortest path, but information distance can deal with the problem of similarity formula exchangeability brought by the directed graph. The semantic similarity of words is expressed as:

$$sr(w_1, w_2) = -\log \frac{\max\{\text{len}(w_1, w_2), \text{len}(w_2, w_1)\}}{2 \text{dep}}$$

- **Skip-gram Model**
  The current word is used to predict the probability of the word appearing before it, and the farther distance from the current word, the smaller weight in the prediction. The vector representation of the word is obtained by optimizing the objective function:

$$\max_{w \in C} \sum_{w_1 \in V_w} \sum_{e_j \in Y} \#(w, c)[\log \sigma(W^T C)$$

$$+ k \sum_{c \in C} \rho(c)\log \sigma(-W^T C)]$$

The semantic similarity between words is measured using cosine similarity. In the experiment of this paper,
we take Negative Sampling to optimize the objective function and using the text in Wikipedia as the corpus.

4.2 Data Set

In order to compare the performance of each model, we compare correlations of model output and manually annotated data. The model with the higher correlation with the human annotated data set the better. In the study of English word semantic similarity, widely used data sets are WordSimilarity-353, SimLex-999, MEN and Rare-Word. However, due to the difference between English and Chinese, we cannot use them directly when we judge the semantic similarity of Chinese words [17]. Here we use the data set Words-240 which is similar with WordSimilarity-353. However it is selected and annotated for the Chinese.

The similarity obtained by each model cannot be directly compared because they are based on different standards. When evaluating the model, we judge the semantic relevance and sequence relevance of specific word pair sequence. When the manually annotated data is obtained, the semantic similarity degree of the manually annotated word pair are predicted by the model in turn, and the obtained similarity sequence is calculated with the manually annotated sequence by Spearman’s correlation coefficient. Then the relevance between the model and the manually annotated is obtained as the model score.

Although the method proposed in this paper is suitable for a variety of textual relation networks, some of baseline methods have more stringent requirements for corpus. For example, Information Content need to have category relationship between the words or similar link relations, and Skip-gram Model requires words relations within the context, or other similar link relations. Therefore, we selected the Wikipedia-ZN corpus which have a wide range of various relationships and is easy to access.

4.3 Experiment Procedure

We download the Wikipedia-EN-2016 corpus through the website WikiDB. It contains 2.7M wiki encyclopedia information, 4.9M inter-page link information, 1.1M classification information and 5.4GB text information. Among them, the category information contains Entity-Category information, inter-page links can be regarded as Entity-Entity information or Word-Entity information, which can also be regarded as Entity-Category information or Word-Category information if combined with the page MetaInfo, and the page body part can be treated as Word-Word information when based on n-gram mode.

The page entry title is considered as Entity that can get about 2.7M Entity information. We delete contents which are too short, containing isolated link, and get about 2.1M Entity information ultimately. Synonym information can be obtained from MetaInfo of the page, and multiple synonymous Entity are treated as the same Entity. We extract the links to other pages in one wiki page to get the Entity-Entity relation matrix. If there are n links to $E_i$ page at $E_0$ page, we set the Entity-Entity matrix element $E_{i0}$ as n. We extract the category information from MetaInfo of each page, combining with title of the page, then the Entity-Category matrix can be obtained. When $E_0$ belongs to category $C_0$, we set the Entity-Category matrix element $E_{01}$ as n. The corpus of the body part can form a word sequence through the simplified conversion, word segmentation, remove the Stop Word and word embedding [18], [19]. N consecutive words are considered related words. Each pair of related words $W_0 – W_1$ will add the elements $W – W_{01}$ in the Word-Word matrix plus 1.

We use each relation matrix derived from the corpus in basic methods to obtain the basic relevancy between words. Since some of the words that appear in the manually labeled evaluation set do not exist in some relational matrices, the semantic relevancy of some words cannot be obtained directly through the basic model or the relation matrix. This model can indirectly obtain the semantic similarity between words through the relationship between the relation matrices.

We divide experiments into two groups: one use completely manually annotated evaluation set, and the other use manually annotated evaluation set which delete the words do not exist in the relation matrix. Since some of the words that appear in the manually labeled evaluation set do not exist in some relational matrices, the semantic relevancy of some words cannot be obtained directly through the basic model or the relation matrix. This model can indirectly obtain the semantic similarity between words through the relationship between the relation matrices.

Although increasing the number of relation matrices associated with the model will improve the accuracy of the model, the cost is to increase the complexity and learning time of the model. When only using one relation matrix, it can converge to a stable result quickly. However, due to the randomness and regularization mechanism introduced by the model, the correlation between the results obtained by the model and the evaluation gather of manually annotation is quite different. The results obtained from the experiment show that a eclectic experimental result can be obtained when the model combines two or three relation matrices.

The number of relation matrices combined with the model will affect the experimental results. The types of the associated relational matrices and the basic model used by the relation matrices will also affect it. The experimental results of the model will decrease obviously when the type of relationship involved in the relation matrices associated with the model tends to be consistent. However, even with fewer models, involving more types of relationships, the accuracy of the model will be improved to a certain extent.
4.4 Experimental Results

We use the correlation degree $P$ between the manually annotation data and the similarities that the model output calculation results to estimate the effect of the model. The higher $P$ correlation degree, the more consistent with the results of manual annotations and the better effect of the model.

Because the difference of the corpus used by different control methods will affect the experimental results to a great extent, we use the set of relations used in several methods in experiments of our model. On the other hand, the words of Word-240 manually annotation data do not exist in some relations of the corpus chosen in this experiment, which is the typical case of the sparse feature of the single corpus. We designs two sets of control experiments to verify the remission effect of the model to the sparse feature. One group calculates the complete vocabulary in Word-240 (complete), and the other group calculates existing words in the corpus (deletion). The comparison of experimental results are given in Table 1.

