Short Text Feature Extraction via Node Semantic Coupling and Graph Structures

Huifang Ma, Xiaoqian Liu, Lan Ma, Yulin Hu

Short Text Feature Extraction via Node Semantic Coupling and Graph Structures

Huifang Ma[1,2]*, Xiaoqian Liu[1], Lan Ma[1], Yulin Hu[3]

[1] College of Computer Science, Northwest Normal University, Lanzhou Gansu, China
[2] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
[3] Tianjin No.1 High School, Tianjin, China
mahuifang@yeah.net

Abstract. In this paper, we propose a short text keyword extraction method via node semantic coupling and graph structures. A term graph based on the co-occurrence relationship among terms is constructed, where the set of vertices corresponds to the entire collection of terms, and the set of edges provides the relationship among terms. The setting of edge weights is carried out from the following two aspects: the explicit and implicit relation between terms are investigated; besides, the structural features of the text graph are also defined. And then, a new random walk method is established to effectively integrate the above two kinds of edge weighting schemes and iteratively calculate the importance of terms. Finally, the terms are sorted in descending order and the top K terms are extracted to get the final keyword ranking results. The experiment indicates that our method is feasible and effective.

Keywords: Semantic coupling, Graph structure, Short text, Random walk

1 Introduction

With the rapid growth of the information age, the rising of a great deal of Internet platforms such as Weibo, WeChat, talk, news, group purchase, mail, and mobile messaging provide a convenient communication environment for people. Mean-while, many forms of short text data have also been introduced into people's daily lives. Different from the traditional texts, these short texts are mainly a brief description, comments and views, simple answer or emotional expression, the length of the text is generally no more than 140 characters. At present, there are many short texts, which contain large amounts of information and include people's reflection and evaluation of all kinds of social phenomena or commodities. Therefore, to quickly and accurately obtain useful information from a large amount of short text data, keywords extraction technology plays a very crucial role.

The traditional keyword extraction algorithms are always suitable for long texts. Compare to long texts, short texts have more distinguishing features, such as decentralized information, more casual language expressions, less grammatical specifications, and sparse features. Therefore, it is very important to propose an effective keyword

adfa, p. 1, 2011.

extraction algorithm for short texts. In general, existing short text keyword extraction algorithms can be roughly classified into three main cat-egories:1) Statistics-based algorithm: The significance of a term is mainly considered with regard to its frequency, position, etc. such as TFIDF algorithm [1,2], N-Gram algorithm [3], but its deficiency lies in not taking into account the implicit semantics between terms. 2） Graph-Based keyword extraction algorithms: This kind of algorithm relies on word frequency statistics. By mapping terms and their semantic relations to text structure diagrams and then extracting some important vertices as keywords. The disadvantage of this approach is that it only considers the structure of the graph, ignoring external information like node properties. 3) Semantic-based algorithm: Using semantic dictionaries or lexical chain methods to acquire semantic knowledge between terms to extract text keywords. The algorithm improves the accuracy of the extraction, but it relies on the text understanding scenarios. It is impossible to extract words or phrases that are not contained in the knowledge base, and strict in the text format.

In this paper, a short text keyword extraction method is proposed, which is named as Short Text Keyword Extraction via Node Semantic Coupling and Graph Structures, SKESCGS, for short. A term graph based on the co-occurrence relationship among terms is established, where the set of vertices corresponds to the entire collection of terms, and the set of edges provides the relationship among terms. And then the setting of edge weights is carried out from the following two aspects: On one hand, the explicit and implicit relation between terms are investigated, On the other hand, the structural features of the text graph are also defined. Then, a new random walk method is established to effectively integrate the above two kinds of edge weighting schemes and iteratively calculate the importance of terms. Experimental results indicate that our method is feasible and effective for short text feature extraction.

The remainder of this paper is organized as follows. In Section 2 we describe the relevant theoretical knowledge. The proposed short text keyword extraction algorithm is detailed in Section 3. In Section 4, we report experimental results of the proposed algorithm. Finally, conclusion and future work are described in Section 5.

2 Problem Preliminaries

2.1 Semantic Intra-couplings within Term Pairs

The semantic intra-couplings within term pairs [8] is to explore the explicit se-mantic relations between terms. It is assumed that terms appearing in the same text have a co-occurrence relationship. The higher the co-occurrence frequency of term pairs, the stronger their relevance is.

