Information Sciences 570 (2021) 262–277

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A graphical decomposition and similarity measurement approach for topic detection from online news

Kejing Xiao [a], Zhaopeng Qian [b], Biao Qin [a,*]

[a] School of Information, Renmin University of China, Beijing 100872, China
[b] School of Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China

ARTICLE INFO

Article history:
Received 17 July 2020
Received in revised form 13 February 2021
Accepted 3 April 2021
Available online 20 April 2021

Keywords:
Topic detection
Word co-occurrence
Capsule semantic graph
Graph kernel
Community detection

ABSTRACT

Topic detection aims to discover valuable topics from the massive online news. It can help people to capture what is happening in real world and alleviate the burden of information overload. It also has great significance since the online news is experiencing an explosive growth. Topic detection is typically transformed into a document clustering problem, whose core idea is to cluster news documents that report on the same topic to the same group based on document similarity. Due to the complex structure and long length of news documents, the similarity measurement of news is very challenging. Existing term-based methods represent news documents based on a set of informative keywords in the document with a vector space model (VSM) and then the relationship between documents is calculated by cosine similarity. However, VSM ignores the relationship between words and has sparse semantics, which leads to low precision of topic detection. In recent years, the probabilistic methods and the graph analytical methods have been proposed for topic detection. However, both of them have high time complexity. To cope with these problems, we first present a novel document representation approach based on graphical decomposition, which decomposes each news document into different semantic units and then relationship between the semantic units is constructed to form a capsule semantic graph (CSG). The CSG can retain the relationship between words and alleviate the sparse semantics compared to VSM representation. We next introduce the graph kernel to measure the similarity between the CSGs based on their substructures. Finally, we use an incremental clustering method to cluster the news documents, in which the documents are represented by CSGs and the similarity between documents is calculated by graph kernel. The experiment results on three standard datasets show that our method obtains higher precision, recall and F1 score than several state-of-the-art methods. Moreover, the experiment results on a large news dataset show that our CSG-SM has lower time complexity than probabilistic methods and graph analytical methods.

1. Introduction

The emerging online news platforms provide a new way for people to acquire information. However, the explosive growth of various news on these platforms makes people overwhelmed in the massive information and difficult to find valuable topics. Therefore, how to discover meaningful topics or important events from the various online news becomes an

[*] Corresponding author.
E-mail address: biaoqin.cs@hotmail.com (B. Qin).

https://doi.org/10.1016/j.ins.2021.04.029
0020-0255/© 2021 Elsevier Inc. All rights reserved.

important task, which is called Topic Detection (TD). Topic detection is a subtask of Topic Detection and Tracking (TDT) and has attracted a lot of attention in recent years [19]. Generally, a topic can be regarded as something non-trivial happening at a specific time or place [39]. It has many applications such as online reputation monitoring [43], emergency management [35] and public opinion monitoring [21].

Although topic detection has been studied for many years, it is still an open problem [20]. Topic detection is a typical unsupervised task [11] whose core idea is to aggregate various articles (probably in different narrative forms and from different sources) that report the same breaking news into the same cluster. News documents in a cluster describe the same topic (event) and each cluster is regarded as a topic. One of the most important issues in topic detection is to identify the relationship between news documents, which usually includes two steps: how to represent news documents and how to measure the similarity between news documents based on the representation. News documents have the following characteristics: (1) have longer length and more complex logical structure than short text (e.g. tweets); (2) contain rich information and semantics; (3) come from various sources and have different narrative forms. Therefore, the representation of news documents and similarity measurement based on the representation are of great challenges in the field of topic detection.

Several approaches have been proposed for topic detection from news and the most prevailing approach is based on document clustering, in which the similarity between documents is calculated and a threshold is applied to determine whether two documents are belong to a same topic. In previous works, the dominant techniques for news representation are based on VSM [37], which represents the document by a group of informative keywords that extracted by Term Frequency-Inverse Document Frequency (TF-IDF) [36], Textrank [25] or Named Entities Recognition (NER) [49]. However, VSM ignores the relationship between words and suffers the problem of sparse semantics, which leads to low precision of topic detection. Since the terms are isolated in the VSM representation, the topic detection methods based on VSM representation are also called term-based methods. In practice, people often use the words relations rather than isolated words to understand the content of the news [4]. Therefore, it is not rational to completely ignore words relations. Besides, probabilistic approaches such as Latent Dirichlet Allocation (LDA) [7] and Probabilistic Latent Semantic Analysis (PLSA) [16] are proposed to deal with the above problems. In these methods, each document is represented as a finite mixture over a set of latent topics, where each topic is characterized by a distribution over words. Then documents are clustered based on their distributional similarity to generate topics. Although the probabilistic approaches take into account the interaction between words, the inference procedure in the model is too complex and leads to high time complexity. Moreover, some new researches proposed the graph analytical approaches for topic detection and achieved great success [39,48,12]. In these researches, words in documents are constructed into a graph structure based on their co-occurrence frequency in documents and then the whole graph is divided into several subgraphs. Each subgraph can be regarded as a topic after division and all the documents are assigned to the most related subgraph (topic). The graph analytical approaches break through the idea of traditional methods that consists of two steps (i.e., document representation and similarity measurement) and achieve good performance. However, it also has a high time complexity and is not suitable for large datasets.

In this paper, we propose a novel topic detection method based on incremental clustering to cluster the news documents into topics, in which the documents are represented by CSGs and the similarity between documents is calculated by graph kernel. The similarity threshold for the clustering process is determined after trying several times and the initial value is set according to our experience that is determined by the preliminary experiments. In our framework, each document is modeled into a keyword graph by keywords co-occurrence and then the keyword graph is decomposed into several subgraphs by community detection. Each subgraph contains a set of closely related keywords and seems like a capsule after community detection. So we call it ”capsule vertex” in this paper. The relationship (edges of the graph) between the capsule vertices is constructed by their semantic vectors. The capsule vertices and the edges together make up the capsule semantic graph (CSG), which can effectively retain the relationship between words as well as the semantic information of documents. In addition, the similarity measurement between graphs is also a challenging task. In recent years, graph kernel has emerged as a powerful tool for similarity measurement between graphs [29], which measures the similarity between a pair of graphs based on their substructures. In this paper, we introduce the graph kernel for similarity measurement between CSGs. Our topic detection method is named Capsule Semantic Graph and Similarity Measurement (CSG-SM).

To the best of our knowledge, we are the first to combine graph based document representation and graph similarity measurement for topic detection from news documents. Our model can detect more accurate topics efficiently. The main contributions of this paper are as follows:

(1) We propose a novel document representation method based on graphical decomposition to transform the document into a Capsule Semantic Graph (CSG), in which each capsule vertex contains a set of closely related keywords, and the words in the vertex serve as features to compare the relationship between vertices. The CSG can effectively capture the words relations and important semantic units of the document, as well as alleviate the semantic sparsity of traditional VSM representation.
(2) We introduce a novel graph kernel to measure the similarity between CSGs according to its characteristics. The comparison of CSGs is decomposed into the comparison between their substructures. Finally, the similarity between news documents is obtained by the similarity between their corresponding CSGs.
(3) We perform extensive experiments on three standard datasets and the experiment results show that our method obtains higher precision, recall and F1 score than several state-of-the-art methods. Moreover, the experiment results

on a large news dataset show that our method has lower time complexity than probabilistic methods and graph analytical methods.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 describes the framework of our approach in detail. Section 4 shows the experiment and results. Conclusion and future work are summarized in Section 5.

2. Related works

2.1. Topic detection

This paper is dedicated to topic detection from news. Methods for this problem can be divided into three categories: term-based approaches, probabilistic approaches and graph analytical approaches.

At the early stage, topic detection methods are mostly the term-based approaches that typically rely on document clustering. In term-based approaches, terms are firstly extracted as document features to represent the documents by using the vector space model (VSM). After that, news documents that describing the same topic are clustered together according to a certain clustering criterion. This kind of method is called term-based method for terms are isolated in the VSM representation. For example, Yang et al. [47] proposed an augmented Group Average Clustering (GAC) method to cluster documents into topics. Masand et al. [24] used a K-Nearest Neighbors (KNN) method to classify news stories. They all represented the documents by the VSM representation. Besides, some variants of TF-IDF and VSM representations are also used for topic detection [10,46,50]. Term-based approaches were the most prevailing approaches for topic detection at the early stage. However, the VSM representation in term-based methods ignores the relationship between words and has sparse semantics, which leads to low precision of topic detection.

