Neurocomputing 242 (2017) 40–50

Contents lists available at ScienceDirect

Neurocomputing

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

Semantic-based topic detection using Markov decision processes

Qian Chen[a,b], Xin Guo[a,*], Hexiang Bai[a]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, PR China
[b] School of Electronics and Information Engineering, Tongji University, Shanghai 201804, PR China

ARTICLE INFO

Article history:
Received 24 July 2015
Revised 9 January 2017
Accepted 7 February 2017
Available online 21 February 2017

Communicated by Huaping Liu

Keywords:
Community discovery
Markov Decision Process
Topic detection
Topic graph
Topic pruning

ABSTRACT

In the field of text mining, topic modeling and detection are fundamental problems in public opinion monitoring, information retrieval, social media analysis, and other activities. Document clustering has been used for topic detection at the document level. Probabilistic topic models treat topics as a distribution over the term space, but this approach overlooks the semantic information hidden in the topic. Thus, representing topics without loss of semantic information as well as detecting the optimal topic is a challenging task. In this study, we built topics using a network called a topic graph, where the topics were represented as concept nodes and their semantic relationships using WordNet. Next, we extracted each topic from the topic graph to obtain a corpus by community discovery. In order to find the optimal topic to describe the related corpus, we defined a topic pruning process, which was used for topic detection. We then performed topic pruning using Markov decision processes, which transformed topic detection into a dynamic programming problem. Experimental results produced using a newsgroup corpus and a science literature corpus showed that our method obtained almost the same precision and recall as baseline models such as latent Dirichlet allocation and KeyGraph. In addition, our method performed better than the probabilistic topic model in terms of its explanatory power and the runtime was lower compared with all three baseline methods, while it can also be optimized to adapt the corpus better by using topic pruning.

1. Introduction

Due to the rapid development of computer networks and social media, the volumes of various types of data have been increasing rapidly, especially user-generated content. Therefore, there is an urgent need to discover interesting patterns hidden in these massive volumes of data. In this study, we focused on text data because texts are generated in natural human language and the semantic information hidden per unit size in text is richer than that in other data formats such as video, images, and audio. We aimed to discover latent hierarchical structures called topics in large-scale corpora by topic detection.

Topic detection was initiated in the topic detection and tracking (TDT) research program early in 1998, which aimed to discover topics or trends in various type of online media text data. TDT has attracted much attention in the last two decades in many application areas, such as online reputation monitoring [6], public opinion detection [7], and user interest modeling [18]. Topic detection is a fundamental application area in the text mining community, including text classification and clustering, information retrieval, and document summarization. [1]. Topic detection plays an important role in information retrieval and data mining, and it is an effective tool for organizing and managing text data such as newswire archives and research literature.

Unlike other existing applications in text mining and information retrieval, topic detection is an entirely unsupervised learning task without any topic classes or structure labels. In general, a topic is represented as related sets of keywords, and thus important descriptions can be given to topics or events. Many text clustering algorithms that typically compute similarities have been developed for topic detection, such as single pass incremental clustering algorithms [2] and incremental clustering algorithms [10]. Since the latent Dirichlet allocation (LDA) method was proposed by Blei in 2003 [4], the probabilistic topic model (pTM) has attracted much attention in the fields of information retrieval, text mining, and other areas. Essentially, pTM is a type of probabilistic model used for topic modeling, including LSA, pLSA, LDA, and various extension versions of pTM, which treat a topic as a distribution over the term space.

Despite the success of pTM, it has several drawbacks, as follows. (1) The inference algorithm used in the model can be too complex and much time is required to generate the topic word

* Corresponding author.
E-mail addresses: chenqian@sxu.edu.cn (Q. Chen), guoxinjsj@sxu.edu.cn (X. Guo), baihx@sxu.edu.cn (H. Bai).

http://dx.doi.org/10.1016/j.neucom.2017.02.020
0925-2312/

distribution, especially for large noisy and unbalanced corpora such as social media data. (2) There is a lack of explanatory power because the methods mentioned above ignore the semantic relationships between terms as well as topics. Existing explicit semantic topic detection methods usually build an ontology or some other structure containing rich semantic information, before employing ontology mapping, calculating, and reasoning to compute the similarity among terms to identify semantic relationships and facilitate semantic-based topic detection. However, building a general ontology requires a long time, despite the relatively low workload of building domain ontology. (3) Most importantly, to the best of our knowledge, topic optimization is not considered in the topic detection algorithm, which aims to optimize the topics generated and select appropriate topic words. Therefore, designing a new topic detection method that considers semantics and automatically selects the optimal topic set with low complexity in terms of time and space is a new challenge. In this study, we investigated the importance of topic semantic explicability and topic optimization, and we developed a topic graph establishment method, which represent topics using a network, where topics are represented as concept nodes and their semantic relationships using WordNet. In order to find the optimal topic that describes the related corpus, we define a topic pruning process and perform topic pruning using Markov decision processes (MDPs).

After completing our study, we recently found that Sayyadi and Raschid [11] proposed a graph analytical approach for topic detection by representing a topic as a graph based on keyword co-occurrence, as in our proposed method. However, there are two differences: (1) in our topic representation, we focus mainly on semantic information using an external knowledge-base; and (2)we propose a topic pruning process based on Markov decision processes, whereas Sayyadi and Raschid [11] did not consider topic optimization. Nevertheless, the conclusion of Sayyadi and Raschid [11] that word co-occurrence can obtain superior runtime performance compared with other solutions demonstrates that a similar approach can outperform pTM in terms of its lower time complexity.

In our proposed method, we first abstract the topics using a novel network called a topic graph, where the topics are represented as concept nodes and their semantic relationships using the WordNet database. Second, in order to find the optimal topic that describes the related corpus, we define a topic pruning processes, which is then used for topic detection. Third, we perform topic pruning using MDPs, which transforms topic detection into a dynamic programming problem. The main contributions of this study are summarized as follows.

(1) We propose a novel graphical representation for topics, which can identify related concept nodes as well as considering the relationships between concept nodes to detect deep semantic information hidden in the topics.
(2) We define a drill-down operator and we perform topic pruning using MDPs, thereby transforming topic detection into a dynamic programming problem, and thus the optimized topics can be adapted better to the corpus.
(3) We annotated the NIPS12 corpus, which include 1740 articles, and we also evaluated our approach using two different categories of corpus, i.e., newsgroup100 and NIPS12, in terms of the precision and recall, where the experiment results verified the efficiency of our approach.

The remainder of this paper is organized as follows. Related research is introduced in Section 2. We formulate the problem in a formal manner in Section 3 and Section 4 explains the topic graph construction process. We define topic optimization in Section 4 and the topic pruning algorithm is described in Section 5. Section 6 presents the details of our experiments and performance evaluations. Finally, we give our conclusions in Section 7.

2. Related work

In general, topic detection can be divided into two modes: online and off-line. Online topic detection aims to discover dynamic topics over time as new topics appear. Many studies have focused on new approaches to event detection, novel topic discovery, online topic evolution, and other problems in the online mode, which requires an incremental algorithm. Off-line topic detection is also known as retrospective topic/event detection, and it treats all documents in a corpus as a batch, before detecting topics one at a time [20]. In this study, we focused mainly on the off-line mode. Topic detection methods can be categorized according to three types: document clustering-based topic detection, pTM-based topic detection, and graph-based topic detection.

In document clustering-based topic detection, each document is represented as a vector using TF-IDF or improved TF-IDF, and each topic is simply a set of keywords. Brants proposed a variation of TF-IDF for detecting topics [19]. Many studies have considered retrospective topic detection using document clustering, including the well-known augmented group average clustering (GAC) method [20].

The LDA model is a Bayesian hierarchical probabilistic generative model, which was first proposed by Blei et al. [4]. In this method, each document is modeled as a discrete distribution over topics, and each topic is regarded as a discrete distribution over terms. LDA is used widely in text mining and other fields, and it is regarded as a powerful tool for topic modeling. The original LDA method used a variational expectation maximization (VEM) algorithm to infer topics for LDA [4], but stochastic sampling inference based on Gibbs sampling was proposed by Steyvers and Griffiths [12] for LDA. Similar to Sayyadi and Raschid [11], we denote LDA-GS as LDA with Gibbs sampling and LDA with VEM as LDA-VEM.