Definition 1(*TPF-IDF*): *TPF* is the number of times a pair of terms appear in the same text, *IDF* is the number of texts a pair of terms that appear together, *TPF-IDF* reflects the importance of paired terms in a corpus for a text, which is defined as:

$$TPFIDF((t_i,t_j),d,D) = TPF((t_i,t_j),d) \times IDF((t_i,t_j),D)$$

$$IDF((t_i,t_j),D) = \log_2(\frac{|D|}{DF(t_i,t_j)}) \tag{1}$$

where (t_i,t_j) represents a pair of terms, $|D|$ is the total number of texts, and d is a single text in a text set D.

For $\forall(t_k,t_i) \in D(k,i \in [1,N], k \neq i)$,we define:

$$P^{Ia}(t_k \mid t_i) = \frac{TPFIDF(t_k,t_i)}{\sum_{k=1}^{N} TPFIDF(t_k,t_i)} \tag{2}$$

as the probability of the term pair (t_k,t_i) in document set D, and $TPFIDF(t_k,t_i)$ represents the *TPF-IDF* of term pair (t_k,t_i).

The probability of a given term t_i in all term pairs is defined as follows:

$$P^{Ia}(t_i) = (P^{Ia}(t_1 \mid t_i), P^{Ia}(t_2 \mid t_i), \ldots, P^{Ia}(t_k \mid t_i), \ldots, P^{Ia}(t_N \mid t_i)) \tag{3}$$

2.2 Semantic Inter-couplings between Term Pairs

The internal coupling of term pairs introduced in the previous section captures only the explicit relationship between two adjacent vertices in the graph and does not take the interactions between the other words in the graph into consideration. Therefore, a method of capturing implicit relations between terms based on graph is proposed.

For any $(t_k,t_i) \in D(k,i \in [1,N], k \neq i)$,we have:

$$P^{Ie}(t_k \mid t_i) = \frac{SP(t_i,t_k)}{\sum_{k=1}^{N} SP(t_i,t_k)}$$

$$SP(t_i,t_k) = \frac{\sum_{k=1}^{N} PL(t_i,t_k)}{PL(t_i,t_k)} \tag{4}$$

Equation (4) reflects the similarity degree between term t_i and term t_k. The closer the distance, the more similar t_i and t_k will be. Besides, $PL(t_i, t_k)$ represents the shortest path between term t_i and term t_k. For a given t_i, its probability distribution is as follows:

$$P^{Ie}(t_i) = (P^{Ie}(t_1 \mid t_i), P^{Ie}(t_2 \mid t_i), \ldots, P^{Ie}(t_k \mid t_i)) \tag{5}$$

Definition 2(IaR and IeR): Given a text set D, the intra-term pair couplings relation(IaR) and the inter-term couplings (IeR) between term (t_i,t_j) in the text set D is defined as follows:

$$IaR(t_i,t_j) = RS(P^{Ia}(t_i), P^{Ia}(t_j)) \tag{6}$$

$$IeR(t_i,t_j) = RS(P^{Ie}(t_i), P^{Ie}(t_j)) \tag{7}$$

IaR(t_i,t_j) and IeR(t_i,t_j) represent the internal coupling relationship and external coupling relationship of the term pair(t_i,t_j) respectively. RS is the relational strength function. We adopt cosine similarity as the relational strength function to evaluate the coupling relationship between term pairs.

2.3 The structural features of the graph

Inspired by [9], vertex attributes can be obtained not only from external data but also from the internal structure information of the graph. For each vertex, we selected four internal attributes to calculate the similarity between terms and they are assortativity, degree of a vertex, the number of neighbours' vertices at 2, the number of neighbours' vertices at 3, respectively. In addition, to avoid large values, this paper takes the logarithm of all internal properties.

3 The Proposed Approach

The proposed SKESCGS mainly contains the following steps:

- Pre-processing of the text, including word segmentation, stop word removal, part of speech tagging, etc;
- Constructing a term graph and initializing vertex weights for term graphs;
- Calculating similarity via node semantic coupling and graph structure features;
- Integrating (2) and (3) to set the weights on the edges, iterative calculations are performed to obtain the final ranking results of keywords.

3.1 Term Graph Construction

Given a text set $D = \{d_1, d_2, ..., d_M\}$, after pre-processing, each text d_i is represented by its attribute vector $d_i = t_j^i = (t_1^i, t_2^i, ..., t_N^i)$, where N is the number of different terms extracted from the entire text set.