The next generation of topic detection often uses probabilistic approaches [3,18,38]. The most widely used probabilistic approach is LDA [7], which represents each document as a finite mixture over a set of latent topics and each topic is characterized by a distribution over words. Then topics are generated by clustering documents based on their distributional similarity. LDA uses Variational Expectation Maximization (VEM) algorithm [7] or stochastic sampling inference based on Gibbs Sampling (GS) [44] to infer topic words and documents distribution. There are also many improved methods for LDA. For example, Andrzejewski et al. [5] incorporated domain knowledge into topic modeling. Hou et al. [17] presented the news as a link-centric heterogeneous network and used a unified probabilistic model for event search. Li et al. [20] proposed a probabilistic model that combined content and time information for topic detection. The probabilistic approaches have achieved great success in the topic detection field and performed better than traditional term-based approaches. However, due to the complex inference procedure in the model, its time complexity is very high.

Recently, some graph analytical approaches for topic detection have been proposed. For example, Sayyadi et al. [39] proposed the keygraph to represent the documents by words co-occurrence relationship, and then divided the keygraph into different parts by community detection. Each part can be regarded as a topic after community detection and the related documents can be assigned to each topic. Zhang et al. [48] proposed a hybrid term-term relations analysis approach to integrate semantic relations and co-occurrence relations for topic detection. Chen et al. [12] proposed to use WordNet as external knowledge and incorporate it to the co-occurrence graph. Their proposed method enriched the semantic information of graph structure for topic detection. Graph analytical approaches have provided a novel idea for topic detection and worked well on news datasets. The performance of graph analytical approaches is significantly higher than that of term-based approaches and slightly higher than that of probabilistic approaches. However, the main drawback is that it can make the algorithm very expensive for a large collection of documents when adding documents to the graph. The time consumption of graph analytical approaches is smaller than that of probabilistic approaches and still significantly higher than that of term-based approaches.

From the previous researches, we can conclude that the term-based methods have low topic detection precision. The probabilistic approaches and graph analytical approaches had achieved higher precision than term-based approaches. However, their time complexity is very high. When the dataset is large, the time consumption of the probabilistic approaches and graph analytical approaches will increase greatly. The previous topic detection methods fail to obtain both high precision and low time complexity at the same time. To solve this problem, we propose a novel method called CSG-SM. In CSG, each capsule vertex contains a set of closely related keywords and the words in the vertex serve as features to compare the relationship between vertices. Then we used the graph kernel to measure the similarity between CSGs according to the relationship between capsule vertices for document clustering. The CSG can retain the relationship between words and alleviate the sparse semantics compared to VSM representation, which can help us to obtain significant higher topic detection precision than the term-based approaches. Moreover, since the last step of our method is the single-pass clustering algorithm that has low time complexity, the time consumption of our CSG-SM is greatly less than that of probabilistic approaches and graph analytical approaches, and the precision of our CSG-SM is still higher than that of probabilistic approaches and graph analytical approaches.

2.2. Graph based document representation

Graph based document representation becomes popular in recent years. The graph based document representation can model the relationship between words and words, sentences and sentences, and even relationship between documents to make the document-related tasks more effective. Some works use words in the document as vertices, and construct edges based on word relationship. For example, Rousseauet al. [34] proposed a document representation method by graph-of-word that captures the co-occurrence relationship between words for the information retrieval task. Schenker et al. [40] used preceding relation of words to represent the inherent structure of document for the web document clustering. Other researches used sentences, paragraphs or documents as vertices. For example, Page et al. [30] established edges by hyperlinks between pages to rate web pages objectively for user navigation. Putra et al. [31] used sentence as vertices and the cosine similarity to measure the similarity between sentences. In addition, there are some researches used hybrid graph to represent documents. For example, Rink et al.[33] built the graph representation of the sentence that encodes lexical, syntactic and semantic information to detect causal event relations. Baker et al.[6] provided a new graph-based display of FrameNet annotation and made the complex data as a graph that separates out entities (nodes) from relations (edges) and clarifies which information is semantic, syntactic, or both. The existing researches show that the graph based document representation has been used in a variety of ways and achieved good results on a variety of tasks.

Recently, graph based document representation achieved great success in application of topic detection [39] [48] [12]. For example, the graph analytic approach proposed by Sayyadi et al. [39] can model words co-occurrence relation in the documents corpus by a graph. After that, the graph is divided into several parts by community detection. Finally, the documents are assigned to each community for topic detection. The graph analytic approach based on graph representation achieved higher precision than term-based methods and probabilistic methods. However, since the graph representation models all words in the document corpus into only one graph and these words need to be traversed many times for community detection, it will result in significant time consumption when the documents dataset is large.

Inspired by the existing graph based document representation methods, we designed a CSG representation for the topic detection task to deal with the problem of high time consumption. Different from modeling words in the corpus into a big graph by words co-occurrence in documents, we model each document by a small keywords graph by the word co-occurrence in sentences. For document clustering, different from assigning documents to each topic after community detection, we perform clustering by using single-pass and the graph kernel is used to compare the similarity between CSGs. The time complexity of our CSG-SM is greatly less than that of graph analytical approaches, and our method also achieves higher precision. The method proposed in this paper enriches both the researches of graph based document representation and topic detection from news.

2.3. Graph kernel

Similarity measurement between graphs has always been a challenging task. In recent years, graph kernel has been successfully used to measure the similarity between two graphs, which employed functions that compare substructures of graphs that are computable in polynomial time. Kernel can be considered as a method to measure the similarity between a pair of objects, while graph kernel is a method to measure the similarity between two graphs according to their substructures [28]. More specifically, various substructures have been proposed to focus on different structural aspects of graphs such as random walks [45], shortest paths [9], subtrees [32] and graphlets [42]. Graph kernel has been successfully applied to information retrieval [15], computational biology [41], chemistry [22], medicine [13], document classification [8] and other fields. Especially, the graph kernel proposed by Nikolentzos et al. [28] has been successfully applied to the document classification task, which is similar to the document clustering task in our work. Thus, we used the graph kernel proposed by Nikolentzos et al. [28] and realized the comparison of similarity between CSGs for the topic detection task.

3. The CSG-SM framework

3.1. Problem definition

In this section, we give some definitions and formally define the problem of topic detection. Generally, news documents from various sources form a news dataset $D = \{d_1, d_2, \cdots, d_m\}$, where d_i represents a news document that comprised by a sequence of words. A topic is usually composed of a set of news documents that report on it. Assume that we have a news documents dataset $D = \{d_1, d_2, \cdots, d_m\}$, our task is to divide the news dataset into multiple clusters $\{T_1, T_2, \cdots, T_n\}$, where each cluster $T_i = \{d_1, d_2, \cdots, d_k\}$ contains several documents that report on the same topic.

Topic detection has two major components, i.e. document representation and document clustering. Document representation is to transform documents into structured form that is suitable for topic detection task. For this purpose, we propose a new document representation method based on graph decomposition in this paper. Based on the document representation, a clustering method is usually applied for topic detection. In this paper, we use an incremental clustering algorithm based on single-pass clustering [2], because it is an efficient algorithm and does not need the prior knowledge of cluster number, which is quite hard to determine and has a big influence on the clustering results. In single-pass clustering, the algorithm performance depends on similarity threshold that is chosen by humans on prior knowledge. In our work, the initial similarity threshold of clustering is set by our experience, which is determined by the preliminary experiments. The optimal threshold is finally determined after several times trying and the threshold setting will be different on different datasets. More details

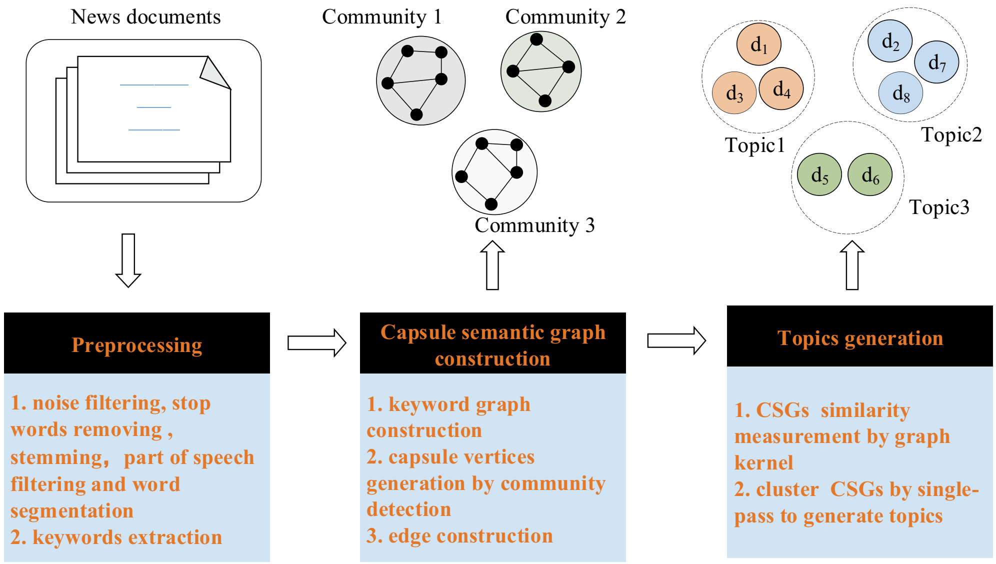

News documents
Community 1
Community 2
Community 3
Topic1
Topic2
Topic3
d1
d2
d3
d4
d5
d6
d7
d8
Preprocessing
1. noise filtering, stop words removing , stemming，part of speech filtering and word segmentation
2. keywords extraction
Capsule semantic graph construction
1. keyword graph construction
2. capsule vertices generation by community detection
3. edge construction
Topics generation
1. CSGs similarity measurement by graph kernel
2. cluster CSGs by single-pass to generate topics

Fig. 1. Overview of the CSG-SM framework.