The two types of topic detection methods mentioned above only consider words, especially in the LDA model, where words are generated conditionally independent of a given distribution. In fact, there are richer relationships between words. Graph-based topic detection methods focus on between-words relationships. The co-occurrence patterns between words were considered in previous studies. For example, Petkos [3] treated the topic detection problem as a frequent pattern mining problem and proposed a soft frequent pattern mining algorithm. Cataldi also built a co-occurrence graph for tweets with an extra temporal dimension [21]. Sayyadi and Raschid proposed a graph analytical approach for topic detection called KeyGraph (KG) [11]. KG is essentially a keyword co-occurrence graph based on an off-the-shelf community detection algorithm for grouping co-occurring keywords into communities. Each community was a constellation of keywords representing a topic. Inspired by KG, we use betweenness-metric-based community detection for topic extraction in our proposed method.

Many studies have aimed to extend the LDA model. Some extended LDA models have been used to model authorship information [22], while others aim to capture the most recent language usage for sentiments and topics [23]. The biterm topic model is applied to short texts such as tweets based on an extension of the LDA [24]. The inverse regression topic model combines metadata with the LDA to utilize structural information in each document [25]. The correlated topic model is used to model the correlations between topics to remove the assumption of independence in the LDA [16]. Recently, deep learning techniques have been used to obtain low-dimensional representations of word and documents by word embedding. Thus, anovel neural topic model [15] was proposed to combine the advantages of topic models and neural networks, but it is essentially a supervised learning model. In pTM,

Table 1
Notations and their corresponding descriptions.

Token	Description
D	Corpus and $\|D\|$ is the number of documents in corpus D
V	Concept vocabulary and $\|V\|$ is the size of the concept vocabulary
G	Topic graph generated from D denoted as $G = (C, E)$
K	Number of topics or communities in a topic graph
C	Concept nodes set in a topic graph G
E	Edges set in a topic graph G
Q	Modularity of a topic graph G
$CC(c_i)$	Closeness centrality of a concept node c_i
R	Quantity representing instant reward in drill-down operation
T	Topic sub-graph, or a topic or community in a topic graph G
A	Action set in TG-MDPs
S	A state set in TG-MDPs, where s_t is a state variable in step t, which takes values in the set S
L	Number of steps iterated and l denotes the current step
λ	Smoothing parameter in the TF-IDF formulation
δ	Threshold parameter for removing weak edges
γ	Discount factor in [0,1)

many variations of LDA are employed, but we focused mainly on retrospective topic detection in this study. According to this review of previously proposed methods, we performed experiments to compare our method with three representative methods, i.e., KG, LDA-GS, and GAC, in terms of their time complexity, precision, recall, and F1-score.

3. Framework for semantic-based topic detection

In this section, we provide an overview of our semantic-based topic detection framework. We refer to our approach as topic graph-MDPs (TG-MDPs). First, we give formal definitions of the terms used in this study.

Definition 3.1. Topic graph: a topic graph is a graph structure containing nodes and edges between nodes denoted by $G = (C, E)$, where C represent a concepts set and the edges set E comprises the semantic relationships between concepts.

The topic graph has two differences compared with the KG proposed in a previous study [11]: (1) each node contains semantic information based on the external knowledge-base Wordnet; (2) each edge has a weight and relationship class information. If we consider a topic graph as a network, then many concepts related to a topic exist, thereby forming a community of high co-occurrence concepts. A topic can be viewed as a subgraph of a topic graph or a community of the topic graph network, which is similar to the concept of a community in social networks.

Definition 3.2. Topic pruning: Topic pruning is a process for optimizing topic selection in order to maximize the reduction of duplicated concepts based on the initial topic graph.

The symbols used in this study are shown in Table 1.

Currently, a topic model can output several keywords for each topic in a corpus, but the system does not know the meanings of the set of keywords Thus, there is a semantic gap between the keywords and events or topics, so the final decision about the actual topic represented by an extracted topic still depends on human intelligence. Therefore, it is very important to develop a method for inferring topics. In our proposed method, we use a novel network to describe topics as concepts as well as the relationships between pairs of concepts and a general otology in order to extend the topic's semantic information; thus, a topic model is a concept network that describes the corresponding events or topic-related keywords. The second feature of a topic comprises a hierarchical structure as well as the related concepts. After the topics have been extracted from the corpus, redundant information is hidden in each topic, and it is usually unclear whether a concept node is redundant or not. Thus, topic pruning is an important step in topic detection. The topics obtained from a corpus typically include redundant information because there are duplicate semantic relationships among words and concepts. Therefore, it is necessary to refine the topics in a process called topic pruning.

As demonstrated by our framework in Fig. 1, before building a document-term matrix based on a vector space model, a preprocessing step should be performed for stop-words removal, token segmentation, and stemming operations. There are three components in this framework, i.e., building a topic graph based on the vector space model, topic extraction based on community discovery, and topic pruning using MDPs. Topic graph generation is employed to generate a topic graph for a large corpus, which is then used by our topic detection algorithm. Topic extraction mainly aims to recover single topics using several methods. In our proposed method, we can either consider an unconnected graph as a topic or detect a community based on social network theory. Topic pruning is regarded as a tuning step for a topic sub-graph in order to optimize the final semantic structure of the topic. It should be noted that the topic concept extracted by our topic extraction method differs slightly from that obtained by previous graph-based methods because only the intercommunity edges were removed by [11], whereas we retain important between-community

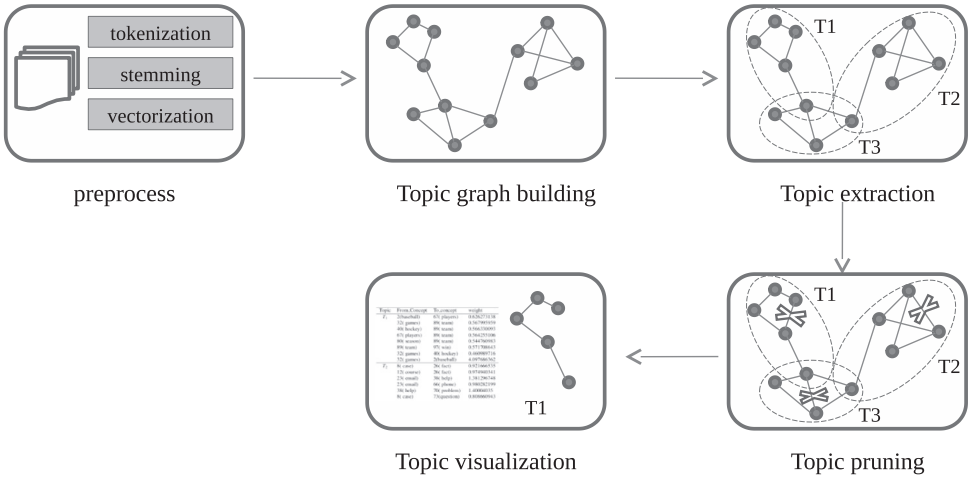

Fig. 1. Framework of the TG-MDPs approach.

information in our approach. This is because two different topics can share the same concept, e.g., *neuron* can be a topic word in both artificial neural networks and biology, which are completely different topic areas. In the next three sections, we describe the three core components in detail.

4. Building a topic graph

In order to embody the semantic characteristics of a topic, we consider the concept as well as the relationships between concepts by using an adjacent matrix over the term space, where the weight of the relationship between any two concept nodes denotes its strength. Furthermore, the relationship type can be assigned according to a universal semantic base, and WordNet [5] is used by our proposed method because of its versatility and generality.