In the term graph construction process, a term corresponds to a vertex, and the edges define the co-occurrence relationship between these terms. The graph $G = (V, E)$ is constructed based on the co-occurrence relationship among terms, where the set of vertices $V = \{t_1, t_2, ..., t_n\}$ corresponds to the entire collection of terms, and the set of edges $E = \{e_{1,1}, e_{2,2}, ..., e_{i,j}\}$ provides the co-occurrence relationship among terms.

3.2 Vertex Weight Initialization

After constructing the term graph, we need to define the weight of the vertex to indicate its importance. In our algorithm, the initial weights are set to terms based on their part of speech, which is defined as:

$$\partial = \begin{cases} 0.8\,(nouns\ and\ verbs) \\ 0.6\,(adjectives\ and\ adverbs) \\ 0\,(other\ parts\ of\ speech) \end{cases} \tag{8}$$

To be more specific, if the term is a noun or a verb, the initial weight of the term is 0.8; if the term is an adjective or an adverb, the initial weight of the term is 0.6; the initial weight of the term is 0 otherwise.

3.3 Calculation of Similarity Based on Semantic Coupling

By synthesizing the internal coupling and external coupling between pairs of terms, the comprehensive semantic relationship between terms can be fully investigated. The semantic coupling similarity of term pair (t_i,t_j) in text set D can be calculated as:

$$SCS(t_i, t_j) = (1-\alpha) \times IaR(t_i, t_j) + \alpha \times IeR(t_i, t_j) \tag{9}$$

Where $\alpha \in [0,1]$ is the parameter to determine the relative importance of the internal coupling relationship and the external coupling relationship. The value of $SCS(t_i,t_j)$ falls into [0,1], 0 indicates that there is no relationship between the two terms, 1 means two words are exactly the same. That is, the higher the value of $SCS(t_i,t_j)$, the higher the similarity between two terms.

In term of the similarity calculation process based on structural features, the weights of the edges between (t_i,t_j) are represented by the similarity s_{ij} between the corresponding vertex attributes (xi,xj), s_{ij} >0.

In this paper, radial basis function (RBF) is adopted as the similarity definition between vertex attributes as:

$$s_{ij} = e^{-\gamma \|x_i - x_j\|_2^2} \tag{10}$$

where the positive parameter γ controls the influence of the attribute distance, the RBF kernel function is equivalent to the inner product $\phi(x_i)^T \phi(x_j)$ of the two infinite-dimensional vectors projected first from x_i and x_j. So s_{ij} can capture the nonlinear similarity between x_i and x_j.

3.4 Edge weight calculations in text graphs

For the constructed text graph G, the similarity between vertices is regarded as the weight of the vertices, and the semantic similarity is used to calculate the similarity between the vertices to obtain the graph G_1, and the similarity between the vertices is calculated using the method of structural features to obtain the graph G_2. The transfer matrices **P** and **Q** of graph G_1 and G_2 are calculated respectively. Since both graphs G_1 and G_2 are undirected graphs, the edges (t_i,t_j) in graphs G_1 and G_2 can be considered as two directed edge (t_i,t_j) and (t_j,t_i), **P** and **Q** are L*L-dimensional matrix, where each entry **P**(i,j) is the similarity calculated by the semantic coupling, and each entry **Q**(i,j) is the similarity calculated by the structural features. Then, randomly walk is performed iteratively calculate the weight of each vertex. The weights are sorted in an ascending order. The top 10 term are chosen as the keyword of the text set. The calculation for vertex weight formula is defined as:

$$\pi^{(t+1)} = (1-d)Q\pi^{(t)} + dP\pi^{(t)} \tag{11}$$

Where

$$Q_{ij} \equiv \frac{s_{ij}}{\sum_{k \in V} s_{kj}}$$

The vertices in this paper are given initial weights, thus $\pi^0 = \partial'$,It is worth noting that ∂' is normalized, and π^t is the keyword weight vector after iterating *t* times.

4 Experiments and Results Analysis

In this section, we conduct a series of experiments to prove the effectiveness of SKESCGS in short text scenario. All the algorithms are implemented in Java and are tested on Intel Core i5-4200U with 2.30GHz processor and 8GB main memory, having 64-bit Windows 10.

4.1 Data Sets and Evaluation Metrics

In order to verify the effectiveness of our approach, we conducted several experiments on both Chinese data sets and English data sets[4], respectively. We adopted 15 classes with 1500 paper titles obtained from CCF recommended list in Rank A and B as English data sets, and collected 6 classes with 2000 paper titles in each category from CSCD as Chinese data sets. 10-fold cross validation is adopted to get the classification accuracy of short text for this method. Repeating the experiments 10 times and calculating the average of the classification accuracy obtained 10 times as the final classification result.