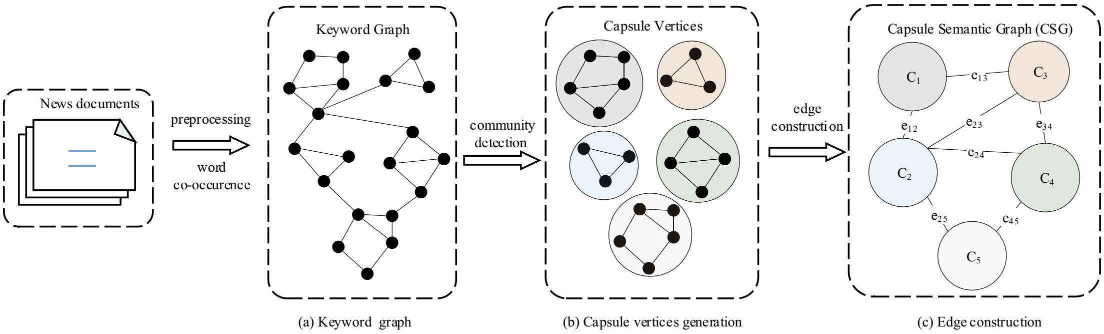


Fig. 2. The process of capsule semantic graph construction.

about the threshold setting and parameter sensitivity analysis are shown in Section 4.5.4. In addition, we use graph kernel to measure the document similarity based on the document representation for the document clustering.

The overview of our CSG-SM (Capsule Semantic Graph and Similarity Measurement) framework is shown in Fig. 1. The framework mainly consists of three components: preprocessing, document representation and document clustering. First, the input news documents are processed by filtering noise, removing stop words, stemming, part of speech filtering and word segmentation for Chinese dataset. Keywords of each document are extracted as document features by Textrank [25]. Second, each document is represented by a capsule semantic graph. The process includes keyword graph construction, capsule vertices generation by community detection and edge construction. Third, document are clustered by single-pass clustering algorithm. In the clustering process, graph kernel is used to measure the similarity between CSGs and finally topics can be generated.

3.2. Capsule semantic graph construction

The outline of the CSG construction and the details of the process are shown in Fig. 2. The process consists of three main components: (a) Keyword graph construction; (b) Capsule vertices generation and (c) Edges construction.

3.2.1. Keyword graph construction

The keyword graph generated by a document can be represented as $G = (V, E)$, where V represents a set of vertices and E represents a set of edges connecting these vertices. In the keyword graph, vertices are the keywords and edges are the relationship between keywords. The generated keyword graph is shown in Fig. 2(a), where the black dots represent keywords, the edges between black dots represent the relationship between keywords. The keyword graph construction mainly includes three steps:

Step1: Divide each preprocessed document d into sentences and the document is represented as $d = [s_1, s_2, \cdots, s_i]$. For each sentence $s_i \in d, s_i = [w_{i1}, w_{i2}, \cdots, w_{ij}]$, in which w_{ij} is the keywords in the sentences. The processed sentences act as the input of the subsequent components.

Step2: Create a vertex v_i for each keyword w_i. An edge e_{ij} is added between two keywords w_i and w_j if they co-occur in at least one sentence and the co-occurrence times is the weight of the edge. Thus, the co-occurrence graph is constructed for a document.

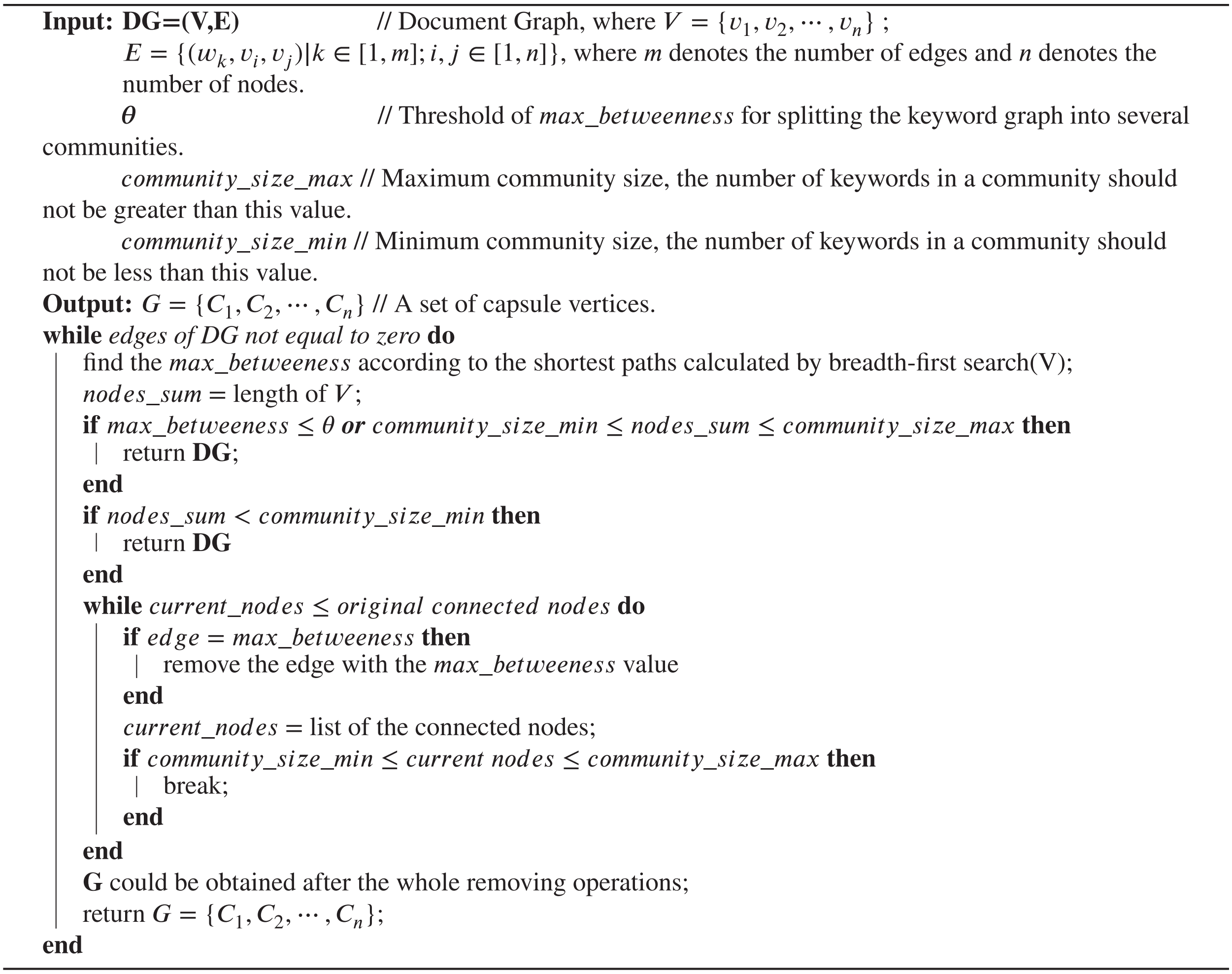
Step3: Calculate $SF_{e_{ij}}$ for each edge e_{ij} that represents the number of sentences containing both w_i and w_j in the document. Then we remove the edge if $SF_{e_{ij}}$ is less than the threshold $edge\_min$.

Thus, the final keyword graph can be obtained and the process is shown in Fig. 2(a).