The basis for building a topic network from a corpus is the vector space model. Given a real world corpus, we can build a term-document matrix, where each column is a vector representing a document. The size of each document vector is $|V|$ where the value of each element in that vector denotes a weight, and $|D|$ document vectors fill the whole vector space, which can be represented as

$$D = (\vec{d_1}, \vec{d_2}, ..., \vec{d_{|D|}}) = \begin{pmatrix} w_{11} & \cdots & w_{1|D|} \\ \vdots & \ddots & \vdots \\ w_{|V|1} & \cdots & w_{|V||D|} \end{pmatrix}, \quad (1)$$

where D is the vector space model of the corpus, $\vec{d_k}$ is the kth document vector, which has $|V|$ weight elements ($k \in 1, 2, \ldots, |D|$), $|V|$ is the size of the concept vocabulary, $|D|$ is the size of the corpus, and w_{ik} denotes the weight, which corresponds to the number of concept c_i appearing in document d_k. Thus, we have

$$w_{ik} = (1 + tf_{ik}) \cdot idf_i = (1 + tf_{ik}) \cdot \ln\left(\frac{\lambda + |D|}{\lambda + df_i}\right), \quad (2)$$

where tf_{ik} denotes the number of concept word c_i occurring in document d_k, idf_i denotes the inverse frequency of concept word t_i, and df_i is the number of documents containing concept word t_i. λ is a Laplace factor used for smoothing.

Next, the *term-document* matrix is decomposed and transformed into a *term-term* adjacent model using Eq. (3).

$$g_{ij} = \begin{cases} 0 & \text{if } i = j, \\ \sum_{k=1}^{|D|} \min(w_{ik}, w_{jk}) & \text{if } i \neq j. \end{cases} \quad (3)$$

Finally, we draw a line between the members of a term pair with a non-zero weight. We set a threshold parameter δ to remove weak edges if the corresponding weight is less than the threshold, and a large topic network called a topic graph is built for the corpus. Clearly, a denser network is obtained when the threshold is smaller.

Next, we use WordNet to add semantic information to the topic graph. WordNet is a large English dictionary created by Miller [5]. The words in WordNet are actually grouped by their semantics to form 117,000 synonym sets. Each synonym set can be treated as a semantic concept, and it also provides the semantic relationships between words, so semantically related words form a network. WordNet can be treated as a general ontology and it has been used widely in the fields of natural language processing, computational linguistics, text mining, artificial intelligence, and other related areas. For topic graphs, the information provided by WordNet is very important. All of the synonym sets in WordNet provide collections of concepts and all of the relationship types in WordNet provide the types of semantic relationships. Furthermore, nearly 80% of the relationship types reflect the hierarchy of the semantics. In contrast to Wikipedia and Freebase, the features of WordNet can help to reduce the semantic dimensions. In addition, WordNet provides APIs for developers, which make it easy to use.

A topic graph is a complex network where the node is a concept from a vocabulary and the relationship between a pair of concepts has a weight. All of the words can be found in WordNet and each relationship type can be labeled using WordNet. In fact, each node in a topic can be viewed as an ontology concept or instance, and each edge expresses the semantic relationships between concepts. Thus, a topic graph can be built into a novel network with both directed and undirected types of edges. For both the nodes of an edge, we consider the terms that the nodes represent and search for a semantic relationship with WordNet. If there is a relationship, we add the semantic relationship as the weight value. In order to reflect the semantic structure of a topic more concisely and hierarchically, we simply use the top-down relationship. For instance, we only use the hypernym but not the hyponym, and the holonym but not the meronym. If no relationship can be found in WordNet, we define the relationship as an undirected edge. In summary, a topic graph can be established based on the external knowledge-base WordNet to obtain a complex network that reflects the deep semantic of a topic.

5. Topic extraction based on community discovery

In the previous section, we described how to build a topic graph, which is essentially a social network of concepts. The topics embedded in the network can be extracted. In general, two methods can be used to address this problem. We can simply segment the subgraphs in the original network based on unconnected subgraph theory. Indeed, we extracted the topics using unconnected sub-graph from the NIPS literature corpus and the experimental results were not ideal because this method ignores the fact that edges between two communities may have greater values than edges within one community. Therefore, our approach employs a second method based on a community discovery algorithm from the social networks field. We note that finding communities within an arbitrary network can be a computationally demanding task.

A community is a subgraph containing nodes that are more densely linked to each other than the remainder of the graph [26]. Many approaches can be used for community detection but we employ the modularity maximization strategy and the well-known state-of-the-art Louvain method (LM), which is a widely used community detection method (as described in detail by [9]) because it is computationally feasible even with large networks. The LM method comprises two simple steps: (1) each node is assigned to a community that is selected in order to maximize the network modularity Q; (2) a new network is made comprising nodes found previously in these communities. The process iterates until a significant improvement in the network modularity is obtained.

Consider a topic graph $G = (C, E)$ partitioned into K topics, where n_k is the number of edges between concept nodes in the k-th topic and d_k is the sum of the degrees of the concept nodes in the k-th topic. The topic graph G has a modularity Q given by [9]

$$Q = \sum_{k=1}^{K} \left\{ \frac{n_k}{|E|} - \left(\frac{d_k}{2|E|} \right) \right\}. \quad (4)$$

The gain ΔQ derived by moving a concept node i into a community T can simply be calculated by computing the change in the modularity when node i is removed from its community (as described in more detail by [9]). Using this algorithm, we can recover a latent community with low time complexity, as shown in Section 7.4.

In order to assign topics to each document, we must compute the similarities between the topics and documents. We temporarily transform a topic community or subgraph into a vector, where each element represents a unique concept node's importance. Three methods can be used to evaluate the importance of

nodes [8]: (1) the degree centrality (DC) for the importance of a node depends on the number of adjacent nodes connected to it; (2) the closeness centrality (CC) emphasizes the importance of central nodes because these nodes diffuse information more rapidly than other nodes; and (3) the betweenness centrality (BC) considers the number of shortest paths that go through a specified node. Unlike Sayyadi and Raschid [11], we employ a CC metric-based approach because the DC metric is very simple and it ignores intercourse nodes, while the BC metric focuses on inter-community information, thereby leading to a more discriminative degree for the inter-community nodes of a topic compared with the real inner-community nodes. The CC is defined as follows:

$$CC(c_i) = \left[\frac{1}{D-1} \sum_{i \neq j}^{D} g(c_i, c_j) \right]^{-1}, \tag{5}$$

where $g(c_i, c_j)$ denotes the summed weight of edges on the shortest path from node i to node j. Thus, a topic community can be vectorized using CC as $T = v_1, v_2, ..., v_D$, where $v_k = CC(w_k)$. For each topic, we compute the similarities using the vector Euclidean distance between the topic and current document, and thus a topic with the maximal similarity value is assigned to that document. In the next section, we explain the topic pruning process.

6. Topic pruning using MDPs

The topic sub-graph or community might not be the optimal topic structure for the redundant information hidden in the topic because two words may express the same or a very similar meaning. Thus, a topic graph needs to be pruned in order to obtain the optimal state.

6.1. Drill-down operator

In this section, we define an important operator for topic pruning, i.e., a drill-down operator that can only be performed for two relationships, hypernyms and holonyms, because only these two semantic relationships are endowed with transitivity and monotonicity. We specify that only edges with hypernym or holonym semantic relationships are prunable. Drill-down is a binary operation on a given topic subgraph/community and a prunable edge. We give the following definition.

Definition 6.1. Drill-down: $T = T_o \oplus c$. Drill-down is an operation that takes a topic sub-graph and a node in the graph as inputs to output a new topic sub-graph with that node removed, where all the node's property information is absorbed into its parent node.

Details of the drill-down process are explained in Algorithm 1. In Algorithm 1, R is the variation quantity, which is the average change in the weight relationship after drill-down has been performed. If R is positive, this means that the total relationship weight is increasing; otherwise, the weight is decreasing, and we regard R as the instant reward in the drill-down operation. T is the new topic sub-graph obtained as an output.

