BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 02/06 pp98–109
Volume 5, Number 2, June 2022
DOI: 10.26599/BDMA.2021.9020023

News Topic Detection Based on Capsule Semantic Graph

Shuang Yang and Yan Tang*

Abstract: Most news topic detection methods use word-based methods, which easily ignore the relationship among words and have semantic sparsity, resulting in low topic detection accuracy. In addition, the current mainstream probability methods and graph analysis methods for topic detection have high time complexity. For these reasons, we present a news topic detection model on the basis of capsule semantic graph (CSG). The keywords that appear in each text at the same time are modeled as a keyword graph, which is divided into multiple subgraphs through community detection. Each subgraph contains a group of closely related keywords. The graph is used as the vertex of CSG. The semantic relationship among the vertices is obtained by calculating the similarity of the average word vector of each vertex. At the same time, the news text is clustered using the incremental clustering method, where each text uses CSG; that is, the similarity among texts is calculated by the graph kernel. The relationship between vertices and edges is also considered when calculating the similarity. Experimental results on three standard datasets show that CSG can obtain higher precision, recall, and F1 values than several latest methods. Experimental results on large-scale news datasets reveal that the time complexity of CSG is lower than that of probabilistic methods and other graph analysis methods.

Key words: news topic detection; capsule semantic graph; graph kernel

1 Introduction

With the rapid development of the mobile Internet, network culture as a brand-new communication medium, has become an important way for modern people to collect and obtain information. In the face of a large number of Internet and traditional news materials, how to quickly and accurately capture and obtain information that people are interested in and how to effectively organize and integrate these network news materials are the main concerns of current natural language processing research. News topic detection is a subtask of topic detection and tracking, which aims to detect topics from various text corpora. In general, a topic can be considered an abstract event, comprising some specific events with semantic relevance. Events are considered nontrivial events that occur at a specific time and a specific place[1]. As a basic problem of information retrieval, news topic detection can help decision makers detect meaningful topics effectively. Therefore, it has attracted widespread attention in public opinion monitoring, decision support, and emergency management[2–5].

Most current news topic detection models are implemented using word-based methods or topic model-based methods. Word-based methods use a vector space model (VSM) to represent news documents and extract keywords in a corpus through a certain method. Ignoring the relationship among words is easy, and a problem of semantic sparseness exists, resulting in low topic detection accuracy. Topic model-based methods use the polynomial distribution to construct topic model structures for mining the topics of document sets. The reasoning process in these models is too complicated, resulting in high time complexity.

• Shuang Yang and Yan Tang are with the College of Computer and Information Science, Southwest University, Chongqing 400000, China. E-mail: doublesunsw@163.com; goosetang910@163.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2021-11-05; accepted: 2021-11-18

In recent years, some researchers have proposed topic detection methods on the basis of graph analysis and have achieved great success. The topic detection method based on graph analysis introduces graph structure for text representation and detects topics by extracting text content feature items as vertices, including the correlation among feature items as edge structure graphs. The graph analysis method provides a new idea for topic detection and has a significant effect on news datasets. However, for a large number of news documents, the algorithm is time consuming when all the documents are added to the graph.

On the basis of the above analysis, a news topic detection model based on capsule semantic graph (CSG) is proposed. TextRank is used to extract text keywords, and the keywords that appear in each text at the same time are modeled as a keyword graph. Words are connected closely to make keywords in the keyword graph, which is divided into multiple subgraphs through community detection. The words in each subgraph are closely related. Each subgraph after community detection is regarded as a vertex, and the words in the vertex are used as features to compare the vertices. To build a capsule semantic map, the semantic information of each text is captured; the incremental clustering method is also used to cluster news texts where each text is represented by CSG, and the similarity among texts is calculated by the graph kernel.

2 Related Works

2.1 Topic detection

Most studies on news topic detection use topic detection methods based on feature extraction, probability models, and graph analysis.

Methods based on feature extraction usually use VSM to represent news documents. The keyword information extracted by a certain method is used as the document feature. Then, news documents describing the same topic are clustered together according to certain clustering criteria. The most commonly used method based on feature extraction is term frequency-inverse document frequency (TD-IDF)[6]. To effectively extract feature words in news texts, Bun and Ishizuka[7] proposed the TF-IDF method to calculate the average weight of word vectors and cluster topics with high weight values. Chen and Jin[8] improved the text feature extraction method and used the formula of the kinetic energy theorem to evaluate the suddenness of words by using the TF-IDF method. In addition, TextRank[9] and named entity recognition[10] are two other effective feature extraction methods. Pu et al.[11] proposed a method of extracting topic sentences on the basis of the weighted TextRank algorithm in a single document to obtain information about key news events. Qu et al.[12] proposed a news event detection method on the basis of key element recognition, which uses named entity recognition and part-of-speech tagging in feature word selection, combined with VSM and time attributes to obtain news core events. Topic detection methods based on feature extraction treat documents as collections of several vocabularies. The appearance of each word in a document is independent, and the relationship among words is ignored; the semantic information of the text cannot be expressed, resulting in low topic detection accuracy.

Topic detection methods based on probability models perform topic detection by processing the probability distribution of a topic and the word in the vector space. Chen and Liu[13] proposed a knowledge-based topic model, which excavates some reliable prior knowledge from past learning and modeling results; they also used prior knowledge to guide model reasoning to generate more coherent topics. Qiu et al.[14] proposed a popular topic detection method on the basis of the hybrid model of VSM and improved Latent Dirichlet Allocation (LDA); the LDA model combined with the PageRank algorithm was used to deeply mine the structures of social networks; the VSM model and the improved LDA model were used for hybrid modeling; the calculation of the similarity of microblogs was obtained through a single-channel clustering algorithm based on a hybrid model to obtain the clustering results of popular topics. Existing methods based on probability models have two problems. First, current topic modeling methods do not explicitly consider the problem of word co-occurrence, which refers to two words appearing in the same document at the same time. Second, due to the complicated reasoning process in the model, its time complexity is high.

