Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05662-4

S.I. : 2019 INDIA INTL. CONGRESS ON COMPUTATIONAL INTELLIGENCE

Nonparametric method of topic identification using granularity concept and graph-based modeling

Isha Ganguli[1] · Jaya Sil[1] · Nandita Sengupta[2]

Received: 16 June 2020 / Accepted: 27 December 2020

Abstract
This paper aims to classify the large unstructured documents into different topics without involving huge computational resources and a priori knowledge. The concept of granularity is employed here to extract contextual information from the documents by generating granules of words (GoWs), hierarchically. The proposed granularity-based word grouping (GBWG) algorithm in a computationally efficient way group the words at different layers by using co-occurrence measure between the words of different granules. The GBWG algorithm terminates when no new GoW is generated at any layer of the hierarchical structure. Thus multiple GoWs are obtained, each of which contains contextually related words, representing different topics. However, the GoWs may contain common words and creating ambiguity in topic identification. Louvain graph clustering algorithm has been employed to automatically identify the topics, containing unique words by using mutual information as an association measure between the words (nodes) of each GoW. A test document is classified into a particular topic based on the probability of its unique words belong to different topics. The performance of the proposed method has been compared with other unsupervised, semi-supervised, and supervised topic modeling algorithms. Experimentally, it has been shown that the proposed method is comparable or better than the state-of-the-art topic modeling algorithms which further statistically verified with the Wilcoxon Rank-sum Test.

Keywords Granularity · Point-wise mutual information · Graph-based modeling · Hierarchical structure · Computationally efficient algorithm

1 Introduction

Advancement of technology, wide use of the internet in almost every area, the advent of cheap and fast storage result in an explosive growth of digitized unstructured text documents. Unstructured text documents are easily available to users via different online news sites, blogs, and social networking sites. Natural language processing (NLP) [27], a sub-field of AI is used for analyzing large text and extract structured knowledge to perform different tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation. However, the NLP systems without prior knowledge often fail to resolve the ambiguity in the complex and diverse text and depend on huge computational resources to process the text. Document classification plays a major role in several text mining applications like organizing, information retrieval, and consciously representation of large voluminous documents. However, two main factors make document classification a challenging task: *feature extraction* and *topic ambiguity*. Firstly, extracting the right set of features plays a crucial role in evaluating the performance of the model, which mainly depends on human perception and therefore, expensive [51]. Secondly, the documents are voluminous, unstructured, containing hidden information, for which

✉ Isha Ganguli
ishacst.rs2016@cs.iiests.ac.in

Jaya Sil
js@cs.iiests.ac.in

Nandita Sengupta
ngupta@ucb.edu.bh

[1] Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

[2] Department of Information Technology, University College of Bahrain, Janabiyah, Bahrain

Published online: 13 January 2021

precise classification of the documents is very difficult without prior knowledge. "Topic" signifies the hidden relations and inter-connectivity among the words in the document, disclosing the theme of it. Topic modeling is a form of unsupervised learning used to identify the topics by analyzing a large text corpus where the set of possible topics is not known a priori. Our research is intended to design a computationally efficient method that precisely identifies the topics, hidden in the large text corpus by encapsulating hierarchical granularity concept and graph-based approach.

Very few works on topic modeling have been reported [31, 67] based on the state-of-the-art methods like latent Dirichlet allocation (LDA) [4]. Different classical topic modeling methods like latent semantic analysis (LSI) [9], LDA, Guided-LDA (GLDA) [28], Labeled LDA (LLDA) [52] also require prior knowledge about data which is very hard to obtain. Different text mining techniques [19, 29, 58] are available to discover the topics of documents, however, they lack in identifying the context, resulting in loss of information. Moreover, several text mining and web mining techniques [3, 48] are not capable of analyzing unstructured data such as PDF files and semi-structured textual data without transforming it into a structured form, which often requires huge computation time. Though some widely used text mining techniques like Bag of N-gram models [57], term frequency-inverse document frequency (TF-IDF) model [70] takes less computation power but at the same time, they focus only on the generated feature vectors and their similarity measures. This lags in encapsulating the proper contextual information about the data. Recently, deep learning-based [30, 44] techniques using recurrent neural network (RNN) [7] and convolution neural network (CNN) [18, 64] are developed to extract contextual information from the large unstructured text corpus using a sequence of words [10, 35]. However, the real challenges are the requirement of huge computational resources, model complexity, time to learn the parameters, and setting of hyper-parameters for categorizing the documents into topics.