To clarify the role of the drill-down operation, if we suppose that the head concept is the parent of the tail concept for a hypernym or holonym, then the drill-up operation involves merging the concepts that have no parents with this child and we change the edge type into non-hypernym and non-holonym. We can obtain a compact formulation using a merging operation conditioned on the hypernym or holonym relationship between related concept nodes in a topic. If we suppose that we have a topic graph $T_1 = (C_1, E_1)$, $c \in C_1$, then a new topic $T = (C, E)$ can be obtained by the pruning operation on T_1 and concept c.

Algorithm 1 Drill-down process.

Require:
topic T_1 ; concept node c
Ensure:
new topic T, reward value R;
Initialize $C \leftarrow C_1$, $E \leftarrow E_1$, $R \leftarrow 0$, $R_1 \leftarrow 0$, $R_2 \leftarrow 0$
if $\bar{e}.asso\_Type \in \{hy, ho\}$ && $\bar{e} \in E$ && $(\bar{e}.c\_1 = c$ or $\bar{e}.c\_2 = c)$ **then**
  $\bar{e}.c\_1 \Leftarrow e.c\_1$
  if $\exists \hat{e} \in E$ such that $\hat{e} = \bar{e}$ **then**
    if $\hat{e}.asso\_Type \neq \bar{e}.asso\_Type$ **then**
      if $\bar{e}.asso\_Type = r$ **then**
        $\bar{e}.asso\_Type \Leftarrow \hat{e}.asso\_Type$
      end if
      if $\bar{e}.asso\_Type \neq r$ && $\hat{e}.asso\_Type \neq r$ **then**
        if $\hat{e}.weight \geq \bar{e}.weight$ **then**
          $\bar{e}.asso\_Type \Leftarrow \hat{e}.asso\_Type$
        end if
      end if
    end if
    $\bar{e}.weight \Leftarrow \max(\bar{e}.weight, \hat{e}.weight)$
    $E \leftarrow E - \{\hat{e}\}$
  end if
  $E \leftarrow E - \{e\}$
end if
$C \leftarrow C - \{c\}$
$R_2 \Leftarrow (\sum_{\forall e'' \in E} e''.weight)/|E|$
$R \leftarrow R_2 - R_1, T = C, E$
return R, T

6.2. Optimal pruning using MDPs

MDPs is a dynamic programming algorithm for formal decision-making problems. MDPs comprises a series of system states and actions that control the states [13]. The solution of MDPs involves finding an optimal strategy that maximizes the performance evaluation to achieve the system's goal. MDPs has been used widely in programming, robot control learning, and game problems, and it has important roles in theory and applications in broad areas of economic management, computer science, control, and clinical decision making [14].

According to the Markov property, topic selection based on topic pruning can be formalized using MDPs. A topic sub-graph and a new topic sub-graph pruned from the original topic can both be viewed as a certain state. The selection on which the operation has been performed is regarded as a certain action and a strategy is a scheme that can be processed according to a certain state.

Definition 6.2 (State set S)**.**

$$S = \{s^1, s^2, \ldots, s^{|S|}\} \tag{6}$$

where $|S|$ is the number of states. s^{init} is the initial topic sub-graph state, and the topics s^1 can be obtained by a drill-down operation based on the initial topic and the concept node in the topic; thus, s^2 is a subsequent state or a new topic pruned from the former topic, and so on. It should be noted that the object pruned is a topic, topic sub-graph, or topic community, as described in the previous section. We also note that a certain topic state is in fact either the original topic sub-graph or a pruned new topic.

Definition 6.3 (Action set A)**.**

$$A = \{a^1, a^2, \ldots, a^{|A|}\}, \tag{7}$$

where $|A|$ is the number of actions, which is equal to the number of edges on which the drill-down operation can be performed.

Each action a^k corresponds to the drill-down operation on some edge.

Definition 6.4 (Transition function *Trans*).

$$Trans : S \times A \times S \rightarrow \{0, 1\} \tag{8}$$

The transition function *Trans* is a process that determines whether $s \in S$ can transform into some new state $s^* \in S$ after an action $a \in A$ has been performed on s, where the mapping value is either 1 or 0. We note that the sum of the transition values of all states transformed initially from state s is one, i.e., $\sum_{s^* \in S} Trans(s, a, s^*) = 1$. Therefore, the new state performed on state s by action a is fixed and unique according to the Markov property:

$$P(s_{l+1}|s_l, a_l, s_{l-1}, a_{l-1}, \ldots) = P(s_{l+1}|s_l, a_l) = Trans(s_l, a_l, s_{l+1}). \tag{9}$$

Definition 6.5 (Reward function *Rew*).

$$Rew : S \times A \rightarrow \mathbf{R} \tag{10}$$

where $\mathbf{R}$ is a real number set. The reward function gives the instant reward value after action $a \in A$ has been performed on state $s \in S$. When the real number is a larger positive number, the result is closer to the expected result, and vice versa. The relationship strength for a topic is expected to be larger after an operation has been performed on that topic previously. According to the definition of drill-down, the value of R obtained by topic pruning is suitable for the reward function $Rew(s, a)$.

Definition 6.6. TG-MDPs $=< S, A, Trans, Rew >$, is a quadruple comprising the state set, action set, transition function, and reward function, which perform the optimal processes for a topic graph based on the pruning operation.

Definition 6.7 (Strategy π).

$$\pi : S \rightarrow A \tag{11}$$

Given a TG-MDPs $< S, A, Trans, Rew >$, the strategy π indicates the action for each state.

Strategy solving is actually a topic reduction process based on the topic pruning operation, i.e., given the current topic, a new compact topic at time $l+1$ is generated according to the pruning process at time l until all the concept edges that can be pruned have been traversed. Given a strategy, we can obtain an action sequence as follows [14]

$$s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots s_l \xrightarrow{a_t} s_{l+1} \xrightarrow{a_{l+1}} \cdots s_{L-1} \xrightarrow{a_{L-1}} s_L, \tag{12}$$

based on which the action can be performed from the initial state to some state until convergence. Assuming that s_1 is the initial topic, s_{l+1} is a subsequent topic obtained by pruning based on some combinable edge in topic s_l. The optimization process does not end until a certain state takes no action and we obtain the state s_L, where s_L is the ultimate steady state and L is the number of iterations. Thus, the optimal strategy solution is the goal of topic pruning.

Definition 6.8 (Strategy Optimal Criterion (OC) [14]). OC is a criterion for selecting an action that maximizes the total reward, i.e., ensuring that the expectation of the sum of all the instant rewards $E[\sum_{l=1}^{\infty} \gamma^l Rew_l]$ is maximized, where $\gamma \in [0, 1)$ is the discount factor, which means that a later reward will be discounted more heavily. Thus, to maximize the expectation, a larger reward is needed as far ahead as possible.

Definition 6.9 (Value function, $V^{\pi}(s)$ [14]).

$$V^{\pi}(s) = E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k Rew_{l+k} | s_l = s\right] \tag{13}$$

The value function $V^{\pi}(s)$ is the expected reward under strategy π starting from state s. Thus, a strategy is evaluated based on OC.

Definition 6.10 (Bellman optimal equation [14]).

$$V^{\hat{\pi}}(s) = \max_{a \in A} \sum_{s^* \in S} Trans(s, a, s*)(Rew(s, a) + \gamma V^{\pi}(s^*)) \tag{14}$$

The Bellman optimal equation gives the expected reward from the best strategy $\hat{\pi}$.

$$\hat{\pi}(s) = \operatorname*{argmax}_{a \in A} \sum_{s^* \in S} Trans(s, a, s*)(Rew(s, a) + \gamma V^{\pi}(s^*)) \tag{15}$$

There are two main optimal strategy iterative solution approaches: strategy iteration and value iteration. Strategy iteration specifies a random strategy π and establishes an equation set with $|S| = N$ unknown variables and N equations according to the Bellman equation. Each unknown variable represents a value function $V^{\pi}(s)$ for one state s under the current strategy π; therefore, the value function of the current strategy under all states can be solved by linear programming.