Researchers have recently proposed some graph analysis methods for topic detection. Topic detection methods based on graph analysis use the content characteristics of texts and the correlation between each feature item to construct a text graph model for topic detection. Sayyadi and Raschid[15] constructed a graph of the co-occurrence relationship of words in the entire document corpus, conducted community detection to divide the graph into several parts, and performed topic

detection by assigning documents to each community. Zhang et al.[16] proposed a multidimensional topic extraction method on the basis of a hierarchical semantic graph model, conducted the hierarchical extraction of feature words to construct a semantic graph, and performed segmentation and structural analysis of the semantic graph to extract topics. Hamm et al.[17] used posIdfRank to sort words to extract keywords and constructed a keyword map to discover topics. Topic detection methods based on graph analysis can mine the correlation among feature words and can continuously expand and construct the corresponding topic relationship graph with the feature words as the center. Azadani et al.[18] constructed a concept-based model of the source document and mapped the document to the concept to obtain the association feature relationship; they then used the association feature relationship to construct a representation graph to obtain the feature words of the document set. Drury et al.[19] used Bayesian networks to construct causal relationships among news report topics, converted them into event graphs, and clustered the event graphs according to topic distribution.

2.2 Graph kernel

As a general data structure representation, graphs can describe complex structured data, such as proteins, genes, and social networks. Given the advantages of structured data in the graph structure itself, the data representation based on the graph structure is becoming increasingly common. How to process these graph data to obtain valuable information has become a concern in many fields.

With increasing attention to structured data, kernel methods based on structure objects and graph structures are gradually being defined. They are called graph kernels. The essence is to use nuclear machines to deal with inaccurate matching problems in graphs. So-called graph kernels map graphs to the feature vector space, so that the similarity between two graphs can be represented by their dot product in the feature vector space. The basic idea is to map the input graph structure data from the original space to the high-dimensional feature space by defining an appropriate graph kernel function, so that dot product operations can be performed in the mapped high-dimensional space. Simply put, the graph kernel function is a mapping from the graph data space to the high-dimensional feature space. Through the graph kernel function, the dot product of the data in the high-dimensional feature space can be directly obtained. Research on graph kernels has developed rapidly in recent years. It maps graphs to vector or dot product space and converts the operations required for pattern recognition from the original graph domain space to the mapped target space. Existing graph kernels include random walk kernels[20], subtree kernels[21], and shortest path kernels[22]. Random path kernels are based on the idea of counting the number of paths matched by two input graphs. The main idea of subtree kernels is to use the substructure or branch of the tree to represent the tree and then calculate the number of common substructures of the two trees. Shortest path kernels refer to the problem of transforming the original graph matching into the shortest path graph matching. This study considers the relationship between vertices and edges on the basis of the shortest path kernel, which improves the matching effect of the graph structure while ensuring the time complexity.

3 News Topic Detection Model Based on CSG

The news topic detection model based on CSG is shown in Fig. 1 and comprises three parts: Keyword graph construction, CSG construction, and topic generation.

(1) Keyword graph construction: The keywords of each text are extracted by TextRank, and the keyword graph is constructed with the keywords as vertices and the co-occurrence relationships of the keywords in sentences as edges.

(2) CSG construction: The keyword graph is divided into multiple subgraphs through community detection. Each subgraph is used as a vertex, and the semantic relationship between each subgraph is used as an edge to construct a CSG.

(3) Topic generation: Topics are generated using the incremental clustering algorithm. During the clustering process, graph kernels are used to measure the similarity among textual semantic graphs. Finally, topics are generated.

3.1 Keyword graph construction

The TextRank algorithm is used to extract keywords from the text to obtain representative words that can represent the semantics of the text. A keyword graph of the text is constructed by analyzing the co-occurrence relationships among keywords in the text.

3.1.1 Keyword extraction

For each document, counting all words as feature items

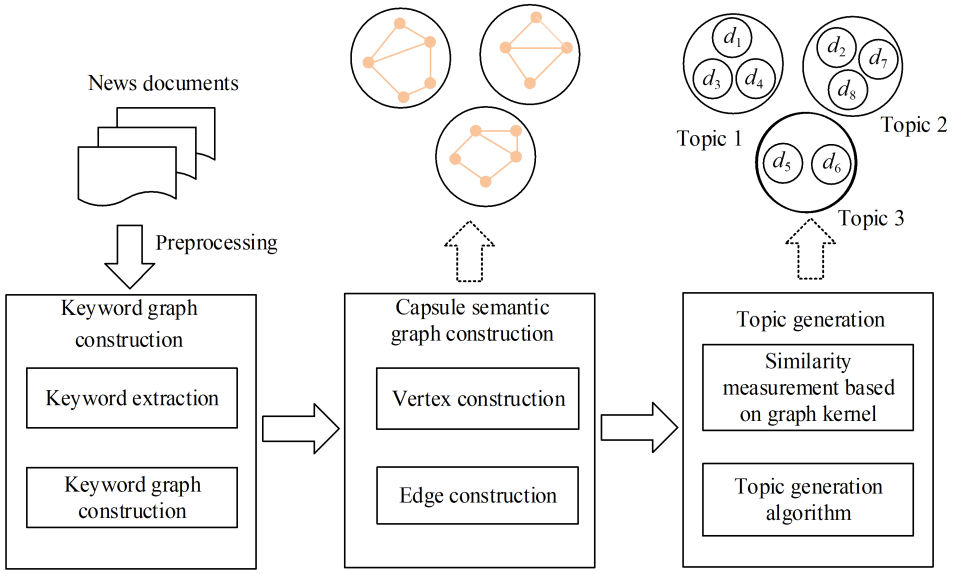


Fig. 1 Frame diagram of news topic detection model based on CSG. d_i represents the i-th news document.

is time consuming and unwise because, in a document, some words are unnecessary for the semantic expression of the document. That is, the contribution value of the word is low. The existence of these words inevitably reduces the efficiency and accuracy of topic detection. Therefore, extracting keywords from documents and using words with high contribution value for text representation are wise choices. Keyword extraction is the process of selecting a subset of representative terms from a preprocessed text term set. The selected term subset is called the feature set of the text. The keyword extraction algorithm is shown in Algorithm 1.