The objective of this paper is to classify the test document into a particular topic by extracting contextual information from the training documents, without involving huge computational resources and prior knowledge. The proposed nonparametric granularity-based word grouping (GBWG) algorithm generates granule of words (GoWs) hierarchically based on the co-occurrence between the words of different granules, evaluated in a computationally efficient way. The algorithm terminates when no new GoW has been generated, anymore. The GoWs thus obtained coarsely represent the topics consisting of common words, resulting in ambiguity in identifying the topics. To remove ambiguity, a graph-based approach has been proposed where point-wise mutual information (PMI) between the words (nodes) of different GoWs is used to cluster the documents by applying Louvain graph clustering algorithm [5]. Each cluster consisting of unique words represents a particular topic. The flow of the proposed method is given in Fig. 1.

Contributions of this paper are summarized as follows:

(i) Using the granularity concept, the proposed non-parametric GBWG algorithm without prior knowledge coarsely classify the large documents into different topics.
(ii) The information has been transferred hierarchically to generate new GoWs at successive layers, without any loss of information.
(iii) The proposed method requires less computational resources and less time using a simple layered architecture.
(iv) The graph-based approach is used to categorize the documents into different topics consisting of unique words, therefore unambiguous.
(v) Detailed time complexity analysis has been discussed to exhibit the efficiency of the proposed method.

The main focus of this work is to propose the novel computationally efficient topic identification method, which clusters the words of the documents without a priori knowledge and each cluster precisely represents a particular topic. It is computationally efficient because the features are automatically extracted as embedded GoWs, like deep learning methods. However, the proposed method does not require high computational resources, unlike deep

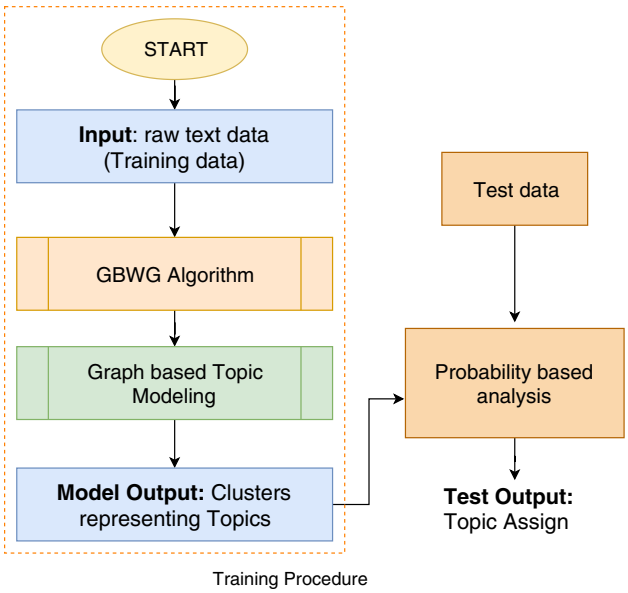


Fig. 1 Flow of the proposed method

learning methods. The time complexity of the proposed method is less or comparable with the time complexity of LDA. If the number of topics in the document is less, then the proposed method is comparable with LDA. In case there is large number of topics in the document, then the time complexity of our method is lower than LDA [59].

The paper has been divided into the following sections. Section 2 presents a literature study on topic modeling. Section 3 describes the GBWG algorithm while the graph-based topic modeling approach is given in Sect. 4. Section 5 contains the detailed time complexity analysis of the proposed method. Experimental results and the comparison with state-of-the-art methods are provided in Sect. 6. Finally, the conclusion is given in Sect. 7.

2 Literature review

This section provides a brief overview of the works on text processing, more precisely on topic modeling, document clustering, and classification tasks.