Value iteration does not require the solution of an equation set. In this method, the Bellman optimal equation (14) is regarded as a value function update rule, as shown in Eq. (16). We set the value functions of all states $V_1^{\pi}(s^1), V_1^{\pi}(s^2), \ldots, V_1^{\pi}(s^N)$ to 0. Each value function can then be updated with an update rule and we obtain $V_2^{\pi}(s^1), V_2^{\pi}(s^2), \ldots, V_2^{\pi}(s^N)$. When the update process reaches convergence, we have $V_K^{\pi}(s^1), V_K^{\pi}(s^2), \ldots, V_K^{\pi}(s^N)$, i.e., the value function of each state no longer changes, where $k \in 1, 2, \ldots, K$ is the current number of iterations and K is the total number of iterations.

$$V_{k+1}^{\pi}(s) = \max_{a \in A} \sum_{s^* \in S} Trans(s, a, s*)(Rew(s, a) + \gamma V_k^{\pi}(s^*)) \tag{16}$$

Finally, the optimal strategy can be solved by substituting the final value function into Eq. (15). To simplify value iteration, there is no need to solve the optimal strategy for all states, so we employ this approach to solve topic pruning, where the process starts from the initial topic graph and the best action is found to move to the next state in a new topic graph, and vice versa, until the value function of the current state no longer increases. We refer to this process as *topic pruning*. Thus, the topic graph for the ultimate state can be generated by the topic pruning process.

In summary, topic pruning using MDPs can be described as follows.

(i) Define the MDP of a topic graph. $< S, A, Trans, Rew >$.
(ii) Solve to obtain the optimal strategy using value iteration.
(iii) According to the optimal strategy, topic pruning starts from the initial state s_1, and a state sequence and action sequence can be obtained $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots s_l \xrightarrow{a_l} s_{l+1} \xrightarrow{a_{l+1}} \cdots s_{L-1} \xrightarrow{a_{L-1}} s_L$. ...

7. Experiments and results

We evaluated TG-MDPs and several benchmark algorithms, i.e., KG, LDA-VEM, LDA-GS, and GAC. The experiments showed that TG-MDPs had similar precision to the benchmarks with the well-known newsgroup100 corpus and NIPS12 data set. The runtime for TG-MDPs was much lower than that for LDA-GS but slightly more than that using KG with NIPS12. The computer environment used for our experiments comprised Windows Server 2008 OS, an Intel Xeon CPU, and 128 GB memory.

7.1. Data sets

We performed experiments using two types of data sets, i.e., science technology literature and newsgroup data sets, in order to evaluate our method for retrospective topic detection.

20newsgroup. The 20newsgroup data set is a well-known text collection comprising approximately 20,000 newsgroup documents, which are partitioned (nearly) evenly across 20 different newsgroups. We downloaded a processed version of 20newsbydate[1]. which is easy to read into Octave as a sparse matrix. This collection comprised 18774 documents and 61188 words in the vocabulary. We performed stop-words removal using a long stopword list[2] and the feature dimensionality was still high. In order to make the algorithm run faster, we retained 2000 items from the vocabulary using standard idf filtering.

NIPS12. The NIPS12 corpus accessed from the homepage of Roweis[3] is an archive of complete texts comprising 1740 papers from the NIPS conference published from 1987 to 1999, and it was obtained using optical character recognition technology. The collection comprises 130 articles per year on average and it covers subjects such as brain imaging, control, learning theory, speech processing, and algorithms. In total, there are 1740 documents in the NIPS corpus with about 3172.34 words in each document. In order to speed up the implementation of the algorithm, we pre-processed the data set to reduce noise, including lower-case words, where we ignored non-alphabet characters and removed rare words that occurred less than 50 times in the corpus.

We needed to transform the pure text in the corpus NIPS12 into a term-document matrix, where the elements represented the number of times that terms occurred in a certain document.

7.2. Methods compared and the evaluation metric

We evaluated TG-MDPs based on comparisons with the following benchmark algorithms.

(1) *GAC*. GAC was proposed by Yang et al. [20] almost 20 years ago but in terms of empirical results, it is still one of the best algorithms for TDT task evaluation. To accelerate the process, GAC split the corpus into 400 bins in its initialization step and clustering was then performed within each bin. The tuning parameters in GAC were set to the same values described in a previous study [20] in retrospective mode.
(2) *LDA-GS*. This LDA algorithm implementation uses the collapsed Gibbs sampler described by Newman,[4], where the parameters were $\alpha = 0.05 \times N/(D \times T)$ and $\beta = 0.01$, with $K = 20$ and $K = 50$ for the newsgroup and NIPS12 data sets, respectively, because the newsgroup data set had 20 topics and NIPS covered almost 50 research sub-directions according to the CFP for the NIPS conference. As shown by[17] for several data sets containing between 1000 and 20,000 documents, LDA-GS converges in less than 500 iterations of sampling. We set the maximum iteration number parameter as $iter = 1000$ for both NIPS12 and 20newsgroup.
(3) *KG*. KG is based on the method proposed by Sayyadi and Raschid, but we only used the terms in each document as features to ensure a fair comparison. KG parameters such as node_min_df and edge_min_df had the same settings given in Table 1 in [11].

We did not use other LDA variations due to the reasons given in Section 2, and we did not make a comparison with LDA-VEM because LDA-GS outperforms LDA-VEM in terms of precision, recall, and the macro-average F1 score according to [11], especially with a formal corpus.

The results are expressed as the average topic precision, average topic recall, micro-average F1 score, and macro-average F1 score, which are denoted simply by p_{avg}, r_{avg}, *F1(micro-avg)*, and *F1(macro-avg)*, respectively. $F1(\textit{macro-avg}) = 2p_{avg} * r_{avg}/(p_{avg} + r_{avg})$. We obtained the *precision* and *recall* first, before taking the average to obtain the corresponding *F1(micro-avg)*, while the *F1(mac-avg)* is produced by determining the per-topic performance measures first and then averaging the corresponding measures.

Table 2
Performance of the four approaches with the 20newsgroup and NIPS12 data sets.

Method	p_{avg}	r_{avg}	$F1_{micro}$	$F1_{macro}$	Corpus
GAC	0.50	0.37	0.43	0.45	20newsgroup
LDA-GS	0.79	0.63	0.70	0.72	
KG	0.74	0.62	0.67	0.70	
TG-MDPs	0.75	0.62	0.68	0.71	
GAC	0.66	0.83	0.76	0.79	NIPS12
LDA-GS	0.89	0.82	0.85	0.88	
KG	0.85	0.80	0.82	0.86	
TG-MDPs	0.84	0.81	0.83	0.85	

Table 3
Topic results generated by TG-MDPs for the 20newsgroup data set.

Topic	From_Concept	To_concept	Weight
T_1	2 (baseball)	67 (players)	0.626273138
	32 (games)	89 (team)	0.567995959
	40 (hockey)	89 (team)	0.566330093
	67 (players)	89 (team)	0.564255106
	80 (season)	89 (team)	0.544760983
	89 (team)	97 (win)	0.571708643
	32 (games)	40 (hockey)	0.460989716
	32 (games)	2 (baseball)	4.097686362
T_2	8 (case)	26 (fact)	0.921666535
	12 (course)	26 (fact)	0.974940341
	23 (email)	38 (help)	1.381296748
	23 (email)	66 (phone)	0.980282199
	38 (help)	70 (problem)	1.40004035
	8 (case)	73 (question)	0.808660943
	12 (course)	73 (question)	0.916499888
	26 (fact)	73 (question)	0.912370452
	38 (help)	73 (question)	0.893294911
	70 (problem)	73 (question)	0.967571099
	11 (computer)	78 (science)	0.916058685
	38 (help)	88 (system)	0.838474
	70 (problem)	88 (system)	1.028261524
	11 (computer)	91 (university)	0.89448725
	23 (email)	91 (university)	1.080396027
	86 (state)	91 (university)	1.085699047
T_3	26 (god)	100 (world)	0.766409743
	10 (christian)	46 (jesus)	0.838817506
	33 (god)	46 (jesus)	1.207593033
	3 (bible)	33 (god)	1.019431822
	10 (christian)	33 (god)	1.167747896
T_4	7 (card)	93 (video)	1.000021589
T_5	19 (dos)	98 (windows)	1.158041473
	71 (program)	98 (windows)	0.841958527

7.3. Experimental results

We used the APIs provided by WordNet 2.1 to determine the semantic relationships between nodes in the topics and to build the topic graph. The results are shown in Table 2.