For the calculation of the contribution value in the algorithm, this article uses the TextRank algorithm. For the contribution degree $S(w_i)$ of each word w_i, the equation is as follows:

$$S(w_i) = \alpha \cdot \sum_{w_i \in \mathrm{In}(w_i)} \frac{W_{ij}}{\sum_{w_k \in \mathrm{Out}(w_j)} W_{jk}} \cdot S(w_j) + (1-\alpha) \tag{1}$$

Algorithm 1 Keyword extraction algorithm

Input: $W = \{w_1, w_2, \ldots, w_n\}$ //word set of text
Output: $K = \{k_1, k_2, \ldots, k_k\}$ //keyword set of text

1: Initialize set $K = \{\}$;
2: Construct a word graph, with words as nodes, and then use the co-occurrence relationship among words to construct an edge between two nodes;
3: Calculate the contribution value of each word in set W and sort the words according to the contribution value;
4: Select top k words and put them into the keyword set K to obtain the keyword set.

where α is the damping coefficient, $\mathrm{In}(w_i)$ represents the set of nodes that point to word node w_i, $\mathrm{Out}(w_j)$ represents the set of other nodes that the word node w_j points to, and W_{ij} represents the weight of the edge (w_i, w_j).

3.1.2 Keyword graph construction

After completing the keyword extraction, a keyword map can be built on the basis of the keywords in the document. The keyword graph, which the news document is transformed into, is represented as $G = (V, E)$ where the vertex set of G is represented by V, and the edge set is represented by E. A keyword graph must be constructed, with keywords as vertices and the co-occurrence relationships of keywords in sentences as edges. The construction process of the keyword graph includes three steps:

Step 1: Obtain the keyword set $K = \{k_1, k_2, \ldots, k_k\}$ in each document through the keyword extraction algorithm where k_i is the keyword of the sentence in the document.

Step 2: Take keywords as the vertices of the graph, and each keyword k_i corresponds to a vertex v_i of the graph. If the two keywords k_i and k_j appear at the same time in at least one sentence, then edge e_{ij} is added between the two keywords k_i and k_j, and the number of common occurrences is the weight of the edge.

Step 3: The number of sentences in the document that contain keywords k_i and k_j is denoted by $\mathrm{SF}_{e_{ij}}$, and $\mathrm{SF}_{e_{ij}}$ of each edge is calculated. If $\mathrm{SF}_{e_{ij}}$ does not exceed threshold edge_min, then delete this edge.

3.2 Construction of CSG

To ensure the semantic integrity of the graph structure representation and improve topic detection efficiency, the keyword graph is divided into multiple subgraphs as vertices through community detection, and the semantic relationships among subgraphs are used as edges to generate a CSG. CSG increases the semantic information of the text, so that the graph structure can represent the text information more completely. The construction process of CSG is demonstrated in Fig. 2. The process includes: (a) vertex construction; (b) edge construction.

3.2.1 Vertex construction

The document keyword graph shows the relationship among keywords, and the keyword graph of each document has a different density. To make the connection between keywords closer than usual, the keyword graph is divided into several subgraphs through community detection. The words in each subgraph are closely related, but the relationship among subgraphs is relatively weak. Each subgraph after community detection is regarded as a whole as the vertex of CSG. After community detection, graph G is expressed as $G = \{C_1, C_2, \ldots, C_n\}$ where n represents the number of subgraphs, and each subgraph C_i contains a set of closely related keywords. This study uses the community detection algorithm proposed by Newman[23] to generate the vertices of the graph. It is a well-known community detection method and has been proven to have good performance. This process can effectively speed up the calculation of subsequent components. The algorithm description is shown in Algorithm 2.

3.2.2 Edge construction

After the community detection, the multiple vertices generated in the previous section are independent of one another, and each vertex contains a different number of keywords. To construct edges between vertices, the semantic vector of each vertex is obtained by taking all the words in the vertex as features. According to the principle of the Word2Vec model, the word vector trained with the language model retains the semantic

Algorithm 2 Vertex construction algorithm

Input: DG $= (V, E)$ //document graph, where $V = \{v_1, v_2, \ldots, v_n\}$;
$E = \{(w_k, v_i, v_j) | k \in [1, m]; i, j \in [1, n]\}$, where m denotes the number of edges and n denotes the number of nodes;
θ //threshold of max_betweenness for splitting the keyword graph into several communities;
community_size_max //maximum community size, the number of keywords in a community should not be greater than this value;
community_size_min //minimum community size, the number of keywords in a community should not be less than this value

Output: $G = \{C_1, C_2, \ldots, C_n\}$

while edges of DG not equal to zero **do**
  find the max_betweenness according to the shortest paths calculated by breadth-first search (V);
  nodes_sum = length of V;
  if max_betweenness $\leqslant \theta$ **or** community_size_min $\leqslant$ nodes_sum $\leqslant$ community_size_max **then**
    return DG;
  end if
  if nodes_sum $<$ community_size_min **then**
    return DG
  end if
  while current_nodes $\leqslant$ original connected nodes **do**
  if edge = max_betweenness **then**
    remove the edge with the max _ betweenness value
  end if
  current_nodes =list of the connected nodes;
    if community_size_min $\leqslant$ current_nodes $\leqslant$ community_size_max **then**
    break;
  end if
  end while
  G could be obtained after the whole removing operations;
  return $G = \{C_1, C_2, \ldots, C_n\}$;
end while

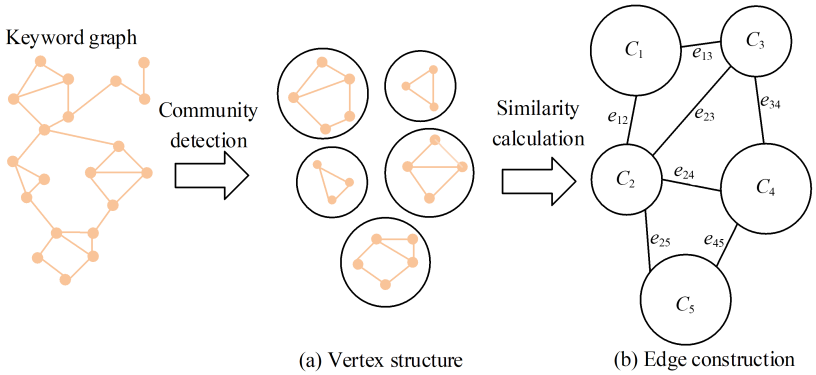


Fig. 2 Process of CSG construction. C_i represents the i-th subgraph and e_{ij} represents the weight of the edge between C_i and C_j.