3.2.2. Capsule vertices generation by community detection

The keyword graph reveals the relationship between keywords and each part of the graph has different density. We divide the keyword graph into several subgraphs by community detection. Words inside each subgraph are highly related, while the relationship between subgraphs is relatively weak. Each subgraph is regarded as a whole after community detection, which is called capsule vertex. Graph G represents the keyword graph of a document and can be represented as $G = \{C_1, C_2, \cdots, C_n\}$ after community detection, where n denotes the number of subgraphs and each subgraph C_i contains a set of closely related keywords. In this paper, the community detection algorithm proposed by Newman [27] is used to generate capsule vertices, which is a well-known community detection method and has been proved to have good performance. The process can effectively accelerate the calculation of the subsequent components. The capsule vertices generation process by community detection is shown in Fig. 2(b) and the algorithm is shown in **Algorithm 1**.

Algorithm 1. Capsule vertices generation by community detection



```
Input: DG=(V,E)          // Document Graph, where $V = \{v_1, v_2, \cdots, v_n\}$ ;
        $E = \{(w_k, v_i, v_j) | k \in [1, m]; i, j \in [1, n]\}$, where $m$ denotes the number of edges and $n$ denotes the
        number of nodes.
        $\theta$          // 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: $G = \{C_1, C_2, \cdots, C_n\}$ // A set of capsule vertices.
while edges of DG not equal to zero do
    find the $max\_betweeness$ according to the shortest paths calculated by breadth-first search(V);
    $nodes\_sum$ = length of $V$;
    if $max\_betweeness \le \theta$ or $community\_size\_min \le nodes\_sum \le community\_size\_max$ then
        return DG;
    end
    if $nodes\_sum < community\_size\_min$ then
        return DG
    end
    while $current\_nodes \le original\ connected\ nodes$ do
        if $edge = max\_betweeness$ then
            remove the edge with the $max\_betweeness$ value
        end
        $current\_nodes$ = list of the connected nodes;
        if $community\_size\_min \le current\ nodes \le community\_size\_max$ then
            break;
        end
    end
    G could be obtained after the whole removing operations;
    return $G = \{C_1, C_2, \cdots, C_n\}$;
end
```

The **Algorithm 1** includes four steps:

Step 1: The shortest paths is obtained by breadth-first search and the betweenness is calculated by the count of the shortest paths for all pairs of nodes in the network that pass through that edge.

Step 2: The *max_betweenness* is obtained according to the betweeness values and the edge corresponding to the *max_betweeness* is removed from the keyword graph.

Step 3: Repeat the step 1) and step 2) until no edges with a score greater than a threshold *max_betweenness* can be found. Or step 3) could be stopped when the stop condition is satisfied. The stop conditions are "$community\_size\_min \leqslant current\_nodes \leqslant community\_size\_max$", "$nodes\_sum < community\_size\_min$".

Step 4: The set of capsule vertices $\{C_1, C_2, \cdots, C_n\}$ is returned after several times removing operations.

3.2.3. Edges construction by vertices similarity

The multiple capsule vertices generated in Section 3.2.1 are independent of each other after community detection. To construct edges between capsule vertices, we take all the words in a capsule vertex as its features to calculate the similarity between capsule vertices and treat the similarity value as the weights of edges between capsule vertices. The similarity value is calculated by the semantic vector of capsule vertices. Semantic vector of the capsule vertex C_i is calculated by $\frac{1}{m}\sum_{i=1}^{m} v_i$ if C_i contains m words, in which the vector of each words v_i can be obtained by the off-the-shelf tool [26]. The similarity between capsule vertices C_i and C_j is calculated by Eq. (1):

$$sim(C_i, C_j) = \frac{V_{C_i} \cdot V_{C_j}}{\|V_{C_i}\| \cdot \|V_{C_j}\|} \tag{1}$$

Where V_{C_i} and V_{C_j} represent the vector of C_i and C_j. $\|\cdot\|$ represents the L2 norm. By Eq. (1), the edges with weights can be constructed between capsule vertices. Especially, vertices of CSG are the sets of keywords and they can represent the key information of a document. Besides, the edges of CSG are the relationship between the key information and the edges can represent the dependencies among these keyword sets. Consequently, the edges can represent the potential semantic relations between capsule vertices. The CSG can capture the dependencies and relationship of various key information and have powerful ability for document representation. The final CSG is shown in Fig. 2(c).

3.3. Topics generation by incremental clustering

3.3.1. Similarity measurement by graph kernel

In this section, we present a novel method for news documents similarity measurement based on the definition of graph kernel between pairs of CSGs. The graph kernel takes into account both the capsule vertices and the relationship between them. The CSG is an undirected weighted graph, in which each capsule vertex contains a set of keywords and the weights of the edges are the similarity between capsule vertices. Let G_1, G_2 denote the CSGs of two news documents d_1 and d_2. The graph kernel on $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is defined as Eq. (2).

$$k(d_1, d_2) = \frac{\sum\limits_{v_1 \in V_1, v_2 \in V_2} k_{node}(v_1, v_2) + \sum\limits_{e_1 \in E_1, e_2 \in E_2} k_{walk}(e_1, e_2)}{norm} \tag{2}$$

Where k_{node} is the kernel function for comparing two vertices, k_{walk} is the kernel function for comparing two walks (a walk represents a path of length 1 in graph G). Norm is the normalization factor calculated by Eq. (4). $k_{node}(v_1, v_2)$ is defined as Eq. (3).

$$k_{node}(v_1, v_2) = max(V_{v_1}^T \cdot V_{v_2}, 0) \tag{3}$$

Where V_{v_1}, V_{v_2} are the vector of capsule vertices v_1 and v_2. $k_{node}(v_1, v_2)$ is calculated by the distance of word embeddings. A normalization factor is introduced since the nominator of the proposed kernel depends on the length of the compared documents. Given the adjacency matrices A_1, A_2 (where the value of each entry in the adjacency matrix is the label of the corresponding edge) and the diagonal matrices D_1, D_2 (where the diagonal entries set to 1 if the corresponding term exists in the corresponding document) of the two CSGs, the normalization factor is calculated by Eq. (4).

$$norm = \|A_1 + D_1\|_F \cdot \|A_2 + D_2\|_F \tag{4}$$

Where $\|\cdot\|$ is the Frobenius norm for matrices.

Let v_1, v_2 denote two capsule vertices in graph G_1, e_1 be the edge between v_1 and v_2. Let u_1, u_2 denote the capsule vertices in graph G_2, e_2 be the edge between u_1 and u_2. $k_{walk}(e_1, e_2)$ is defined as Eq. (5).

$$k_{walk}(e_1, e_2) = k_{node}(v_1, u_1) \times k_{edge}(e_1, e_2) \times k_{node}(v_2, u_2) \tag{5}$$

Where k_{edge} is a kernel function for comparing two edges defined as Eq. (6).

$$k_{edge}(e_1, e_2) = l(e_1) \times l(e_2), \quad if \quad e_1 \in E_1, e_2 \in E_2 \tag{6}$$

Where $l(e_1)$ and $l(e_2)$ are the weights of edges e_1 and e_2. Thus, we can obtain the similarity value $k(d_1, d_2)$ between the two documents d_1 and d_2. The value of $k(d_1, d_2)$ lies in the interval [0, 1].

3.3.2. Single-pass clustering

For the news documents dataset, we apply the single-pass clustering to generate topics. The process of the single-pass clustering algorithm is shown in Algorithm 2. It sequentially processes the input documents $D = \{d_1, d_2, \cdots, d_m\}$ and generates clusters incrementally. A news document is added to the most similar cluster if the similarity between this document and the centroid of the cluster exceeds a pre-defined threshold δ. Otherwise, the document is regarded as the seed of a new cluster. The initial similarity threshold is set by our experience, which is determined by the preliminary experiments. The optimal threshold is finally determined after several times trying and the threshold setting will be different on different datasets. Finally, the algorithm returns the topic clusters $\{T_1, T_2, \cdots, T_K\}$ and each cluster T_i represents a topic.

Algorithm 2. Single-pass clustering

```
Input: $D = \{d_1, d_2, \cdots, d_m\}$    // News Documents
       $\delta$                               //Similarity Threshold
Output: $T = \{T_1, T_2, \cdots, T_K\}$ //Topic Clusters
for the i-th document $d_i$ do
    construct the $CSG_i$ of $d_i$;
    if $T=\emptyset$ then
        create $T_1$;
        let $d_i \in T_1$;
        represent $T_1$ by $CSG_i$;
    else
        for each cluster $T_k$ do
            calculate the similarity $sim(d_i, T_k)$ by graph kernel;
            let maxS = $\max_i sim(d_i, T_k)$ ;
            let maxT = $T_k \mid sim(d_i, T_k) = maxS$ ;
        end
        if $maxS \geq \delta$ then
            let $d_i \in maxT$ ;
            recalculate the representation of $maxT$ by the centroid ;
        else
            create a new cluster $T_{new}$;
            let $d_i \in T_{new}$;
            represent $T_{new}$ by $CSG_i$;
        end
    end
    i++
end
return $\{T_1, T_2, \cdots, T_K\}$
```

4. Experiment

4.1. Datasets

We use three standard datasets CEC[1], 20newsgroup[2] and THUCNews[3] to evaluate the performance of our CSG-SM and the baseline methods. In addition, we use a large news dataset THUCNews to test the running efficiency of these methods. The CEC is a small annotated Chinese dataset constructed by Shanghai University that contains 332 documents and consists of 5 topics

[1] https://github.com/shijiebei2009/CEC–Corpus.