Pre-processing the data set includes data denoising, text segmenting, stop words filtering. Among them, Chinese text segmentation and part-of-speech tagging are implemented through a Java call to the Chinese Academy of Sciences Segmentation System (NLPIR) function. Stem Segmentation is achieved by the classical porter algorithm. The results obtained by the method in this paper are converted in the form of keyword vectors and *k*-NN and SVM classifiers are used for classification. Besides, we adopt Accuracy and F-measure as the evaluation of metrics [10].

4.2 Experimental Results and Analysis

In this section, we aim to observe the efficiency of our methods from two aspects: First, we visualize the selection results and evaluate our schemes for short text feature selection and compare the performances with other selection methods. Then, the keyword sets extracted by different methods are applied to the SVM and the *k*-NN classifier to test the effect of different algorithms on the classification of short texts.

We chooses keyword extraction method which considers the semantic coupling without considering the structural features of graph, KES, for short; keyword extraction method that considers only the structural features of the graph but does not consider the semantic coupling, KEGS, for short; and a graph based keyword extraction method TKG2|W1|Cc [8] .

The reason that we select the above three methods as the comparison method of our method is based on the following considerations: 1) our method is based on the improvement of the semantic coupling and the features of the text graph structure, the keyword extraction method that just considers the semantic coupling without considering the structural features of graph, and the keyword extraction method that only considers the structural features of graph without considering the semantic coupling are the most similar to the method of this paper. 2) TKG2|W1|Cc method is also a graph-based keyword extraction algorithm, and the rules for constructing the text graph in this method are the same as those in the method in TKG2 [8].

Influence of Keyword Set Size on Short Text Classification: Because the limitation of this paper, we only show the experimental results on Chinese dataset. We take the first 30, 60, 100, 110, 130, 160, 180, 200, 230, 250, 280 and 300 terms of the keyword set as the feature dictionary and utilize the SVM and the *k*-NN classifier respectively for testing.

As is shown in Figure 1 and Figure 2, the keyword set obtained by this method can effectively classify short texts on both SVM and k-NN classifiers, and the classification effect of SVM classifiers is better， and more consistent with the method of this paper. As the length of feature lexicon gradually increases, the model trained by SVM is superior for classification. Both Accuracy and F-measure value first show an increasing trend, and after the number of feature vocabulary reaches 200, it reaches a peak and is prone to be stable. Using the k-NN classifier trained model classification, the accuracy and F-measure value showed a similar trend of increasing first and then decreasing and reached the peak, when the number of feature dictionary was 110, and the classification effect was the best.

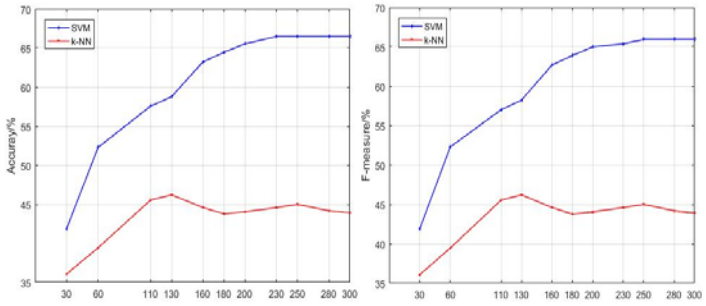

Figure 1. *Accuracy and F-measure*

Comparison of Keyword Sets: We compare feature dictionaries obtained from the above 4 kinds of strategies to verify that our method can get a high accuracy for short text feature selection. Table 1 and Table 2 are the comparison results of different feature selection methods. It is obvious that the keyword sets obtained by the KEGS method are relatively poor in that it does not consider the semantic information and does not represent text category features. The KES method and the TKG2|W1|Cc algorithm consider the terms as textual forms but ignore the social attribute factors carried in the document itself, and the results obtained need to be improved. It can be proved that the semantic information between terms and the attribute characteristics of terms cannot be ignored. Obviously, our algorithm fully considers the implicit semantics between terms

and comprehensively considers the structural features of the text graph itself, and the obtained results are more reasonable.