information of the word. Therefore, this study uses the language model Word2Vec[24] for the training to obtain the vector of each word v_i in vertex C_i. Assuming that C_i contains m words, the semantic vector V_{C_i} of vertex C_i has the following equation:

$$V_{C_i} = \frac{1}{m}\sum_{i=1}^{m} v_i \tag{2}$$

The distance of the word vector on the space model represents the semantic similarity among words, and more similar words have similar positions on the vector space. Therefore, the edges are constructed by calculating the similarity among vertices, and the similarity value is regarded as the weight of edges between vertices. The similarity is calculated on the basis of the semantic vector of the vertices. The similarity equation between vertices C_i and C_j is as follows:

$$\mathrm{sim}(C_i, C_j) = \frac{V_{C_i} \cdot V_{C_j}}{\left\|V_{C_i}\right\| \cdot \left\|V_{C_j}\right\|} \tag{3}$$

where V_{C_i} and V_{C_j} represent the semantic vector of C_i and C_j, respectively. According to Eq. (3), weighted edges can be constructed between vertices. The vertex of CSG is a set of keywords, which can represent important information of a document. The edges of CSG are the relationships among key information, and these edges can represent the dependency relationships among keyword sets. Therefore, these edges can represent latent semantic relationships among vertices. The text semantic graph can capture the dependency and semantic relationships of various key information and has a strong document representation ability.

3.3 Topic generation

The last step of topic detection is usually to use a clustering algorithm to generate topics. Considering that this article transforms a text into a graph structure for representation, the similarity calculation method of the graph core is used during the clustering process to perform topic clustering. Given that CSG introduces semantic relationships among keyword sets and represents texts as complex graph structures, the traditional vector-based similarity calculation method is unsuitable for the graph structure model proposed in this study. Aiming at the characteristics of graph structure, a text similarity calculation method based on graph core is adopted.

3.3.1 Similarity measurement based on graph kernel

Graph kernel is a kernel method based on structural objects and graph structures, which can deal with the matching problem of graphs in a machine-based way. Graph kernel is used to map a graph into a feature vector space, and similar graphs can be represented by the dot product in the feature vector space.

According to the above analysis, the news text data have been transformed into a CSG. The similarity of the two texts is mainly judged on the basis of the degree of agreement between the detected features, and the graph kernel compares the degree of similarity of the substructures between the two graphs. Thus, this research uses graph kernel to measure the similarity of CSG.

Let G_1, G_2 denote the CSGs of two news documents d_1 and d_2, and the definition of the graph core on $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is shown in Eq. (4):

$$k(d_1, d_2) = \frac{\sum\limits_{v_1 \in V_1, v_2 \in V_2} k_{\mathrm{node}}(v_1, v_2) + \sum\limits_{e_1 \in E_1, e_2 \in E_2} k_{\mathrm{walk}}(e_1, e_2)}{\mathrm{norm}} \tag{4}$$

where k_{node} represents the kernel function of two vertices, k_{walk} is the kernel function of random walk on two edges, and norm is the normalization factor.

For the kernel function k_{node} used to compare two vertices, $k_{\mathrm{node}}(v_1,\ v_2)$ is defined in Eq. (5):

$$k_{\mathrm{node}}(v_1, v_2) = \max\left(\left(V_{v_1}^{\mathrm{T}}, V_{v_2}\right), 0\right) \tag{5}$$

where V_{v1} and V_{v2} are the semantic vectors of capsule vertices v_1 and v_2, respectively; $k_{\mathrm{node}}(v_1, v_2)$ is calculated on the basis of the dot product represented by the word vector, considering the distance of the word embedding.

The graph kernel measures the similarity by comparing the substructures between the two graphs. The news text with longer documents contains more structural information. Hence, normalizing the graph kernel equation and introducing a normalization factor are necessary. Given the adjacency matrixes A_1, A_2 represented by the transformation graph of two documents, the value of each item in the adjacency matrix is equal to the weight of the corresponding edge. Given diagonal matrixes D_1, D_2 of the corresponding document, if the item on the diagonal side exists in the corresponding document, then the value of the item is set to 1. The normalization factor calculation formula is presented in Eq. (6):

$$\mathrm{norm} = \left\|A_1 + D_1\right\|_{\mathrm{F}} \cdot \left\|A_2 + D_2\right\|_{\mathrm{F}} \tag{6}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm of the matrix.

k_{walk} can be expressed as the product of the vertices along the wandering vertices and the kernels on the

edges. Thus, k_{walk} can be calculated on the basis of the original vertices, target vertices, and the edges connecting them. Let v_1 and v_2 denote the two capsule vertices in graph G_1, and e_1 is the edge between v_1 and v_2. Let u_1 and u_2 denote the vertices of the capsule in graph G_2, and e_2 is the edge between u_1 and u_2. $k_{\text{walk}}(e_1, e_2)$ is defined in Eq. (7).

$$k_{\text{walk}}(e_1, e_2) = k_{\text{node}}(v_1, u_1) \times k_{\text{edge}}(e_1, e_2) \times k_{\text{node}}(v_2, u_2) \tag{7}$$

where k_{node} is the kernel function defined above and k_{edge} is the kernel function used to compare two edges. The definition of k_{edge} is shown in Eq. (8).

$$k_{\text{edge}}(e_1, e_2) = l(e_1) \times l(e_2) \tag{8}$$

where $e_1 \in E_1, e_2 \in E_2$; $l(e_1)$ and $l(e_2)$ are the weights of edges e_1 and e_2, respectively. Therefore, the similarity value $k(d_1, d_2)$ between the two documents d_1 and d_2 can be obtained, and the value of $k(d_1, d_2)$ is between [0, 1].

3.3.2 Topic generation algorithm

Input document $D = \{d_1, d_2, \ldots, d_m\}$ is processed according to the order and gradually generates clusters. If the similarity between news document d_i and the cluster center exceeds the predefined threshold δ, then the news document is added to the most similar cluster; otherwise, a new cluster containing the document is created. Finally, the algorithm returns a set of topic clusters $\{T_1, T_2, \ldots, T_K\}$, and each cluster T_i represents a topic. Algorithm 3 introduces the process of the topic generation algorithm.