[2] http://www.qwone.com/ jason/20Newsgroups/.

[3] http://thuctc.thunlp.org/.

Table 1
Initial parameter settings.

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

(including earthquakes, fires, traffic accidents, terrorist attacks and food poisoning). The 20newsgroup is a well known English dataset that contains 20 k news documents, which are evenly distributed to 20 different newsgroups. THUCNews is a very large Chinese news dataset constructed by Tsinghua University. We use a subset of the THUCNews that contains 100 k news documents and distributed to 10 categories to test the running efficiency of CSG-SM and the baseline methods. The datasets are preprocessed before experiment.

4.2. Baseline methods

We designed experiments to evaluate the performance of CSG-SM. The baseline methods are term-based methods, probabilistic methods and graph analytical methods. Term-based methods include GAC and KNN, whose documentation representation are based on VSM.The probabilistic methods include LDA-VEM and LDA-GS. We used single-pass and cosine similarity to cluster LDA vectors. The graph analytical methods include KeyGraph (KG) and KeyGraph+ (KG+).

(1) GAC. GAC (Group Average Clustering) is a popular algorithm for TDT task that was proposed by Yang et al. [47] in 1998.
(2) KNN. The KNN (k-Nearest Neighbor) is another popular solution for TDT task that was proposed by Masand et al. [24] in 1992.
(3) LDA-VEM. LDA-VEM is the original LDA proposed by Blei et al. [7] in 2003 and has been widely used in TDT task. LDA-VEM uses a variational expectation maximization (VEM) algorithm to infer topic words.
(4) LDA-GS. LDA-GS is a LDA variant with stochastic sampling inference based on Gibbs sampling (GS), which was proposed by Steyvers et al. [44] in 2007.
(5) KG (KeyGraph). KG is a graph analytical approach that was proposed by Sayyadi et al. [39] in 2013 and used keywords in each document as features.
(6) KG+ (KeyGraph +). KG+ is a variant of KG that was also proposed in [39]. The method adds noun phrases and named entities as features, and doubles their term frequencies to increases their weights.

4.3. Implementation details

In the capsule semantic graph construction process, the minimum community size (number of keywords contained in a capsule vertex) is set to 2 and the maximum size is set to 6. The community size is predefined by our experience, which is determined by the preliminary experiments. The dimension of word2vec is set to 300. Besides, the topics number K of the LDA-VEM and LDA-GS are set to 5, 20 and 10 respectively on the three datasets. The parameters α and β for LDA are set to 0.05 and 0.01, respectively. The maximum iteration times is set to 1000. We use the Stanford CoreNLP [23] for named entity recognition and word segmentation. Besides, our method involves several parameters and their initial settings are shown in Table 1.

The parameters in Table 1 are the initial values set by our preliminary experiments. In which, the minimum value of *edge_min* and *max_betweenness* are 1. The preliminary experiments shows that the two parameters have little influence on the final results when they were set to small values. So we set initial value of *edge_min* and *max_betweenness* to the minimum value 1. The value of *min_similarity* lies between 0 and 1. The preliminary experiments show that the variation of *min_similarity* has a great influence on the final results and the optimal *min_similarity* value is around 0.4. So we set the initial

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

Methods	$(C_{det})_{norm}$ on CEC	$(C_{det})_{norm}$ on 20newsgroup	$(C_{det})_{norm}$ on THUCnews
GAC	0.2265	0.2208	0.2423
KNN	0.1853	0.1950	0.2050
LDA-VEM	0.1625	0.1653	0.1688
LDA-GS	0.1472	0.1483	0.1520
KG	0.1105	0.1164	0.1185
KG+	0.0925	0.0945	0.1080
CSG-SM	0.0860	0.0820	0.0900

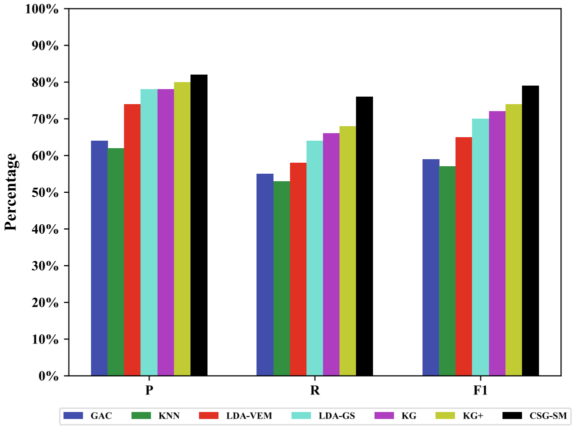


Fig. 3. Performance of the approaches on CEC dataset.

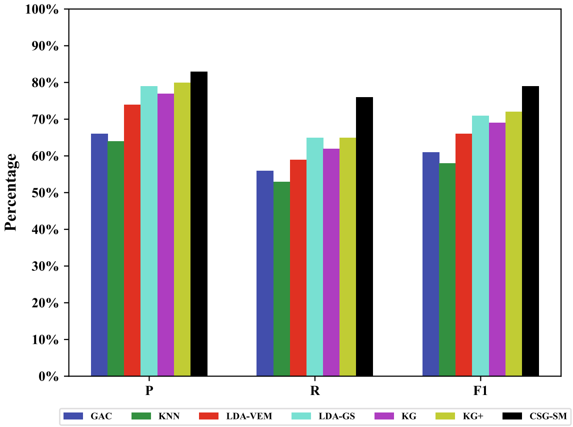


Fig. 4. Performance of the approaches on 20newsgroup dataset.

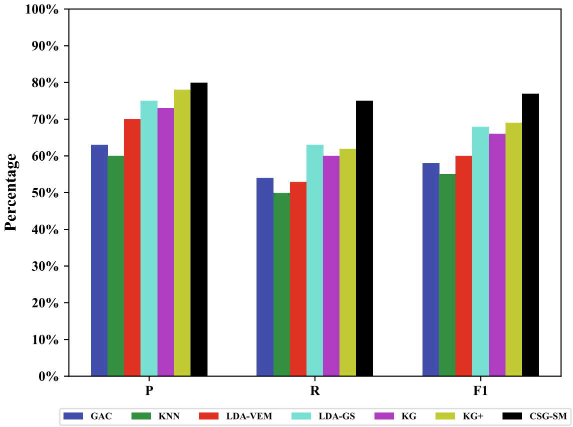


Fig. 5. Performance of the approaches on THUCNews dataset.

min_similarity value to 0.4. We then adjusted the thresholds values and carried out experiments on these values to find the optimal thresholds for three parameters. More details about parameter sensitivity analysis is shown in Section 4.5.4.

Table 3
Running time of KG-SM and CSG-SM (seconds).

Methods	CEC	20newsgroup	THUCNews
KG-SM	5.6	84	895
CSG-SM	2.9	44	468

4.4. Evaluation metrics

We use several metrics to evaluate the performance of our proposed CSG-SM and the baseline methods.

1) Precision (P), Recall (R) and F1 score. The three metrics are most widely used metrics to evaluate the performance of topic detection in previous works. Their calculation is shown in Eq. (7).

$$\begin{aligned} P &= \tfrac{TP}{TP+FP} \\ R &= \tfrac{TP}{TP+FN} \\ F1 &= \tfrac{2PR}{P+R} \end{aligned} \tag{7}$$

Where TP is the true positive cases, FP is the false positive cases, FN is the false negative cases.

2) $(C_{det})_{norm}$. The normalized detection cost function $(C_{det})_{norm}$ is also widely used to evaluate the system performance [14]. The smaller the value of $(C_{det})_{norm}$, the better performance of the system would be. $(C_{det})_{norm}$ is calculated by Eq. (8)

$$(C_{det})_{norm} = \frac{C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{non-target}}{min(C_{miss} \cdot P_{target}, C_{fa} \cdot P_{non-target})} \tag{8}$$

Where C_{miss} is the cost of miss rate, C_{fa} is the cost of false alarm rate, P_{target} is the a priori probability of finding a target as specified by the application and $P_{non-target} = 1 - P_{target}$.The miss rate P_{miss} and false alarm rate P_{fa} are calculated by Eq. (9).

$$\begin{aligned} P_{miss} &= \tfrac{TN}{TP+FN} \\ P_{fa} &= \tfrac{FP}{FP+TN} \end{aligned} \tag{9}$$

Where TN is the true negative cases.