The topic results generated by TG-MDPs for the 20newsgroup data set are shown in Table 3.

When we selected topic T_1, there were three prunable edges called e_1, e_2, e_3, with three actions called a_1, a_2, a_3, respectively, whereas another action a_4 was called a motionless action, which means that no action was taken. Thus, four actions could be performed for each state and the overall state transition chart is shown in Fig. 3.

[1] http://www.qwone.com/~jason/20Newsgroups/20news-bydate-matlab.tgz.
[2] Downloaded from http://www.ranks.nl/stopwords.
[3] http://www.cs.nyu.edu/~roweis/data/nips12raw_str602.tgz.
[4] Code can be downloaded from http://www.ics.uci.edu/~newman/code.

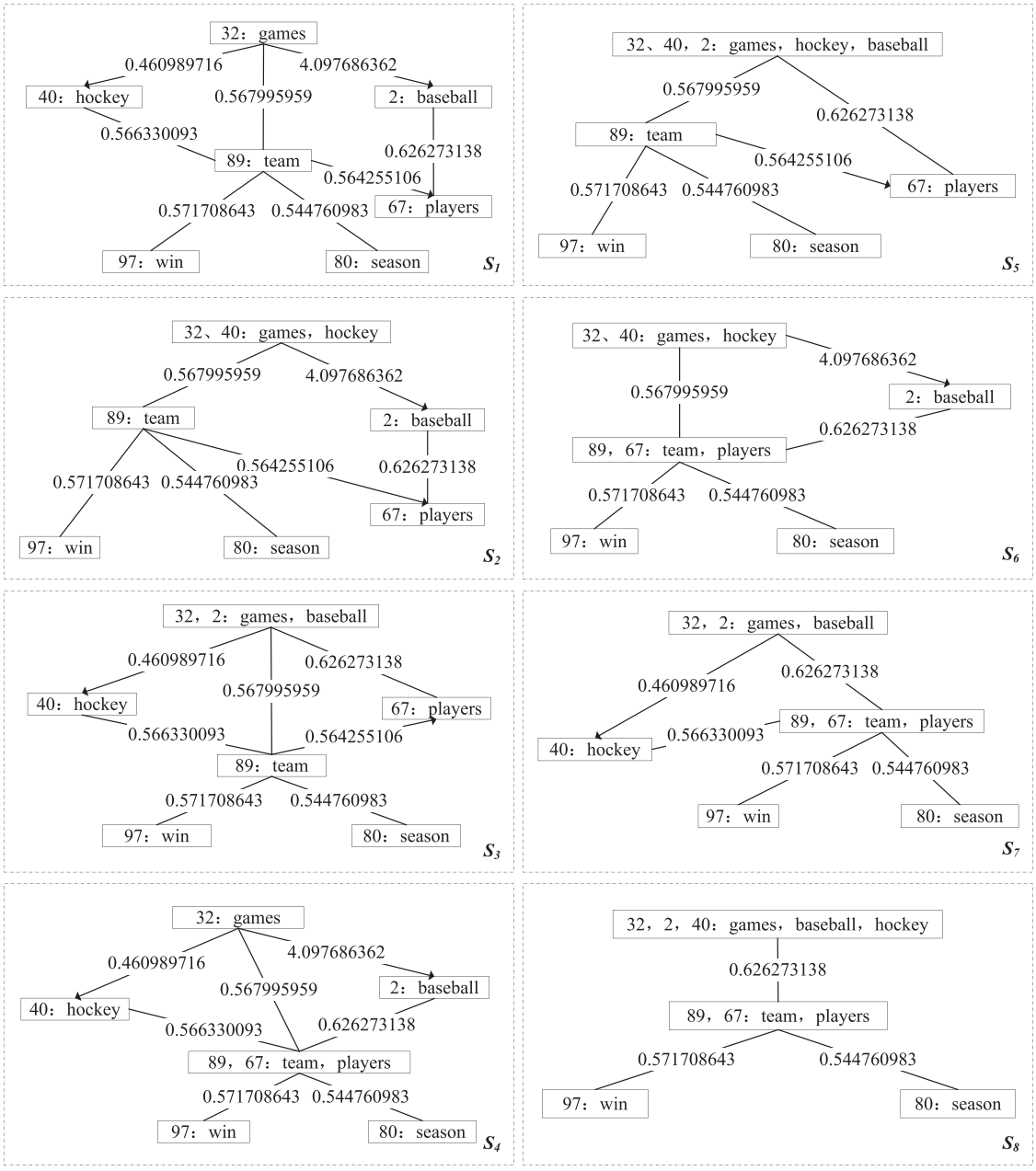

Fig. 2. Chart showing all the states for topic T_1.

When the number of prunable edges in the topic graph T_0 was N, there were $\sum_{i=1}^{N}(C_N^i)+1$ states in total starting from the topic graph, each of which corresponded to a certain topic graph. Thus, eight states started from topic graph T_2 and all of these states are shown in Fig. 2. There were two types of edges and three directed edges, i.e., *89.team* to *67.players* was a holonym relationship, and *game* to *baseball* or *hockey* were both hypernym relationships.

We calculated the value of R in terms of each state s and each action taken by s using Algorithm 1, and we present the *Rew* functions for topic graph T_1 in Table 4.

Note that the R value for actions that were not defined for a certain state were set to -100 in order to give a higher penalty and guarantee that no action would be taken. A self-transition action was set to 0 because the state did not change when action $a4$ was performed during the iterative process.

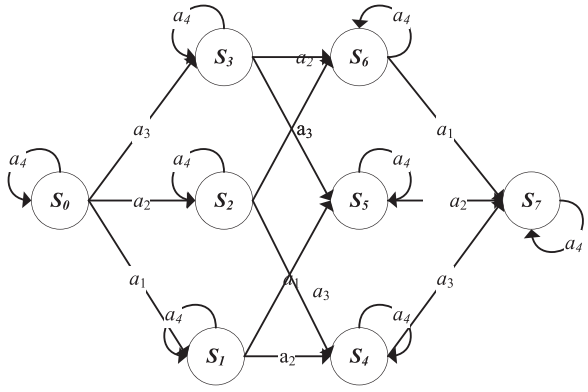

Fig. 3. State transitions generated for topic T_1 as described in the experimental section.

Table 4
Reward function table for certain states and actions.

State	a_1	a_2	a_3	a_4
S_1	0.16211	−0.44253	0.06224	0
S_2	−100	−0.58711	0.11957	0
S_3	0.01752	−100	−0.00346	0
S_4	0.21943	−0.50824	−100	0
S_5	−100	−100	0.00591	0
S_6	−10000	−0.70077	−100	0
S_7	0.0269	−100	−100	0
S_8	−100	−100	−100	0

7.4. Time complexity analysis

As described in the previous section, compared with LDA-GS and LDA-VEM, our approach performed better in terms of the time cost. However, due to the pruning process, our approach required slightly more time compared with KG. Nevertheless, the GAC algorithm was the fastest because of the simplicity and high efficiency of hierarchical clustering, but at the cost of losing semantic information. We analyzed the complexity of TG-MDPs compared with KG, LDA-GS, and GAC in order to theoretically verify the time and space complexities of our TG-MDPs approach.