Algorithm 3 Topic generation algorithm

Input: $D = \{d_1, d_2, \ldots, d_m\}$ // news documents;
δ //similarity threshold
Output: $T = \{T_1, T_2, \ldots, T_K\}$ // topic clusters

for $i = 1, 2, \ldots, m \& d_i \in D$ **do**
construct the CSG_i of d_i;
if $T = \varnothing$ **then**
create T_1;
let $d_i \in T_1$;
represent T_i by CSG_i;
end if
if $T = \varnothing$ **then**
for each cluster T_k **do**;
calculate the similarity $\text{sim}(d_i, T_k)$ by graph kernel;
let $\max S = \max \text{sim}(d_i, T_k)$;
let $\max T = T_k | \text{sim}(d_i, T_k) = \max S$;
end for
if $\max S \geqslant \delta$ **then**
let $d_i \in \max T$;
recalculate the representation of $\max T$ by the centroid;
end if
if $\max S < \delta$ **then**
create a new cluster T_{new};
let $d_i \in T_{\text{new}}$;
represent T_{new} by CSG_i;
end if
end if
i++
end for
return $\{T_1, T_2, \ldots, T_K\}$

4 Experiment

4.1 Dataset

Three standard datasets are used for the experiment, namely, CEC (Chinese Emergency Corpus)[25], 20newsgroup, and Tsinghua University Chinese News Text Classification (THUCNews)[26]. The CEC corpus is constructed by Shanghai University and collects news reports on five topics from the Internet, namely, earthquakes, fires, traffic accidents, terrorist attacks, and food poisoning. It contains 332 documents. The 20newsgroup dataset is a well-known English dataset containing 20 000 news documents, evenly distributed to 20 different news categories. THUCNews is a large Chinese news dataset created by Tsinghua University. It uses a subset of the THUCNews dataset, which contains 100 000 news documents and is evenly distributed into 10 categories to test the operating efficiency of the proposed model and baseline method.

4.2 Evaluation metric

For the evaluation of topic detection, various evaluation indicators are generally used.

(1) Precision (P), Recall (R), and F1 score. These three indicators are the most widely used for evaluating topic detection performance, and the calculation formula is shown below.

$$\begin{cases} P = \dfrac{\text{TP}}{\text{TP} + \text{FP}}; \\ R = \dfrac{\text{TP}}{\text{TP} + \text{FN}}; \\ \text{F1} = \dfrac{2 \times P \times R}{P + R} \end{cases} \tag{9}$$

where TP represents a true case, FP represents a false positive example, and FN represents a false negative example.

(2) Normalized detection cost ($(C_{\text{det}})_{\text{norm}}$). $(C_{\text{det}})_{\text{norm}}$[27] is also widely used to evaluate topic detection performance. The smaller the $(C_{\text{det}})_{\text{norm}}$, the better the performance of the model. $(C_{\text{det}})_{\text{norm}}$ is

calculated, as shown in Eq. (10).

$$(C_{\text{det}})_{\text{norm}} = \frac{C_{\text{miss}} \cdot P_{\text{miss}} \cdot P_{\text{target}}}{\min\{(C_{\text{miss}} \cdot P_{\text{target}}), [C_{\text{fa}} \cdot (1 - P_{\text{target}})]\}} + \frac{C_{\text{fa}} \cdot P_{\text{fa}} \cdot (1 - P_{\text{target}})}{\min\{(C_{\text{miss}} \cdot P_{\text{target}}), [C_{\text{fa}} \cdot (1 - P_{\text{target}})]\}} \tag{10}$$

where C_{miss} is the missed detection rate, C_{fa} is the error rate, and P_{target} is a prior probability, they are predefined parameters, here set to 1.0, 0.1, 0.02, respectively; P_{miss} and P_{fa} represent the conditional probability of FNs and FPs, respectively. The calculation formula is presented below.

$$\begin{cases} P_{\text{miss}} = \dfrac{\text{TN}}{\text{TP} + \text{FN}}; \\ P_{\text{fa}} = \dfrac{\text{FP}}{\text{FP} + \text{TN}} \end{cases} \tag{11}$$

where TN represents a true negative example. These parameter settings are the best values selected by Allan[28] after many experiments, and many subsequent studies have used the same parameter settings.

4.3 Experimental parameter setting

In the construction of CSG, the community size (number of keywords contained in a capsule vertex) is predetermined by our experience, the minimum community size is set to 2, and the maximum community size is set to 6. The Word2Vec word embedding dimension is set to 300. In addition, the settings of parameters edge_min, max_betweenness, and min_similarity have a particularly important impact on the performance of the model. Their initial settings are shown in Table 1. The optimal value of the parameter will be discussed in detail in Section 4.7.

4.4 Comparison model

We designed the following comparison model to evaluate the performance of CSG.

JS-IDF: Zhou et al.[29] proposed a news event detection model on the basis of JS-IDF, using Jaccard similarity coefficient and IDF for feature selection.

LDA-VSM: Qiu et al.[14] proposed a topic detection method on the basis of VSM and the improved LDA hybrid model to calculate text similarity.

Hierarchical semantic graph model (HGM): HGM is a topic mining method based on hierarchical semantic graph model proposed by Zhang et al.[16] The construction of a semantic graph is mainly based on hierarchical feature word extraction.

Table 1 Initial parameter setting.

Step	Parameter	Value
Keyword graph construction	edge_min	1
Community detection	max_betweenness	1
Document clustering	min_similarity	0.4

WebKey: Rasouli et al.[30] constructed a word co-occurrence graph by identifying words that suddenly appear in news documents. The nodes and edges were weighted according to the number of word occurrences in the burst interval and document frequency.

4.5 Experiment 1

Experimental content: Experiments with the above models on the three datasets of CEC, 20newsgroup, and THUCNews are compared to verify the effectiveness of the CSG model.

Experimental results: Figure 3 displays the performance comparison of the CSG model and other news topic detection models on the three datasets.

On the three datasets, *P*, *R*, and F1 values of the probability-based LDA-VSM model are significantly higher than those of the word-based JS-IDF model. The reason is that LDA-VSM uses VSM and an improved LDA model for hybrid modeling. It unearths potential semantic information and has better performance. *P*, *R*, and F1 values of the two graph analysis models HGM and WebKey are slightly higher than those of the probability model LDA-VSM. However, *P*, *R*, and F1 values of LDA-VSM are slightly higher than those of HGM on the THUCNews dataset. *P*, *R*, and F1 values of the CSG model are higher than those of the probability method and the graph analysis method, reaching 81.44%, 76.60%, and 78.95% on the CEC dataset, respectively; on the 20newsgroup dataset, they are 82.88%, 70.60%, and 76.25%, respectively; On the THUCNews dataset, they are 83.01%, 69.88%, and 75.88%, respectively. The CSG method has obvious advantages over the word-based and probability methods and has the best performance on the three datasets.