2.1 Topic modeling

The importance of text data analysis and the task of obtaining the latent topics from it is progressively increasing and some well-known methods are widely in use. A mathematical linear projection technique latent semantic analysis (LSA) [9] has been proposed, which is easy to understand, implement, and has usage. However, the LSA method is not efficient for handling nonlinear dependency which leads to incompetent results. The model is also not properly understood because of its dense hierarchical representation. Another widely used probabilistic topic modeling method LDA [4] interprets the generation of different topics. LDA has huge usage in different fields including author-topic analysis [40], supervised topic models [52], latent Dirichlet co-clustering and LDA-based bioinformatics [50] etc. However, LDA requires prior knowledge, i.e., number of topics like LSA. A hierarchical extension of LDA (HDP) [68] has been reported which learn the topics automatically and can be unbounded. However, it is more complicated to implement and unnecessary in the case where a bounded number of topics is acceptable. Besides these unsupervised approaches, a semi-supervised version of LDA, Guided-LDA (GLDA) [28] has been proposed where the procedure is guided by the selection of seed words, used as representative of the underlying topics in a corpus. Though the GLDA outperforms the traditional LDA in many cases, it is not easy to choose the seed words for underlying topics, which need knowledge about the data. A supervised improvisation of LDA and a generative model Labeled LDA (LLDA) [52] have been proposed which learn the one-to-one correspondence between LDA's latent topics and user tags. Though the LLDA outperforms most of the traditional topic extraction methods including support vector machines (SVM), but supplying the number of topics as prior knowledge is difficult for real-life implementation. Very recent method correlation explanation (CorEx) [21] approach has been reported which learns maximally informative topics as encoded by their total correspondence instead of relying on the detailed assumptions and specification of hyper-parameters. However, CorEx relies on binary count data in its sparsity optimization rather than the standard count data which may hamper the efficiency in case of larger documents and increase computational complexity. A novel discriminative variant of LDA, i.e., logistic LDA [37] has been proposed that can be applied to groups of images, arbitrary text embeddings, and integrates well with deep neural networks. Despite being a discriminative model, it has been shown that logistic LDA can learn from unlabeled data in an unsupervised manner by exploiting the group structure present in the data. In a very recent task [42], discriminative topic mining takes a set of user-provided category names to mine discriminative topics from text corpora. This task helps a user to understand the topics clearly and distinctively about the topics he/she is interested in and also benefits directly to the keyword-driven classification tasks. Hierarchical topic mining [43] has been proposed to guide the hierarchical topic discovery process with minimal user supervision. It takes a category tree described by category names only and aims to mine a set of representative terms for each category from a text corpus to help a user comprehend his/her topic of interest. It is a novel joint tree and text embedding method along with a principled optimization procedure.

2.2 Document clustering

In the field of text clustering, standard clustering algorithms such as K-means, hierarchical clustering, singular value decomposition, and affinity propagation have been successfully used [53]. Different other approaches like spectral graph analysis [6], sparse matrix factorization [63], probabilistic models [41] were also used to improve the performance. These aforementioned works depicted that feature selection [26] and dimension reduction [22] plays major role in the task. Most of these traditional approaches require access to prior knowledge of the number of clusters, which in certain real-world circumstances is not always possible. To overcome this problem, the Dirichlet process mixture model (DPMM) predicts the arbitrary number of clusters automatically [73, 74] in the post-inference process. Gibbs sampling [13, 45] can be applied to infer cluster assignments in these models. Very recent work

has been done by Duan et. al. in their proposed SiDPMM [12] model where clustering takes place by extracting sequential features using the encoder-decoder model; a collapsed Gibbs sampling algorithm is also derived in the task for enabling efficient inference. Besides these conventional methods of clustering, evolutionary algorithms [32] and graph algorithms [56, 60] are also being used in text clustering. There has also been a lot of similar studies, but with a different reach. Dorpinghaus et al [11] have mapped the clustering problem as a graph-theoretical problem where firstly a suitable similarity measure for the data domain is identified followed by a graph-based approach.