In Table 1, $|D|$ is the number of documents and $|V|$ is the number of unique words in the corpus. Assuming that there are M words in a document, the number of edges in a topic graph is N. For the 20newsgroup data set, we can see that $|D| \gg |V|$, whereas in NIPS $|D| \ll |V|$, K is the number of topics, which is much smaller than both $|D|$ and $|V|$. The complexity of TG-MDPs is explained as follows. The production of the document-terms matrix requires $O(|D| \cdot M)$, the topic graph is created in $O(|V|^2)$, semantic concept and edge assignment requires $|V| + N$, and thus the first component runs in $O(|D| \cdot M + |V|^2 + |V| + N) = O(|D| \cdot M + |V|^2)$. During topic extraction, the complexity depends on LM and we assume that a "pass" is a combination of the two phases described in Section 5. The number of sub-communities decreases in each pass, and thus most of the computational time is consumed in the first pass. Passes are iterated until no more changes occur and the maximum modularity is attained. Communities of communities are built during this process and the height of the hierarchy constructed is determined by the number of passes, denoted by L, which is generally a small number. Thus, the complexity of topic extraction is not related to the size of the network and the time complexity is extremely low. The third component is MDPs, and we see can that $|S|$ is the number of states, $|A|$ is the number of actions, and L is the number of iterations, where L is usually a low number less than 50 for topics with 1000 concept nodes when using the value iteration strategy. Typically, $|A| < |S| \ll |V|$, which is much less than the number of concept nodes. Thus, topic pruning is achieved in $O(|A| \cdot |S| \cdot |L|)$. In summary, the total runtime for our approach can be completed in $O(|D| \cdot M + |V|^2 + |L|^2 + |A| \cdot |S| \cdot |L|)$, which is relatively much smaller than those of $O(|D| \cdot |V| \cdot KI)$ for LDA-VEM and $O(|D|MKI)$ for LDA-GS, where I is the number of iterations in each version of LDA.

For the GAC algorithm, assuming that the number of bins is B, then in each iteration, it divides the current set of active

Table 5
Iteration values for the value function in TG-MDPs.

$\gamma = 0$	Iter=1	2	3	π	$\gamma = 0.4$	Iter=1	2	x3	π
s1	0	0	0	1	s1	0.2099	−0.4346	0.15	1
s2	−1	−0.0001	0	3	s2	−100	−0.5847	0.1196	3
s3	0	−1	0	1	s3	0.0199	−100	0.0073	1
s4	0	−0.0001	−1	1	s4	0.2194	−0.4975	−100	1
s5	−1	−1	0	3	s5	−100	−100	0.0059	3
s6	−1	−0.0001	−1	4	s6	−100	−0.7008	−100	4
s7	0	−1	−1	1	s7	0.0269	−100	−100	1
s8	−1	−1	−1	4	s8	−100	−100	−100	4
$\gamma = 0.5$	iter=1	2	3	π	$\gamma = 0.8$	itepr=1	2	3	π
s1	0.2219	−0.4323	0.172	1	s1	0.2578	−0.4247	0.2378	1
s2	−100	−0.5842	0.1196	3	s2	−100	−0.5824	0.1196	3
s3	0.0205	−100	0.01	1	s3	0.0223	−100	0.0181	1
s4	0.2194	−0.4948	−100	1	s4	0.2194	−0.4867	−100	1
s5	−100	−100	0.0059	3	s5	−100	−100	0.0059	3
s6	−100	−0.7008	−100	4	s6	−100	−0.7008	−100	4
s7	0.0269	−100	−100	1	s7	0.0269	−100	−100	1
s8	−100	−100	−100	4	s8	−100	−100	−100	4
$\gamma = 0.6$	iter=1	2	3	π	$\gamma = 0.9$	iter=1	2	3	π
s1	0.2339	−0.4299	0.1939	1	s1	0.2697	−0.422	0.2597	1
s2	−100	−0.5836	0.1196	3	s2	−100	−0.5818	0.1196	3
s3	0.0211	−100	0.0127	1	s3	0.0228	−100	0.0208	1
s4	0.2194	−0.4921	−100	1	s4	0.2194	−0.484	−100	1
s5	−100	−100	0.0059	3	s5	−100	−100	0.0059	3
s6	−100	−0.7008	−100	4	s6	−100	−0.7008	−100	4
s7	0.0269	−100	−100	1	s7	0.0269	−100	−100	1
s8	−100	−100	−100	4	s8	−100	−100	−100	4
$\gamma = 0.7$	iter=1	2	3	π	$\gamma = 1.0$	iter=1	2	3	π
s1	0.2458	−0.4274	0.2159	1	s1	0.2817	−0.4191	0.2817	3
s2	−100	−0.583	0.1196	3	s2	−100	−0.5812	0.1196	3
s3	0.0217	−100	0.0154	1	s3	0.0234	−100	0.0234	3
s4	0.2194	−0.4894	−100	1	s4	0.2194	−0.4813	−100	2
s5	−100	−100	0.0059	3	s5	−100	−100	0.0059	3
s6	−100	−0.7008	−100	4	s6	−100	−0.7008	−100	4
s7	0.0269	−100	−100	1	s7	0.0269	−100	−100	1
s8	−100	−100	−100	4	s8	−100	−100	−100	4

Table 6
Detailed topic label information for the 20newsgroup data set.

T1	T2	T3	T4
alt.atheism	comp.sys.ibm.pc.hardware	rec.autos	sci.crypt
talk.politics.guns	comp.graphics	rec.motorcycles	sci.electronics
talk.politics.mideast	comp.os.ms-windows.misc	rec.sport.baseball	sci.spaces
talk.politics.misc	comp.sys.mac.hardware	rec.sport.hockey	sci.med
talk.religion.misc	comp.windows.x		
soc.religion.christian			misc.forsale

Table 7
Topic results generated by LDA using the 20newsgroup data set.

Topic	Top 10 concept words
T_1	god Jesus Bible does Christian people question believe sin lord
T_2	car BMW health drive question power engine email course university
T_3	problem help Windows edu problem university case fact medicine doctor
T_4	team games win players league human baseball season hockey car
T_5	email problem software hard system PC university computer help program
T_6	space NASA shuttle data citizens Moon Earth system orbit secure
T_7	God Jews religion Christian fact Jesus faith question life world
T_8	Windows program DOS card software help files system problem email
T_9	key government use clipper law enforcement public fact course phone
T_{10}	NSA Clinton security new enforcement encryption people archive technology board
T_{11}	cancers UIUC help disease red right health food used science
T_{12}	mideast gun believe religion crisis contradictions holy east road African
T_{13}	image black graphic Holloway paper ink beam pink enlightening view
T_{14}	motorcycles car new engine Louis motor assembly make company traffic
T_{15}	tax year income federal bills pay amount service million economic
T_{16}	police arrest officer charge law enforcement drug cocaine authorities last
T_{17}	weapon nuclear military base strategy baker missile help arms soviet
T_{18}	CBS network television show time homes series coverage news week
T_{19}	waste garbage company park town dump year trash Disney recycling
T_{20}	space Mars rocket satellite telescope Earth mission shuttle launch flight

clusters/documents into bins and performs local clustering within each bin. The process is repeated to generate clusters at increasingly higher levels, until a pre-determined number of top-level clusters are obtained. Thus, GAC typically has a complexity of $O(|D| \cdot B)$.

7.5. Analysis of robustness for the model parameters

We investigated the sensitivity of TG-MDPs with respect to δ and γ, where we ran TG-MDPs using the 20newsgroup data sets. In Table 1, a smoothing parameter is shown and according to previous studies, we set $\lambda = 0.5$ so the TF-IDF value could reach a good result.

For the discount factor γ, we checked whether the final state was sensitive to γ. We performed topic pruning using TG-MDPs and the iteration process is shown in Table 5 with different values of γ from 0 to 1. We can see that the ultimate stable state sequence was not variable in terms of the value of the discount factor γ. Thus, the long-term expected reward with respect to a fixed action was the same as the short-term reward, so our results were not sensitive to γ. This was the case for our topic graph with eight states and three pruning edges. We used the ultimate stable strategy $\pi = [1, 3, 1, 1, 3, 4, 1, 4]$ to determine the best pruning path $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_3} s_6$, except when $\gamma = 1.0$, where the stable optimal strategy was $\pi = [3, 3, 3, 2, 3, 4, 1, 4]$, and thus the best pruning path was $s_1 \xrightarrow{a_3} s_4 \xrightarrow{a_2} s_7 \xrightarrow{a_1} s_8$.