Table 1. *Keywords Extracted from Chinese Data Sets with Different Algorithms*

Method	KES	KEGS	$TKG_2\|W^1\|C^c$	**SKESCGS**
1	algorithm	classical	encrypt	**safety**
2	model	artificial	algorithm	**software**
3	optimization	grade	public key	**social**
4	network	request	software	**verification**
5	detection	difference	sexy	**difference**
6	application	delimitation	safety	**entity**
7	improvement	guide	ontology	**recognition method**
8	oriented	taxi	attack	**internet**
9	data	writing	user	**position**
10	analysis	performance index	texture	**sensor**

Table 2. *Keywords Extracted from English Data Sets with Different Algorithms*

Method	KES	KEGS	$TKG_2\|W^1\|C^c$	**SKESCGS**
1	design	two-joint	activity	**attack**
2	network	impact	topic	**probabilistic**
3	base	seminally	elastic	**cluster**
4	optimal	subpixel	optical	**live**
5	interact	visual interact	femtocell	**sensor**
6	learn	mechanic	radio	**encrypt**
7	data	repetition	cooperative	**mechanic**
8	analysis	transmittal	efficiency	**device**
9	model	note	computer	**layer**
10	efficiency	spherical	firewall	**impact**

Table 3. Classification performance of different feature selection methods

Method	***F1-Measure***
KES	**0.6283**
KEGS	**0.5918**
$TKG2|W^1|C^c$	**0.6297**
SKESCGS	**0.6594**

(a) Chinese data sets

Method	***F1-Measure***
KES	**0.4659**
KEGS	**0.3973**
$TKG2|W^1|C^c$	**0.4725**
SKESCGS	**0.6263**

(b)English data sets

Effects of Different Extraction Methods on Short Text Classification: In the previous experiment, we confirmed that the classification effect of this method is superior to the k-NN classifier on the SVM classifier, and the classification accuracy is highest when the length of feature dictionary is 200. Thus, we select SVM classifiers to perform experiments on Chinese and English data sets respectively to verify the effect of different methods on short textual classification. The F-measure values are summarized in Table 3.

It is clear that our method outperforms the other three methods, which suffices to show that our method is more effective for short textual classification and is applicable to different languages. Thus, the implicit semantics between words and structural features in the text graphs has a greater impact on the classification of short texts, and the result indicates that our method is more accuracy.

5 Conclusion

The aim of this paper is to introduce a new method to extract keywords from short text. Both the explicit and implicit relation between terms are investigated together with the structural features of text graph are considered to set the edge weights. And then a random walk method is established to effectively integrate the above two kinds of edge weighting schemes and iteratively calculate the importance of terms. Finally, the top K terms are sorted in descending order to extract to get the final keyword ranking results. Experiments on both Chinese and English datasets proves the effectiveness of our approach.

Acknowledgement. The work is supported by the National Natural Science Foundation of China (No.61762078, 61363058, 61762079) and Guangxi Key Laboratory of Trusted Software (No. kx201705).

6. References

[1] Guo A, Yang T. Research and improvement of feature words weight based on TFIDF algorithm[C]// Information Technology, Networking, Electronic and Automation Control Conference, IEEE. IEEE, 2016:415-419. *(references)*

[2] Doen Y, Murata M. Construction of concept network from large numbers of texts for information examination using TF-IDF and deletion of unrelated Words[C]// International Conference Joint on Soft Computing and Intelligent Systems. 2014:1108-1113.

[3] Sidorov G, Velasquez F, et al. Syntactic N-grams as machine learning features for natural language processing[J]. Expert Systems with Applications, 2014, 41(3):853-860.

[4] Ma H F, Xing Y Y, et al. Leveraging term co-occurrence distance and strong classification features for short text feature selection[C]// International Conference on Knowledge Science, Engineering and Management. Springer, Cham, 2017:67-75.

[5] Hua W, Wang Z, et al. Short text understanding through lexical-semantic analysis[C]// IEEE, International Conference on Data Engineering. IEEE,2015:495-506.

[6] Tang J, Wang X, et al. Enriching short text representation in microblog for clustering[J]. Frontiers of Computer Science in China, 2012, 6(1):88-101.

[7] Abilhoa W D, Castro L N D. A keyword extraction method from twitter messages represented as graphs[J]. Applied Mathematics & Computation, 2014, 240(4):308-325.

[8] Chen Q, Hu L, et al. Document similarity analysis via involving both explicit and implicit semantic couplings[C]// IEEE International Conference on Data Science and Advanced Analytics. IEEE, 2016:1-10.

[9] Hsu C C, Lai Y A, et al. Unsupervised ranking using graph structures and node attributes[C]// Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017:771-779.

[10] Gao L., Zhou S, et al. Effectively classifying short texts by structured sparse representation with dictionary filtering[J]. Information Sciences,2015, 323:130-142.

[11] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine[J]. Computer Networks & Isdn Systems, 1998, 56(18):3825-3833.