There has also been a lot of similar studies, but with a different objectives in various domains where document clustering is used to achieve further goals in different domains like search query [25, 36], automatic classification of documents [47, 65], topic modeling [62, 71], recommendation system [2, 54] etc. Document classification is a domain where the text clustering is hugely used [16, 20]. Traditional supervised document classifiers require a large number of the labeled data set which is very expensive. Hingmire [24] proposes LDA-based document classification method which does not require any labeled data set. Power [51] proposes a simple feature extraction algorithm that achieves high classification accuracy in the context of development-centric topics based on *popularity* and *rarity* matrices. Another very interesting work regarding the classification of research papers is based on term frequency-inverse document frequency (TF-IDF) and LDA schemes have been reported by Kim [34]. The proposed classification system clusters the research papers, which are very likely to have similar subjects. The system extracts representative keywords from the abstracts of each paper and topics are obtained using the LDA scheme. Then, the K-means clustering algorithm is applied to classify the papers with similar subjects based on the TF-IDF values of each paper.

Different deep learning-based techniques like RNN, CNN are applied for the task to identifying the latent topics from large documents [10, 35, 66]. Besides topic modeling, deep learning-based methods are also used in document classification [35, 38, 72]. Document representation and classification using deep neural network models for short and long documents [39] have been proposed. However, a large text corpus needs a costly computation platform and time for learning the parameters. Otherwise, we need a pre-trained network where hyper-parameters are tuned for a new set of a text corpus. Deep architectures use a parametric learning procedure, which increases computational cost and time with the size of the text data.

3 Coarsely generated topics: granularity-based word grouping (GBWG) algorithm

The proposed GBWG algorithm in a computationally efficient way generates GoWs hierarchically, using a corpus of documents. Any two GoWs of a layer may generate a new GoW at its higher layer using co-occurrence analysis of the words in different granules. The GoWs of different layers, when do not generate a single new GoW at its respective higher layer, the algorithm terminates and the GoWs obtained at termination represent different topics.

3.1 Preprocessing

The text corpus consisting of multiple documents is pre-processed to obtain a sequence of dictionary words. *Stop words* which frequently appear in the documents but not significant (like "The," "as," "a," "anyone" etc) are removed from the corpus. Different inflected forms of the basic word like "category," "categories" and "categorizing" are *lemmatized* and the word "category" is kept in the corpus. In the work, we consider different forms of nouns and verbs as raw input because they mostly carry context, related to different topics. If the same word appears in consecutive positions of a sequence, we consider it once only. Let, a document is: "run Lola run. run fast." Word *run* is placed consecutively in third and fourth positions of the text sequence and we consider *run* only once, obtaining the sequence as: "run Lola run fast."

3.2 Word selection for input sequence

A corpus consisting of a large number of documents and each document contains different words, though all are not equally important to identify the topics automatically. It has been observed that usually, the documents contain many low-frequency words and few high-frequency words. High-frequency words contain more information and have the possibility to be distributed over the topics in comparison with the low-frequency words. However, low-frequency words may contain information that is essential for building the model. So, quantitative as well as qualitative text are used to select the input words by introducing a threshold, defined in equation (1). Based on the analysis, we determine a threshold, called *term frequency threshold* (*tfth*) for selecting the words from the corpus as input sequence to build the model.

Let the corpus contains n words and respective frequency of each word constitutes a set, say $F = \{f_1, f_2, \ldots f_n\}$. We define parameters $\alpha = \max(\{f_i | i \in 1, 2, \ldots, n\})$ and $\beta = \min(\{f_i | i \in 1, 2, \ldots, n\})$ while γ counts number of words having frequency β. The threshold

tfth is calculated using equation (1). The words with frequency greater than *tfth* have been selected and used as input sequence of words of the GBGW algorithm.

$$tfth = \left\lfloor \frac{\alpha}{\gamma} \right\rfloor \quad (1)$$

We heuristically define the threshold for selecting the input words. In a document, the number of words with minimum frequency (γ) is high, so the threshold is low (see, equation (1)). Therefore, both the low and high-frequency words are selected as input.

3.3 Formation of GoWs

The proposed GBWG algorithm uses Look-up-Table-I and Look-up-Table-II to generate GoWs at different layers in a computationally efficient way as described below.