It can be seen from the table that the experimental results of this experiment depend on the types of relationship and the pre-model. When using a single model, the performance of the pre-model can be improved with a slight computational cost. With the increase of the associated relations, the performance of this model gradually increases. In particular, when combining the two relatively simple models ID + SP, we can get results similar with Skip-gram model, and better results can be obtained when combined ID + SGM + SP. To deal with sparse data items, we can combine multiple relationships to reduce the impact of sparse data to a certain extent. However, in the case of low data sparseness and less use of relationship, this method is not as good as a single basic method. In general, this model can improve the effect of semantic correlation model of words by combining various models.

4.5 Computational Complexity

The computational complexity of the model proposed in this paper has a great relation with the number of associated relational types. On the one hand, the result can be obtained in a relatively short period of time when using a single relationship to representation learning. When the number of associated relations increases, and the learning efficiency rises, the learning complexity also increases. Due to some of the relationships depend on the pre-model, it is necessary to calculate the pre-model when associated with some relationships, which increases the computational complexity of the model. On the other hand, the convergence time of learning can be increased through increasing the dimension of the model when more relations are combined.

The space complexity of model learning can be expressed as the number of entity vector representation matrix. And it is slightly difficult to estimate the computational complexity of model learning. If the learning process uses fixed learning rate and fixed learning rotation, the complexity of the model learning process is related to many factors, such as the number of entities associated with the model (de-
noted as $C_E$), the number of relationships between entities (denoted as $C_X$), the semantic vector dimension defined by the model (denoted as $C_{dim}$) and the number of iterations (denoted as $T$).

$$O(C_E, C_X, C_{dim}, T) = \prod C_E \cdot \prod C_X \cdot C_{dim} \cdot T \quad (15)$$

In the actual model learning process, we often use the size of the error rather than the number of iterations as the end condition of the learning. The model uses different iterations for different initial values, different training data, and different learning rates. In this case, we tend to use the number of iterations of learning to evaluate the computational complexity of the model. Here we use the above learning strategies and parameters to perform ten times learning experiments with different training data at different random initial values, and calculate the average of the final iteration times to get the Table 2.

When there are more than four types of combinations, the computational complexity is very high and the time cost is unacceptable. According to the experimental experience, the combination of two or three relationships can be completed in a relatively acceptable time cost, and get better results.

### 4.6 Results Description

Experiments show that the improved method proposed in this paper has obvious effect, but the results obtained from the Chinese corpus are generally lower than similar experiments in English corpus. We believe that the manually annotated data used in this paper has the problem of inconsistent criteria. Some cases for which experiment results is greatly different from Words-240 are given in Table 3.

There are two approaches to judge the semantic similarity of words:

1. Assume $c_1, c_2$ are the information that completely describes the words $w_1, w_2$, so $\text{sim}(w_1, w_2) = (c_1 \cap c_2)/(c_1 \cup c_2)$.
2. The words $w_1, w_2$ describe the degree of similarity in the category.

For example, “clothes” and “wardrobe”, according to approach1 can get a higher degree of similarity because the two words have “clothing” concept; but through approach2 will get almost irrelevant results, because “clothes” and “wardrobe” are quite different categories.

Words-240 artificial annotation results belong to the mix of two kinds of approach. This type of mixing may cause the low effectiveness of the experimental results to a certain extent. For artificial annotation data set of English corpora, there are much better choices such as WordSimilarity-353, SimLex-999, MEN, Rare-Word and so on. However, there are few authoritative artificial annotation data set of Chinese corpora. As the translation of words will greatly affect the effect of artificial annotating data set, it is impossible to use English data sets directly through translation. As a result of the above reasons, this experiment finally chooses the relative authoritative Chinese artificial annotation data Words-240.

### 5. Conclusion

In this paper, we propose a kind of word relevance models which can fuse multiple relationships. By linking the relations of each corpus to the vector space of the same semantics, we can perform the representation learning of the word vector, use a variety of expected relationships. Experiments show that word correlation model which fuses multiple relationships gets higher accuracy than the model with single relation, especially in the case of data sparseness. Increasing the number of relational linked by the model can effectively improve the accuracy of the model in the case of data sparseness. However, as the number of the association increases, the computational complexity of the model increases obviously. In future work, we plan to focus on how to simplify the model and reduce the computational complexity of the model combined with multiple relationships.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China (61672040, 61972003). We would also like to thank the anonymous reviewers for their helpful comments.

### References

Jianyong Duan is a Professor, born in 1978.10. He graduated from Department of computer science, Shanghai Jiao Tong University by 2007.12. His major research field including natural language processing and information retrieval.

Yuwei Wu is a master in College of Computer Science and Technology, North China University of Technology. Her major research field is Semantic similarity calculation.

Mingli Wu received the B.S. degree in computer science and technology from Beijing Information and Technology Institute, Beijing, China in 1999 and M.S. degree in computer science and technology from North China Institute of Computing Technology, Beijing, China in 2002. He received the Ph.D. degree in computer science and technology from Hong Kong Polytechnic University, Hong Kong in 2008. Since 2009, he has been a lecture with Department of Computer Science and Technology, North China University of Technology. He is the author of more than 20 articles. His research interests include natural language processing, information retrieval and data mining.

Hao Wang received the Ph.D. degree in Computer Application Technology from Tsinghua University in 2013. He is now an assistant professor in College of Computer Science and Technology, North China University of Technology. He has published more than 20 papers in international conferences and journals. His research interests include machine learning and data analysis.