In addition, the $(C_{\text{det}})_{\text{norm}}$ value results of all methods in the three datasets are shown in Table 2. The results in Table 2 indicate that the CSG model has the smallest $(C_{\text{det}})_{\text{norm}}$ value on the three datasets. Compared with all the comparison models, CSG has the best performance. Under different size datasets, the results are similar, showing that the CSG model has not only good performance but also strong robustness.

4.6 Experiment 2

Experimental content: A comparative experiment is conducted on the THUCNews dataset, whose size is

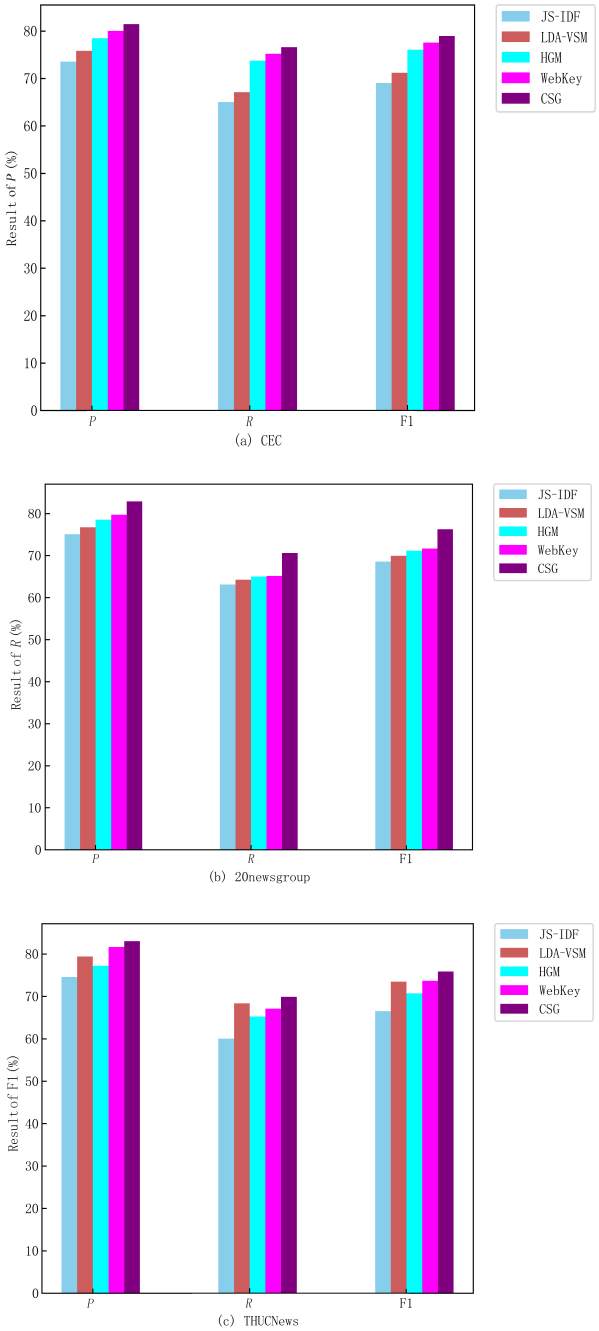


Fig. 3 Performance of different models on three datasets. (a), (b), and (c) correspond to three datasets respectively. The abscissa represents the three evaluation metrics of *P*, *R*, and F1 corresponding to the current dataset, and the ordinate represents the values of the three evaluation metrics.

set from 10 000 to 100 000 to evaluate the operating efficiency of the CSG model.

Experimental results: As illustrated in Fig. 4, the running time of the LDA-VSM model is significantly higher than that of other methods, indicating that the probabilistic method has the highest time complexity mainly because of the complicated reasoning process in

Table 2 $(C_{\text{det}})_{\text{norm}}$ of all methods on three datasets.

Method	CEC	20newsgroup	THUCnews
JS-IDF	0.1562	0.1603	0.1620
LDA-VSM	0.1437	0.1483	0.1498
HGM	0.1076	0.1104	0.1007
WebKey	0.0901	0.0925	0.0998
CSG	0.0852	0.0813	0.0894

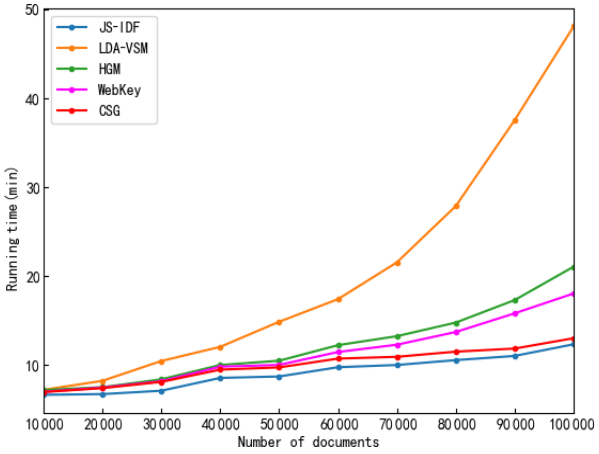


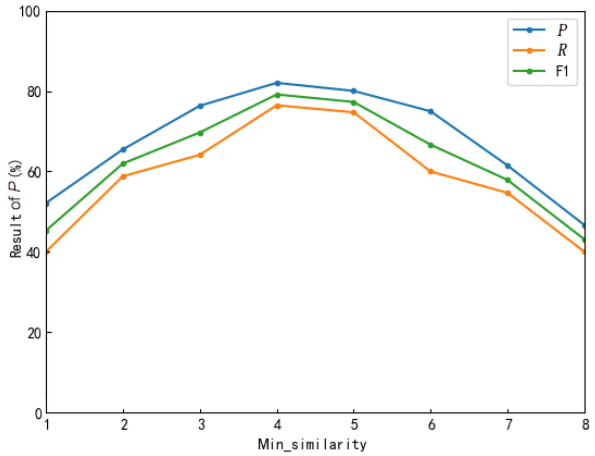
Fig. 4 Run time comparison results.

the model. The running time of the HGM model and the WebKey model is higher than that of the JS-IDF model and grows in a nonlinear manner mainly because the construction of the graph is based on the entire corpus, which consumes time. By contrast, the running time of the CSG model is between the JS-IDF model and the WebKey model. The reason is that the single-pass clustering in the last step is a clustering algorithm with low time complexity. In addition, the construction process of the keyword graph is based on each document, making the document presentation process further effective. The results suggest that although the running time of CSG is slightly higher than that of the JS-IDF model, it is still effective.