The threshold parameter δ controls the final topic number, so the F1-score is inevitably sensitive to δ. In the initial experiment, we fixed δ according to the empirical results. We set $\delta = 0.85$ for the 20newsgroup data set and $\delta = 0.62$ for NIPS12 to ensure that the topic number corresponded to the ground truth information.

7.6. Topic visualization

In order to evaluate the semantic topic detection approach, we show the topic information results generated by LDA-GS [4] only for the 20newsgroup data set using the detailed labeled topic information in Table 6 with four topics and 20 sub-topics. The top 10 words for each of the 20 sub-topics are shown in Table 7.

The experimental results show that most of the topics are covered in Table 6. The types of redundant information between T_1 and T_7, and T_6 and T_{20} were related to each other. However, LDA could not tell the difference or perform a merge operation to remove redundant information. Furthermore, our approach was easier to adapt to a corpus and it automatically selected the topic number, whereas the topic number must be manually specified in LDA.

8. Conclusions

In this study, we presented a novel graph-based topic detection approach called TG-MDPs. Our goal was to design a new topic detection method that considers semantic information as well as automatically selecting the optimal topic set with low time complexity. TG-MDPs is essentially based on word co-occurrence and it captures semantic information based on an external knowledgebase. TG-MDPs comprises three steps: topic graph building, topic extraction, and topic pruning. First, topic graph building represents topics as concept nodes and their semantic relationships using WordNet. In topic extraction, we employ a Louvain modularity-based community discovery algorithm to extract topic communities from a corpus and we assign each topic using the CC metric. In order to identify the optimal topic that describes a related corpus, we defined a topic pruning process, which is used for topic detection. Finally, we perform topic pruning using MDPs, which transforms topic detection into a dynamic programming problem.

The experimental results obtained using a newsgroup corpus and science literature corpus showed that our method had almost the same precision and recall as baseline models such as LDA and KG. In addition, our method performed better than pTM in terms of its explanatory power and the runtime was lower compared with the three baseline methods. In contrast to KG, our approach can be optimized to adapt to a corpus better by using topic pruning.

As discussed in Section 2, several problems are still challenging, such as topic detection in a dynamic text stream and online event detection. Our proposed approach is not suitable for these tasks, which require incremental algorithms, and we will investigate this issue in future research. We will combine metadata, including author and time information, in unstructured text data to improve the performance of our algorithm. Novel topic detection involves one-class classification without any previous knowledge, especially in social media. Much research has focused on social media because of the massive volumes of data generated, high dynamics in terms of temporal dimensionality, frequent interactions, and great variation in sample size. In our future research, we will mainly focus on topic detection and evaluation in social media.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under Grant nos. 61403238, 61502288, 61502287, and 61673248, Natural Science Foundation of Shanxi under Grant no. 2014021022-1, National High-Tech Research and Development Plan of China under Grant no. 2015AA015407, and Program of Shanghai Science Research Project by Science and Technology Commission of Shanghai Municipality under Grant no. 16JC1403000.

References

[1] M.W. Berry, J. Kogan (Eds.), Text Mining: Applications and Theory, John Wiley & Sons, 2010.
[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, Melbourne, Australia, 1998, pp. 137–145.
[3] G. Petkos, S. Papadopoulos, L. Aiello, et al., A soft frequent pattern mining approach for textual topic detection, in: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), Thessaloniki, Greece, 2014, pp. 25:1–25:10.
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[5] G.A. Miller, R. Beckwith, C.D. Fellbaum, et al., Wordnet: an online lexical database, Int. J. Lexicogr. 3 (4) (1990) 235–244.
[6] S. Damiano, G. Julio, A. Enrique, 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, Queensland, Australia, 2014, pp. 527–536.
[7] K. Lerman, A. Gilder, M. Dredze, F. Pereira, Reading the markets: forecasting public opinion of political candidates by news analysis, in: Proceedings of the 22nd International Conference on Computational Linguistics, volume 1, 2008, pp. 473–480.
[8] L. Tang, H. Liu, Community Detection and Mining in Social Media, Synthesis Lectures on Data Mining and Knowledge Discovery vol. 2 (1) (2010) 1–137.
[9] V. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. 10 (2008) P10008.
[10] A. Ahmed, Q. Ho, C.H. Teo, et al., Online inference for the infinite topic-cluster model: storylines from streaming text, in: Proceedings of International Conference on Artificial Intelligence and Statistics, 2011, pp. 101–109.
[11] H. Sayyadi, L. Raschid, A graph analytical approach for topic detection, ACM Trans. Internet Technol. 13 (2) (2013) 4:1–4:23.
[12] M. Steyvers, T. Griffiths, Probabilistic topic models, Handbook of Latent Semantic Analysis vol. 427 (7) (2007) 424–440.
[13] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons (2009) 414.
[14] C.C. Bennett, K. Hauser, Artificial intelligence framework for simulating clinical decision-making: a Markov decision process approach, Artif. Intell. Med. 57 (1) (2013) 9–19.
[15] Z. Cao, S. Li, Y. Liu, W. Li, H. Ji, A novel neural topic model and its supervised extension, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015, 2015, pp. 2210–2216.
[16] D.M. Blei, J.D. Lafferty, A correlated topic model of science, Ann. Appl. Stat. 1 (1) (2007) 17–35.
[17] A. Asuncion, M. Welling, P. Smyth, Y.W. Teh, On smoothing and inference for topic models, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009.
[18] C. Wang, M. Zhang, L. Ru, et al., Automatic online news topic ranking using media focus and user attention based on aging theory, in: ACM Conference on Information and Knowledge Management,CIKM 2008, Napa Valley, California, 2008, pp. 1033–1042.
[19] T. Brants, F. Chen, A. Farahat, A system for new event detection, in: Proceedings of 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 330–337.
[20] Y. Yang, T. Pierce, J.G. Carbonell, A study on retrospective and online event detection, in: Proceedings of 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 28–36.
[21] M. Cataldi, L.D. Caro, C. Schifanella, Emerging topic detection on twitter based on temporal and social terms evaluation, in: Proceedings of the 10th International Workshop on Multimedia Data Mining, MDMKDD, 2010, pp. 4:1–4:10.
[22] M. Rosen-Zvi, T. Griffiths, M. Steyvers, P. Smyth, The author-topic model for authors and documents, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, UAI '04, Arlington, Virginia, United States, 2004, pp. 487–494.
[23] Y. He, C. Lin, W. Gao, K.-F. Wong, Dynamic joint sentiment-topic model, ACM Trans. Intell. Syst. Technol. 5 (1) (2013) 6:1–6:21.
[24] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, New York, 2013, pp. 1445–1456.
[25] M. Rabinovich, D.M. Blei, The inverse regression topic model, in: Proceedings of the 31st International Conference on Machine Learning, ICML'14,, Beijing, China, 2014, pp. 199–207.
[26] M.E.J. Newman, Detecting community structure in networks, Eur. Phys. J.: B Condens. Matter Complex Syst. 38 (2004) 321–330.

Qian Chen was born in Huanggang, China in November 1983. He received his BSc in computer science and technology from Donghua University, China, in 2009. He received his PhD in computer science from Tongji University in 2012. He is currently working at the School of Computer and Information Technology, Shanxi University and has been a lecturer since 2013. His major interests are text mining, machine learning, and TDT.

Xin Guo was born in Taiyuan, Shanxi, China in December 1982. She received her BSc in computer science and technology from Donghua University, China, in 2009. She received her PhD in computer science from Tongji University in 2014. She is currently working at the School of Computer and Information Technology, Shanxi University and has been a lecturer since 2014. Her major interests are text mining, feature learning and dimensionality reduction.

Hexiang Bai is with the School of Computer and Information Technology, Shanxi University. His major interests are rough sets and machine learning.