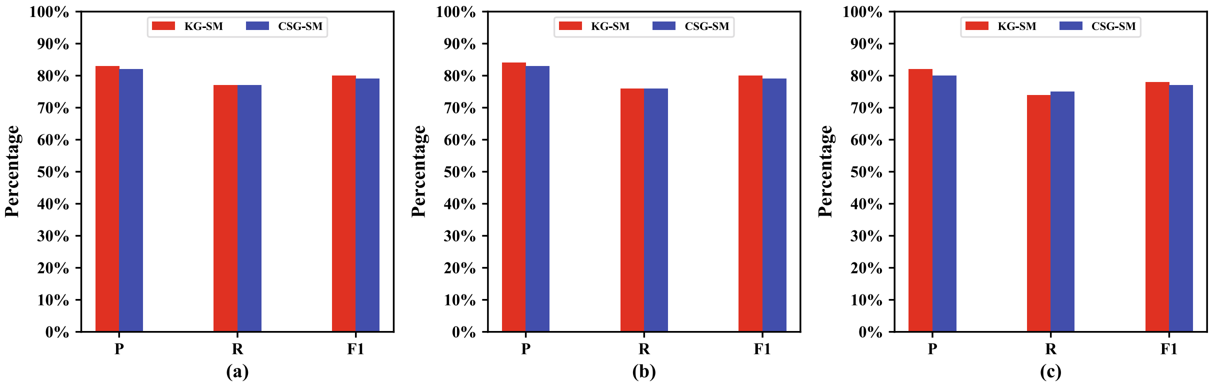
$C_{miss}, C_{fa}, P_{target}, P_{non-target}$ are predefined parameters. In this paper, these parameter are set to 1.0, 0.1, 0.02 and 0.98, respectively. These parameter settings are according to [1] and many subsequent studies have used the same parameter settings.

4.5. Results

4.5.1. Performance of topic detection

We compared the experimental results of the baseline methods and our CSG-SM on CEC, 20newsgroup and THUCNews datasets. The results on CEC, 20newsgroup and THUCNews datasets are shown in Figs. 3,4,5, respectively.

Figs. 3,4,5 show that on the three standard datasets, our CSG-SM method outperforms all the baseline methods. Besides, the P, R and F1 score of GAC are slightly higher than that of KNN and the P, R and F1 score of LDA-GS are slightly higher than that of LDA-VEM. The small difference indicates that GAC and KNN have similar performance, as well as LDA-GS and LDA-



(a)
(b)
(c)

Fig. 6. Performance of KG-SM and CSG-SM on three datasets.

VEM. However, the P, R and F1 score of the two probabilistic approaches are significantly higher than that of the two term-based approaches on three datasets. Besides, the P, R and F1 score of KG+ are slightly higher than that of KG on three datasets, which means that there is no significant difference between two graph analytical approaches. The P, R and F1 score of the two graph analytic approaches are slightly higher than that of the two probabilistic approaches on three datasets, except that the P, R, F1 score of the LDA-GS are slightly higher than KG on 20newsgroup dataset. The P, R and F1 score of our CSG-SM are higher than that of the probabilistic approaches and graph analytical approaches, reaching to 82%, 77% and 79%, respectively on CEC dataset; 83%, 76% and 79%, respectively on 20newsgroup dataset; 80%, 75% and 77%, respectively on THUCNews dataset. Our CSG-SM method has obvious advantages over the graph analytical approaches and probabilistic approaches, and has significant advantages over the term-based approaches on three datasets.

In addition, we also calculated the value of $(C_{det})_{norm}$ for all the methods on three datasets. Results are shown in Table 2. Results in Table 2 show that our CSG-SM has the smallest $(C_{det})_{norm}$ value on three datasets. The results indicate that the CSG-SM has the best performance compared with the baseline methods.

The graph analytical approaches and the probabilistic approaches have better performance than the term-based approaches mainly because they both take into account the relationship between terms. The performance of LDA-GS is slightly better than LDA-VEM mainly because the inference process of Gibbs sampling performs better than that of VEM. The performance of KG+ is slightly better than KG, which indicates that double the weights of noun phrases and named entities can slightly improve the performance. Our CSG-SM has the best performance compared with the several baseline methods mainly attributed to several reasons: (1) The input documents are reorganized into graph structure and the relationship between words are connected by edges, which make the words no longer independent of each other. (2) The capsule vertices are coarse-grained vertices that contains a set of closely related words, thus each capsule vertex contains more semantic information. (3) The graph kernel is used to measure the similarity between CSGs by comparing the different semantic units of the capsule semantic graph, i.e., the capsule vertices comparison and edges comparison. Besides, results are similar under different size of the datasets, which indicates that our CSG-SM method has not only good performance but also strong robustness.

The above results also show that the performance of KG+ is very close to our proposed CSG-SM. This may be because KG+ and our proposed method are all graph-based method. However, different from KG+ that modeling all documents in the dataset into one graph by the word co-occurrence relationship in documents, we model each document into a keyword graph by the word co-occurrence in sentences, which can reduce the information loss and make our method achieve higher precision. Besides, different from KG+ that cluster the documents by assign documents to each community, we use the graph kernel to compare the similarity of the graphs and then use the single-pass algorithm for clustering, which can make our method more efficient for a large collection of documents. Running efficiency of different approaches is shown in Section 4.5.3.

4.5.2. Influence of community detection on topic detection

We evaluated the influence of community detection on the topic detection performance. Especially, we propose a KG-SM (Keyword Graph and Similarity Measurement) method, which measures the similarity of the keyword graph directly without community detection. We compare the performance of KG-SM with our CSG-SM on the three datasets and the results are shown in Fig. 6.

Fig. 6(a)–(c) show the P, R and F1 score on CEC, 20newsgroup and THUCNews dataset, respectively. The running time of KG-SM and CSG-SM on three datasets is shown in Table 3. Fig. 6(a)–(c) show that KG-SM performs slightly better than CSG-SM on three datasets and the difference is not significant. However, Table 3 shows that running time of the CSG-SM is significantly lower than the KG-SM and reduced by about 50% on three datasets, which indicates that by using community

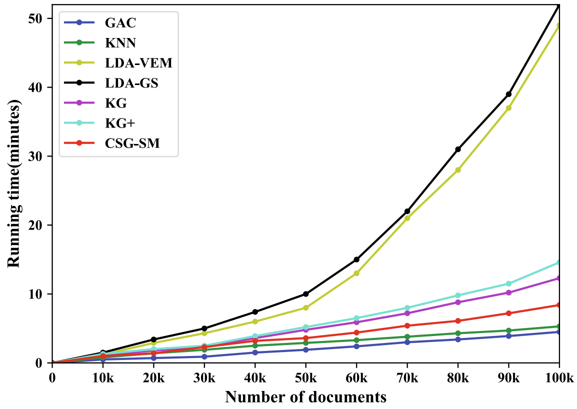


Fig. 7. Running time of the approaches under different dataset sizes.

detection, the CSG-SM can greatly improve the efficiency and reduce the running time for topic detection with only a small precision loss.

4.5.3. Evaluation of the running time

In this section, we use the THUCNews dataset to evaluate the efficiency of our CSG-SM. We designed the experiments to test the running time of the several approaches under datasets of different sizes (number of documents contained). The dataset size varies from 10 k to 100 k. The approaches are run on a PC with an Intel Xeon@2.1 GHz (two CPUs) processor and 32 GB memory, and the operating system is Windows10. Please note that it did not include the time consumption of the preprocessing and keyword extraction, because these steps are done by highly efficient off-the-shelf tools. The running time of the approaches is shown in Fig. 7.

Fig. 7 shows that the running time of LDA-GS and LDA-VEM increases with a nonlinear manner. As the number of documents exceeds 60 k, running time of the two probabilistic approaches is significantly higher than other approaches, which indicate that the time complexity of probabilistic approaches is the highest. It mainly because the inference algorithms used in the model are too complex. The running time of KG and KG+ is higher than that of GAC, KNN and CSG-SM under all dataset size and also increases with a nonlinear manner. It mainly because the graph construction in KG is based on the whole corpus and the time consumption of this process is very high. GAC and KNN have less running time compared with other methods and increase with a linear manner, mainly because the basic algorithm GAC and KNN used in topic detection are very simple and efficient. In contrast, the running time of CSG-SM is between the term-based approaches and graph analytical approaches. It mainly because the single-pass clustering in the last step is an efficient algorithm. In addition, the keyword graph construction process is based on every single document, which makes the document representation process more efficient. The results show that although the running time of our CSG-SM is slightly higher than GAC and KNN, it is still very efficient.