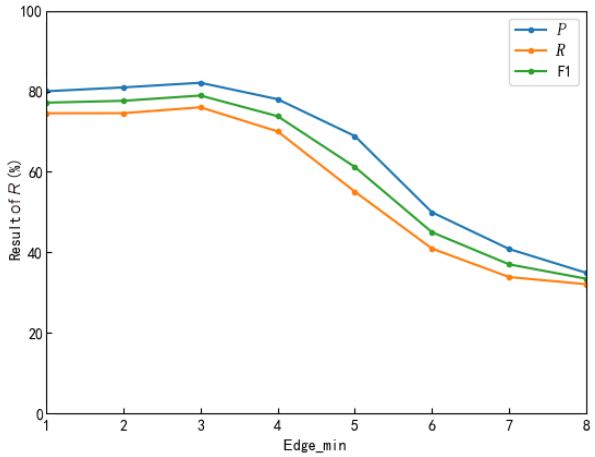
4.7 Experiment 3

Experimental content: A comparative experiment is conducted on the CEC dataset to evaluate the impacts of different parameters on the performance of the CSG model.

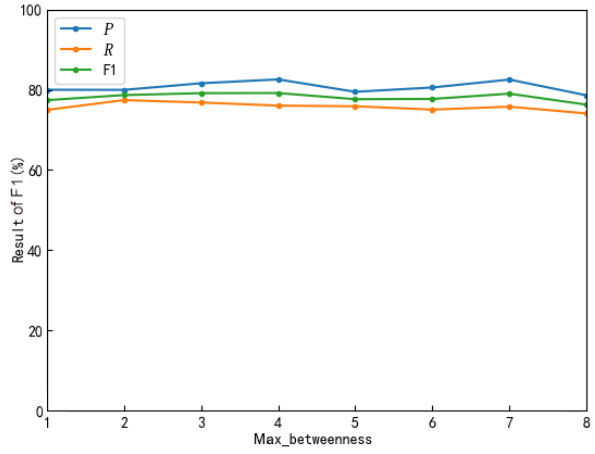
Experimental results: Figure 5 shows the effects of parameters edge_min, max_betweenness, and min_similarity on the performance of the CEC dataset. Figure 5a illustrates the performance of CSG with min_similarity on the CEC dataset. As min_similarity increases, the *P*, *R*, and F1 values on the CEC dataset first increase and then decrease. When min_similarity is 0.4, the best performance on the CEC dataset can be



(a)



(b)



(c)

Fig. 5 Parameters sensitivity analysis of CSG on CEC. The abscissa represents the values of different parameters and the ordinate represents the values of three evaluation metrics.

obtained. Figure 5b shows the performance of CSG with edge_min on the CEC dataset. The min_similarity on the CEC dataset is set to 0.4, and the max_betweenness on the CEC dataset is still set to the initial value. The results show that on the CEC dataset, the *P*, *R*, and F1 values reach the best performance when edge_min is 3, and the performance drops slightly when edge_min is less than 3. However, when edge_min exceeds 3, the *P*, *R*, and F1 values decrease significantly. Moreover, in news documents, the probability of a pair of words co-occurring in more than three sentences is small. If the value of edge_min is set too large, then many edges of the keyword graph will be truncated, which may affect the construction result of the keyword graph. Therefore, the value of edge_min should not be set too large. Figure 5c displays the performance of CSG with the max_betweenness on the CEC dataset. The min_similarity on the CEC dataset is set to 0.4, and edge_min is set to 3. Furthermore, *P*, *R*, and F1 values of CSG only slightly change with the change of max_betweenness, which indicates that the size of the max_betweenness parameter has little effect on the performance of CSG.

5 Conclusion

Aiming to address the problems of semantic sparsity and high time complexity in current news topic detection models, a news topic detection model based on CSG is proposed. By modeling relationships among words, a document is represented as a CSG, which overcomes the problem of semantic sparseness in word-based methods. The similarity among the semantic graphs of the capsule is measured by the graph core, and the similarity measurement among documents is transformed into the similarity measurement among graphs. The experimental results on three datasets show that the model accuracy is significantly better than word-based and probability methods. The experimental results on the large THUCNews dataset reveal that the time complexity of the model is lower than that of probabilistic and other graph analysis methods.

In the next step, we will consider how to use the Bidirectional Encoder Representation from Transformers (BERT) pretraining model to generate pretraining encoding vectors of words to better represent the text, thereby achieving the purpose of improving topic detection effects. How to track the evolution of topics over time and how to detect fine-grained topics are also issues that must be studied in future.

References

[1] J. Allan, S. Harding, D. Fisher, A. Bolivar, S. Guzman-Lara,

and P. Amstutz, Taking topic detection from evaluation to practice, in *Proc. 38th Annu. Hawaii Int. Conf. on System Sciences*, Big Island, HI, USA, 2005, p. 101a.

[2] Y. Chen and L. Liu, Development and research of Topic Detection and Tracking, in *2016 7th IEEE International Conference on Software Engineering and Service Science* (*ICSESS*), pp. 170–173, 2016.

[3] L. Hong and B. W. Li, Hot topic detection research of internet public opinion based on affinity propagation clustering, in *Computer, Informatics, Cybernetics and Applications*: *Proceedings of the CICA 2011*, X. G. He, E. T. Hua, Y. Lin, and X. Z. Liu, eds. Dordrecht, Netherlands: Springer, 2012, pp. 261–269.

[4] T. Sakaki, M. Okazaki, and Y. Matsuo, Tweet analysis for real-time event detection and earthquake reporting system development, *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 919–931, 2013.

[5] X. F. Lu, X. Zhou, W. T. Wang, P. Lio, and P. Hui, Domain-oriented topic discovery based on features extraction and topic clustering, *IEEE Access*, vol. 8, pp. 93648–93662, 2020.

[6] J. Z. Li, Q. N. Fan, and K. Zhang, Keyword extraction based on tf/idf for Chinese news document, *Wuhan Univ. J. Nat. Sci.*, vol. 12, no. 5, pp. 917–921, 2007.