Look-up-Table-I

At the first layer of the proposed hierarchical structure, the preprocessed input sequence of words is fed and we slide a fixed size window across the sequence with stride one. Any two distinct words in each window, as appear in the sequence are paired as granule and each granule is searched in the corpus as co-occurred words, irrespective of order. The co-occurred granules in the first layer are sorted alphabetically and stored in Look-up-Table-I as paired GoWs.

Look-up-Table-II

Next, we slide a window of size three over the input sequence. Three successive words together starting from the first word of each window are defined as *triple_of_words*. If any two words in a *triple_of_words* are the same, we do not consider it for further processing as it does not carry important semantic. The *triple_of_words* are searched in the corpus as co-occurred words irrespective of order and if found are sorted and stored in Look-up-Table-II. Each word of a *triple_of_words* in Look-up-Table-II is strongly associated with its predecessor (except the starting word) and successor (except the end word), so represent the important context and used to form granules at successively higher layers.

Few terminologies are given here to illustrate the GoW formation procedure at different layers:

- *LGoW and RGoW* Any two GoWs of a layer are used to build a new GoW at its higher layer, one is denoted as left-GoW (LGoW) and the other as right-GoW (RGoW). Say, two GoWs (*politics, role*) and (*play, country*); (*politics, role*) is considered as LGoW and (*play, country*) as RGoW.
- *CWPG* In a particular layer, by scanning the LGoW and RGoW from left to right, we build a sequence consisting of the last word of LGoW, and the first word of RGoW, defined as Connecting Word Pair Granule (CWPG). CWPG for earlier example is (*role, play*).

  Symbolically we may say that the last element of y_i and the first element of y_j are evaluated as functions $last(y_i)$ and $first(y_j)$, respectively, where y_i is denoted as LGoW and y_j is denoted as RGoW. The sequence of text, represented as: $[last(y_i)\ ,\ first(y_j)]$ is defined as CWPG.
- *LSW-triple* The predecessor of $(last(y_i))$ is evaluated as function $(pred(last(y_i)))$ and the sequence of text, named LSW-triple is represented as: $[(pred(last(y_i))), last(y_i), first(y_j)]$.

  In the previous example, $last(y_i)$ is "role" and $pred(last(y_i))$ is the predecessor word of "role," i.e., "politics."
- *RSW-triple* Similarly, corresponding RSW-triple is: $[last(y_i), first(y_j), (succ(first(y_j)))]$ where function $(succ(first(y_j))$ is evaluated as successor of $(first(y_j))$.

  Here the $first(y_j)$ is "play" and $(succ(first(y_j))$ gives the successor word of "play," i.e., "country." For the earlier example (*politics, role, play*) is LSW-triple and (*role, play, country*) is RSW-triple, where corresponding CWPG is (*role, play*).

In the proposed GBWG algorithm, LGoW and RGoW of layer L are used to generate new GoW at its successive higher layer, i.e., layer L+1, where $L > = 1$, if the following conditions are satisfied:

Condition 1 A CWPG of layer say, L must be present in Look-up-Table-I, irrespective of the order of words.

Condition 2 The corresponding LSW-triple and RSW-triple of layer L must be present in Look-up-Table-II, irrespective of the order of words.

Both the conditions are evaluated for generating new GoWs at layer $L + 1$ after merging any two GoWs (LGoW and RGoW) of the previous layer (L). The GBWG algorithm first searches the CWPG of layer L in Look-up-Table-I to satisfy Condition 1 and if found then search Look-up-Table-II for corresponding LSW-triple and RSW-triple to satisfy Condition 2.

Neighboring words in a sequence are more contextually related compared to the distant words. Experimentally, we have chosen three as the size of the window (i.e., w = 3). The fixed-size window has been scanned across the sequence of input data to generate GoWs in the first layer. Therefore, for each window maximum of three pairs of granules are generated (assuming each word in the window is distinct), irrespective of the order of appearance of the words. To generate GoWs in the second and subsequent layers, both condition 1 and condition 2 (check Look-up-Table-II) are to be satisfied.

For generalization, let us consider any two GoWs $y_i(LGoW)$ and $y_j(RGoW)$ of Lth layer for generating new GoWs at $(L+1)$th layer where i and j vary from 1 to no.-of-GoWs, present in the Lth layer and $i \neq j$.