4.5.4. Evaluation on the parameters sensitivity

Figs. 8,9,10 show the results of parameter sensitivity analysis of our CSG-SM on CEC, 20newsgroup and THUCNews, respectively. The initial settings of three evaluated parameters including *edge_min*, *min_similarity* and *max_betweenness* are

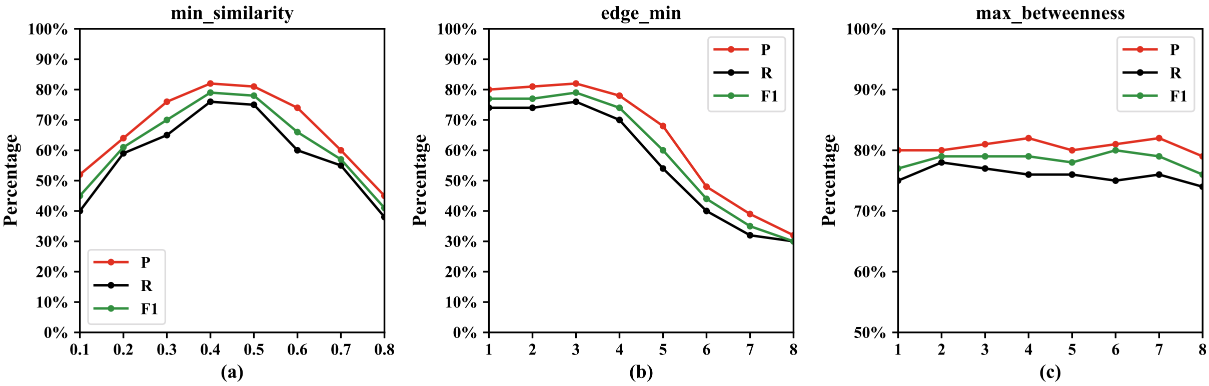


Fig. 8. Parameters sensitivity analysis of CSG-SM on CEC.

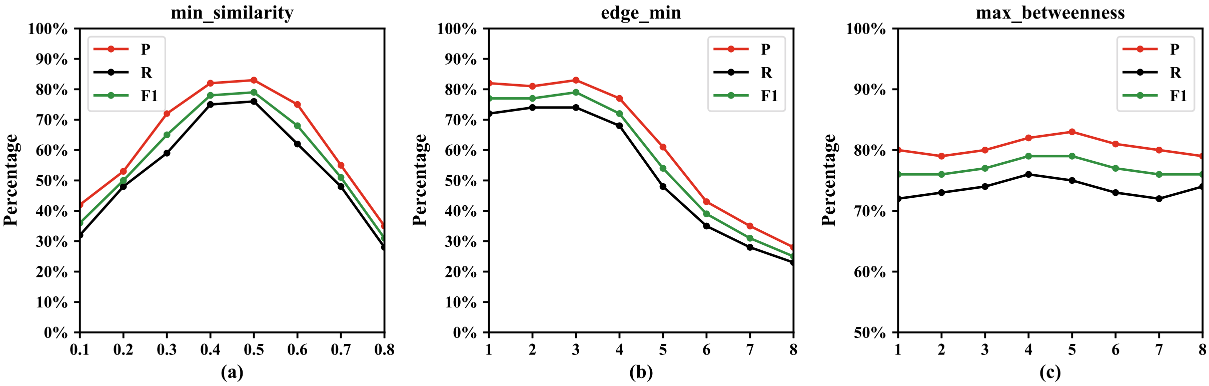


Fig. 9. Parameters sensitivity analysis of CSG-SM on 20newsgroup.

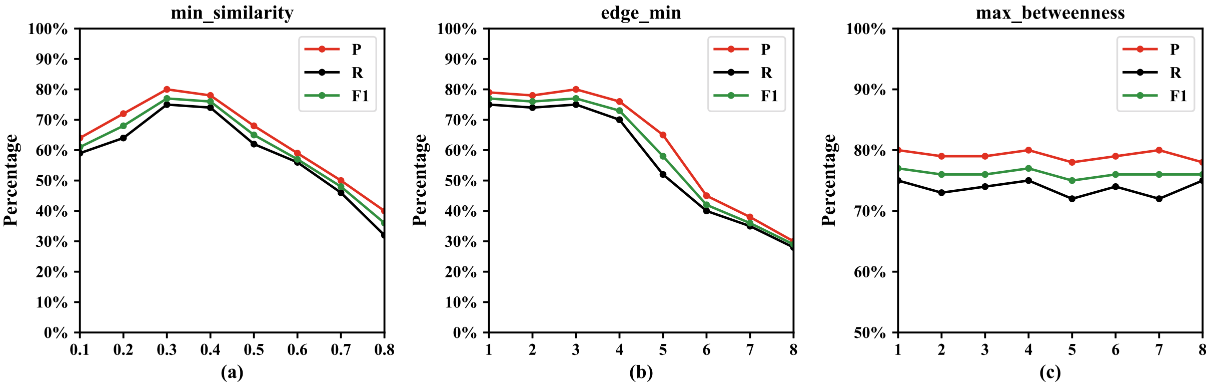


Fig. 10. Parameters sensitivity analysis of CSG-SM on THUCNews.

listed in Table 1. These initial values were set by our experience, which is determined by the preliminary experiments. The three parameters were adjusted manually and separately to optimize the results of CSG-SM. The process is as follows: (1) We first set the initial *max_betweenness* and *edge_min* value according to Table 1 and vary the parameter *min_similarity* from 0.1 to 0.8. The value that corresponds to the best system performance is the best *min_similarity* value. (2) Under the best *min_similarity* value obtained in (1) and the initial *max_betweenness* value in Table 1, the parameter *edge_min* was varied from 1 to 8. The value that corresponds to the best system performance is the best *edge_min* value. (3) Under the best *min_similarity* value obtained in (1) and best *edge_min* value obtained in (2), the parameter *max_betweenness* was varied from 1 to 8. The value that corresponds to the best system performance is the best *max_betweenness* value. According to the preliminary experiments, the parameter *max_betweenness* and *edge_min* have low sensitivity when they were set to a small values. Therefore, when testing the sensitivity of the model, the two parameters *max_betweenness* and *edge_min* were set firstly.

Figs. 8(a), 9(a) and 10(a) show the performance of CSG-SM with the variation of *min_similarity* on CEC, 20newsgroup and THUCNews datasets, respectively. It can be seen that as the *min_similarity* increases, the values of P, R and F1 score increase first and then decrease on three datasets. The best performance on CEC dataset can be obtained when *min_similarity* is 0.4. The best performance on 20newgroup dataset can be obtained when *min_similarity* is 0.5 and the best performance on THUCNews dataset can be obtained when *min_similarity* is 0.3.

Figs. 8(b), 9(b) and 10(b) show the performance of CSG-SM with the variation of *edge_min* on CEC, 20newsgroup and THUCNews datasets, respectively. The *min_similarity* on the three datasets are set to 0.4, 0.5, 0.3 respectively and the *max_betweenness* is still set to the initial value on three datasets. The results show that P, R and F1 score achieve the best performance when *edge_min* is 3 and slightly drop when *edge_min* is smaller than 3 on three datasets. However, the P, R and F1 score drop significantly when *edge_min* exceeds 3. The results indicate that there is a small probability that a pair of words co-occur in more than 3 sentences in a news document. Many edges of the keyword graph would be cut off if the value of *edge_min* is set too large, which may affect the result of keyword graph construction. Therefore, the value of *edge_min* should not be set too large.

Figs. 8(c), 9 (c) and 10(c) show the performance of CSG-SM with the variation of *max_betweenness* on CEC, 20newsgroup and THUCNews datasets, respectively. The *min_similarity* on three datasets are set to 0.4, 0.5, 0.3 respectively and the *edge_min* is set to 3. The results show that P, R and F1 score of our CSG-SM only varies slightly with the variation of *max_betweenness* on three datasets, which indicate that the CSG-SM is robust to the parameter of *max_betweenness*.

5. Conclusion and future work

In this paper, we propose a graphical decomposition and similarity measurement approach for topic detection from news called Capsule Semantic Graph and Similarity Measurement, which represents the document as a capsule semantic graph and then measures the similarity between the CSGs by graph kernel. Our CSG-SM effectively improved the performance of topic detection. The CSG not only can model the relationship between words and overcome the limitation of VSM that words are independent of one another, but also can retain the semantic information of documents. Moreover, the problem of similarity measurement between documents is transformed into the similarity measurement between CSGs. Then the CSGs of documents can be clustered to generate topics. In addition, the keyword graph is transformed into the CSG by community detection, which greatly reduces the computation of similarity measurement between graphs.