[7] K. K. Bun and M. Ishizuka, Topic extraction from news archive using TF*PDF algorithm, in *Proc. 3rd Int. Conf. on Web Information Systems Engineering*, Singapore, 2002, pp. 73–82.

[8] S. Chen and Z. Jin, Weibo topic detection based on improved TF-IDF algorithm. *Science & Technology Review*, vol. 34, no. 2, pp. 282–286, 2016.

[9] R. Mihalcea and P. Tarau, TextRank: Bringing order into text, in *Proc. Conf. on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 404–411.

[10] K. Zhang, J. Zi, and L. G. Wu, New event detection based on indexing-tree and named entity, in *Proc. 30th Annu. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Amsterdam, Netherlands, 2007, pp. 215–222.

[11] M. Pu, F. Zhou, J. J. Zhou, X. Yan, and L. J. Zhou, Topic sentence extraction of key news events based on weighted textrank, (in Chinese), *Comput. Eng.*, vol. 43, no. 8, pp. 219–224, 2017.

[12] X. T. Qu, J. Yang, B. Wu, and H. M. Xin, A news event detection algorithm based on key elements recognition, in *Proc. 2016 IEEE 1st Int. Conf. on Data Science in Cyberspace* (*DSC*), Changsha, China, 2016, pp. 394–399.

[13] Z. Y. Chen and B. Liu, Mining topics in documents: Standing on the shoulders of big data, in *Proc. 20th ACM SIGKDD Int. Conf. on Knowledge Discovery And Data Mining*, New York, NY, USA, 2014, pp. 1116–1125.

[14] L. Q. Qiu, H. Y. Liu, X. Fan, and W. Jia, Hot topic detection based on VSM and improved LDA hybrid model, in *Proc. 12th Int. Conf. on Genetic and Evolutionary Computing*, Changzhou, China, 2019, pp. 583–593.

[15] H. Sayyadi and L. Raschid, A graph analytical approach for topic detection, *ACM Trans. Internet Technol.*, vol. 13, no. 2, p. 4, 2013.

[16] T. T. Zhang, B. Lee, Q. H. Zhu, X. Han, and E. M. Ye, Multi-dimension topic mining based on hierarchical semantic graph model, *IEEE Access*, vol. 8, pp. 64820–64835, 2020.

[17] A. Hamm, J. Thelen, R. Beckmann, and S. Odrowski, TeCoMiner: Topic discovery through term community detection, arXiv preprint arXiv: 2103.12882, 2021.

[18] M. N. Azadani, N. Ghadiri, and E. Davoodijam, Graph-based biomedical text summarization: An itemset mining and sentence clustering approach. *Journal of Biomedical Informatics*, vol. 84, pp. 42–58, 2018

[19] B. Drury, C. Rocha, M.-F. Moura, and A. Lopes, The extraction from news stories a causal topic centred bayesian graph for sugarcane, in *Proceedings of the 20th International Database Engineering & Applications Symposium*, Montreal, Canada, pp. 364–369, 2016.

[20] U. Kang, H. H. Tong, and J. M. Sun, Fast random walk graph kernel, in *Proceedings of the 12th SIAM international conference on data mining* (*SDM*), Los Angeles, CA, USA, pp. 828–838, 2012.

[21] N. Shervashidze and K. M. Borgwardt, Fast subtree kernels on graphs, in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, eds. Red Hook, NY, USA: Curran Associates Inc., pp. 1660–1668, 2009.

[22] G. Nikolentzos, P. Meladianos, F. Rousseau, M. Vazirgiannis, and Y. Stavrakas, Shortest-path graph kernels for document similarity, in *Proc. 2017 Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017, pp. 1890–1900.

[23] M. E. J. Newman, Detecting community structure in networks, *Eur. Phys. J. B*, vol. 38, no. 2, pp. 321–330, 2004.

[24] T. Mikolov, I. Sutskever, C. Kai, G. Corrado, and J. Dean, Distributed representations of words and phrases and their compositionality, in *Proc. 26th Int. Conf. on Neural Information Processing Systems*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[25] X. J. Zhang, Z. T. Liu, W. Liu, J. H. Yang, and S. N. Fei, Chinese event classification for event ontology construction, *J. Comput. Inf. Syst.*, vol. 9, no. 9, pp. 3511–3519, 2013.

[26] M. S. Sun, J. Y. Li, Z. P. Guo, Y. Zhao, Y. B. Zheng, X. C. Si, and Z. Y. Liu, THUCTC: An efficient Chinese text classifier, (in Chinese), https://github.com/diuzi/THUCTC, 2016.

[27] J. G. Fiscus and G. R. Doddington, Topic detection and tracking evaluation overview, in *Topic Detection and Tracking: Event-Based Information Organization*, Dordrecht, Netherlands: Kluwer Academic Publishers, 2002, pp. 17–31.

[28] J. Allan, R. Papka, V. Lvrenko, On-line new event detection and tracking, http://omega.sp.susu.ru/books/acm_sigmod/vol2/is3/SIGIR1998/P037.pdf, 2017.

[29] P. P. Zhou, Z. Cao, B. Wu, C. Z. Wu, and S. Q. Yu, EDM-

JBW: A novel event detection model based on *JS-ID'F*$_{order}$ and Bikmeans with word embedding for news streams, *J. Comput. Sci.*, vol. 28, pp. 336–342, 2018.

[30] E. Rasouli, S. Zarifzadeh, and A. J. Rafsanjani, WebKey: A graph-based method for event detection in web news, *J. Intell. Inf. Syst.*, vol. 54, no. 3, pp. 585–604, 2020.

Yan Tang received the MS degree in computer science from Southwestern Normal University, Chongqing, China, in 1991. Currently, she is a full professor in the School of Computer and Information Science of Southwest University, Chongqing, China. She has worked as a visiting scholar at Hiroshima City University, Hiroshima, Japan, in 1997 and Deakin University, Burwood, Australia, in 2002. She has published more than 50 academic papers in academic journals at home and abroad, co-published two academic works, and presided over and researched more than 10 national, provincial, and ministerial funds and research projects. She is mainly engaged in the research of intelligent science and web application technology.

Shuang Yang is currently pursing the BS degree at the School of Computer and Information Science, Southwest University, Chongqing, China. Her main research direction is natural language processing, including text classification and topic detection.