According to condition 1, the CWPG should be present in Look-up-Table-I.

According to condition 2, if both the LSW-triple and RSW-triple are present in Look-up-Table-II, then the new GoW(y_i, y_j) consisting of words is generated at $(L+1)$th layer by analyzing GoWs y_i and y_j of Lth layer.

The procedure for generating GoWs at different layers are continued till no new GoW is obtained and the algorithm outputs different GoWs, each of which coarsely represents different topics of the corpus.

By satisfying Condition 1 and Condition 2, now information of LGoW and RGoW of layer L has been transferred hierarchically to form new GoWs at a higher layer $(L+1)$ without involving huge computing resources and time as explained below.

Assume a set X consisting of elements LSW-triple, CWPG, RSW-triple of a particular layer L. A homogeneous relation R on the set X is a transitive relation, if LSW-triple R CWPG and CWPG R RSW-triple, then LSW-triple R RSW-triple. According to the GBWG algorithm, since LSW-triple and RSW-triple of a layer are formed in association with the corresponding CWPG, we observe that based on the homogeneous relation R, LSW-triple has an association with RSW-triple (LSW-triple R RSW-triple). Moreover, CWPG, LSW-triple, and RSW-triple are defined in terms of LGoW and RGoW of a particular layer, therefore, the GBWG algorithm generates new GoWs at layer $L+1$ by merging LGoW and RGoW of layer L.

This algorithm is computationally efficient because to generate a new GoW at layer $L+1$, we only look for CWPG of layer L in Look-up-Table-I and corresponding LSW-triple and RSW-triple in Look-up-Table-II. In the new GoW, the remaining words, if any (other than CWPG, LSW-triple, and RSW-triple), already satisfied two conditions at its earlier layers, so they have relations. It is worth mentioning that the corresponding LGoW and RGoW are generated in the earlier layer based on the same logic and thus information has been transferred hierarchically. Therefore, the proposed method is computationally efficient, which does not require the learning of huge parameters using a feedforward approach. The procedure of GoW formation continues layer by layer and the algorithm terminates when no new GoW has been generated.

Figure 2 illustrates an example of GBWG algorithm. In this example, let's consider LGoW in the first layer is (*politics, role*) and RGoW is (*play, country*). Based on the aforementioned conditions, these two GoWs are merged to generate new GoW of four words, say ((*politics, role*); (*play, country*)) in the 2^{nd} layer. The CWPG (*role, play*) is present in Look-up-Table-I, satisfying Condition 1 while LSW-triple (*politics, role, play*) and RSW-triple (*role, play, country*) are found in Look-up-Table-II, satisfying Condition 2 as well. Similarly in the next higher layer, GoW of eight words ((*politics, role, play, country*); (*role, smoke, country, drink*))are generated with LGoW (*politics, role, play, country*) and RGoW (*role, smoke, country, drink*) from 2^{nd} layer. The CWPG (*country, role*) is present in Look-up-Table-I, satisfying Condition 1 while LSW-triple (*country, role, smoke*) and RSW-triple (*role, smoke, drink*) are found in Look-up-Table-II, satisfying Condition 2 as well. The procedure continues and at termination we obtain the GoWs, coarsely representing different topics.

Lemma 1 *The GBWG algorithm ensures no loss of information.*

Proof In the proposed GBWG algorithm, information has been embedded in the GoWs, generated in different layers. For generating GoWs at the first layer, we consider a fixed size window, and the window is scanned across the sequence of words of the preprocessed corpus. Pair of co-occurred words in each window are searched in the corpus and if found, GoWs of length two are generated at the first layer. In the subsequent layers, any two GoWs (LGoW and RGoW) of the corresponding lower layer are analyzed to generate GoWs at its higher layer.