We evaluated the CSG-SM on CEC, 20newsgroup and THUCNews datasets. The experimental results show that the precision of CSG-SM is significantly better than the traditional term-based approaches, probabilistic approaches and graph analytical approaches. In addition, the experiment results on the large THUCNews dataset show that the time complexity of our CSG-SM is significantly lower than probabilistic approaches and graph analytical approaches. However, several problems of topic detection are still challenging, such as how to track the evolution of topics over time and how to detect fine-grained topics. In addition, how to combine news with multiple data forms, such as social media data and multimedia data for topic detection also remain to be challenges, and we will investigate these problems in future research.

CRediT authorship contribution statement

Kejing Xiao: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Zhaopeng Qian:** Software, Validation, Investigation, Data curation, Writing - review & editing, Visualization. **Biao Qin:** Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft, Funding acquisition, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61772534 and Beijing Natural Science Foundation under Grant No. 4194079.

References

[1] J. Allan, S.M. Harding, D. Fisher, A. Bolivar, S. Guzmanlara, P. Amstutz, Taking topic detection from evaluation to practice, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005.
[2] J. Allan, R. Papka, V. Lavrenko, On-line new event detection and tracking, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 51(2), 1998, pp. 37–45..
[3] L. AlSumait, D. Barbara, C. Domeniconi, On-line lda: adaptive topic models for mining text streams with applications to topic detection and tracking, in: Eighth IEEE International Conference on Data Mining, 2008, pp. 3–12.
[4] G. Altmann, M. Steedman, Interaction with context during human sentence processing, Cognition 30 (3) (1988) 191–238.
[5] D. Andrzejewski, X. Zhu, M. Craven, Incorporating domain knowledge into topic modeling via dirichlet forest priors, International Conference on Machine Learning 328 (26) (2009) 25–32.
[6] C.F. Baker, M. Ellsworth, Graph methods for multilingual framenets, in: Workshop on Graph based Methods for Natural Language Processing, 2017, pp. 45–50..
[7] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. (2003) 993–1022.
[8] S. Bleik, M. Mishra, J. Huan, M. Song, Text categorization of biomedical data sets using graph kernels and a controlled vocabulary, IEEE/ACM Trans. Comput. Biol. Bioinf. 10 (5) (2013) 1211–1217.
[9] K.M. Borgwardt, H. Kriegel, Shortest-path kernels on graphs, in: Fifth IEEE International Conference on Data Mining, 2005, pp. 74–81.
[10] T. Brants, F.R. Chen, A. Farahat, A system for new event detection, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 330–337.
[11] A. Castellanos, J. Cigarran, A. Garcia-Serrano, Formal concept analysis for topic detection: a clustering quality experimental analysis, Inf. Syst. 66 (2017) 24–42.
[12] Q. Chen, X. Guo, H. Bai, Semantic-based topic detection using markov decision processes, Neurocomputing 242 (2017) 40–50.
[13] A. Feragen, N. Kasenburg, J. Petersen, M.D. Bruijne, K.M. Borgwardt, Scalable kernels for graphs with continuous attributes, Neural Inf. Process. Syst. (2013) 216–224..
[14] J.G. Fiscus, G.R. Doddington, Topic detection and tracking evaluation overview, Topic Detect. Track. (2002) 17–31.
[15] L. Hermansson, T. Kerola, F. Johansson, V. Jethava, D. Dubhashi, Entity disambiguation in anonymized graphs using graph kernels, in: AMC Conference on Information and Knowledge Management, 2013, pp. 1037–1046.
[16] T. Hofmann, Probabilistic latent semantic analysis, Uncertain. Artif. Intell. (1999) 289–296..
[17] L. Hou, J. Li, Z. Wang, J. Tang, P. Zhang, R. Yang, Q. Zheng, Newsminer: multifaceted news analysis for event search, Knowl. Based Syst. 76 (2015) 17–29.
[18] B. Huang, Y. Yang, A. Mahmood, H. Wang, Microblog topic detection based on lda model and single-pass clustering, in: International Conference on Rough Sets & Current Trends in Computing, 2012, pp. 166–171.
[19] L. Hu, B. Zhang, L. Hou, J. Li, Adaptive online event detection in news streams, Knowl. Based Syst. 138 (2017) 105–112.
[20] Z. Li, B. Wang, M. Li, A probabilistic model for retrospective news event detection, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 106–113.
[21] H. Liu, B.W. Li, Hot topic detection research of internet public opinion based on affinity propagation clustering, Lect. Notes Electr. Eng. 107 (2012) 261–269.
[22] P. Mahe, J. Vert, Graph kernels based on tree patterns for molecules, Mach. Learn. 75 (1) (2009) 3–35.
[23] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. Mcclosky, The stanford corenlp natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.
[24] B.M. Masand, G. Linoff, D. Waltz, Classifying news stories using memory based reasoning, in: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1992, pp. 59–65.
[25] R. Mihalcea, P. Tarau, Textrank:bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411.
[26] T. Mikolov, I. Sutskever, K. Chen, C.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inf. Process. Syst. (2013) 3111–3119..
[27] M.E.J. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2) (2004) 321–330.
[28] G. Nikolentzos, P. Meladianos, F. Rousseau, Y. Stavrakas, M. Vazirgiannis, Shortest-path graph kernels for document similarity, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1890–1900.
[29] G. Nikolentzos, P. Meladianos, M. Vazirgiannis, Matching node embeddings for graph similarity, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2429–2435.
[30] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web, in: The Web Conference, 1999, pp. 161–172.
[31] J.W.G. Putra, T. Tokunaga, Evaluating text coherence based on semantic similarity graph, in: 2017 Workshop on Graph Based Methods in Natural Language Processing, Annual Meeting of Association for Computational Linguistics, 2017, pp. 76–85..

[32] J. Ramon, T. Gaertner, Expressivity versus efficiency of graph kernels, Proceedings of the First International Workshop on Mining Graphs Trees & Sequences 109 (4) (2003) 65–74.
[33] B. Rink, C.A. Bejan, S.M. Harabagiu, Learning textual graph patterns to detect causal event relations, in: Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, 2010, pp. 265–270.
[34] F. Rousseau, M. Vazirgiannis, Graph-of-word and tw-idf: new approach to ad hoc ir, in: Conference on Information and Knowledge Management, 2013, pp. 59–68.
[35] T. Sakaki, M. Okazaki, Y. Matsuo, Tweet analysis for real-time event detection and earthquake reporting system development, IEEE Trans. Knowl. Data Eng. 25 (4) (2012) 919–931.
[36] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manage. 24 (5) (1988) 323–328.
[37] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[38] K. Sasaki, T. Yoshikawa, T. Furuhashi, Online topic model for twitter considering dynamics of user interests and topic trends, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1977–1985.
[39] H. Sayyadi, L. Raschid, A graph analytical approach for topic detection, ACM Trans. Internet Technol. 132 (2013) 1–23.
[40] A. Schenker, M. Last, H. Bunke, Clustering of web documents using a graph model, Ser. Mach. Percept. Artif. Intell. 55 (2003) 3–18.
[41] B. Scholkopf, K. Tsuda, J. Vert, Kernel Methods in Computational Biology, MIT Press, 2004.
[42] N. Sherashidze, S.V.N. Vishwanathan, T. Petri, K. Mehlhorn, K.M. Borgwardt, Efficient graphlet kernels for large graph comparison, in: International Conference on Artificial Intelligence and Statistics, 2009, pp. 448–495.
[43] D. Spina, J. Gonzalo, E. Amigo, Learning similarity functions for topic detection in online reputation monitoring, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 527–536.
[44] M. Steyvers, T. Griffiths, Probabilistic topic models, 2007, pp. 424–440..
[45] S.V.N. Vishwanathan, N.N. Schraudolph, R. Kondor, K.M. Borgwardt, Graph kernels, J. Mach. Learn. Res. 11 (2) (2010) 1201–1242.
[46] C. Wei, Y. Lee, Y. Chiang, C. Chen, C.C. Yang, Exploiting temporal characteristics of features for effectively discovering event episodes from news corpora, J. Assoc. Inf. Sci. Technol. 65 (3) (2014) 621–634.
[47] Y. Yang, T. Pierce, J.G. Carbonell, A study of retrospective and on-line event detection, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 28–36.
[48] C. Zhang, H. Wang, L. Cao, A hybrid term–term relations analysis approach for topic detection, Knowl.-Based Syst. 93 (2016) 109–120.
[49] K. Zhang, J. Zi, L.G. Wu, New event detection based on indexing-tree and named entity, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 215–222.
[50] P. Zhou, Z. Cao, B. Wu, Edm-jbw: a novel event detection model based on js-id'f_order and bikmeans with word embedding for news streams, J. Comput. Sci. 28 (2018) 336–342.