Information transmission is measured using conditional entropy, which quantifies information content in the GoWs (random variables), generated at a particular layer based on the correlated GoWs, corresponding to its previous layer. Conditional entropy of the First Layer:

$$H(W_1|W_2) = -\sum_{w_2 \epsilon W_2} \sum_{w_1 \epsilon W_1} p(w_1, w_2) \log_2[p(w_1|w_2)/p(w_2)].$$

In the first layer GoWs consist pair of co-occurred words, present in the corpus. Say, W1 is the set of words co-occurred with the words of set W2. We do not consider the order of appearance of the words in the sequence, so W1 and W2 may be interchanged in the equation. Conditional entropy of second and subsequent layers:

$$H(W_i|W_{i+1}) = -\sum_{w_{i+1} \epsilon W_{i+1}} \sum_{w_i \epsilon W_i} p(w_i, w_{i+1}) \log_2[p(w_i|w_{i+1})/p(w_{i+1})].$$

where $i \geq 1$ and $GoW_{L_{i+1}}$ is the generated GoW at layer L_{i+1} using LGoW ($GoW_{L_i}^{Left}$) and RGow ($GoW_{L_i}^{right}$) of layer L_i.

According to the GBWG algorithm, at the higher layers, the formation of GoWs is less likely events due to less possibility and more uncertainty to satisfy Condition 1 and Condition 2, compare to the lower layers. Therefore, GoWs

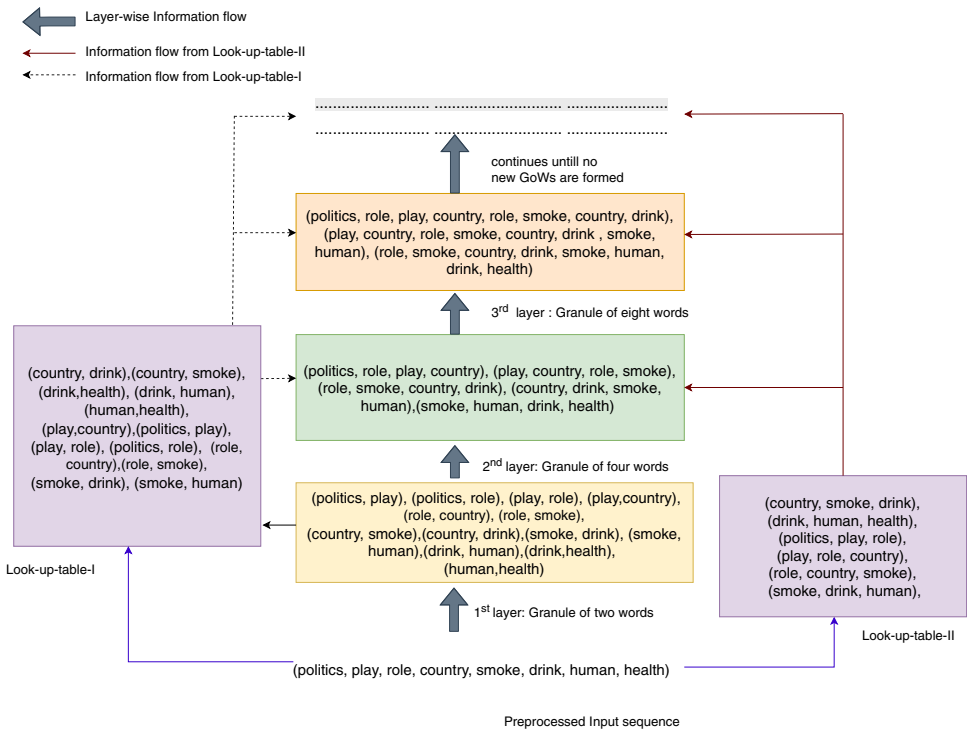


Fig. 2 Illustration of GBWG Algorithm using Window size three

at higher layers is more informative than learning the likely events which are generated at the lower layers. It is worth mentioning here that in this method, the number of GoWs at a lower layer, usually is more than the GoWs at its higher layer.

Therefore, $H(W_1|W_2) \geq H(W_2|W_3) \geq H(W_3|W_4)$.

This formulation implies that the conditional entropy decreases with the increase of layer number, which indicates an increase of average information content. So, we observe that the proposed GBWG algorithm ensures no loss of information. □

Algorithm 1 GBWG Algorithm

Begin
Input word sequence based on *tfth*, $P = \{p_1, p_1 \ldots\ldots p_m\}$
Empty lists: X, S, $S1$; window-size: w
for $\forall i = 1 : (m - w + 1)$ **do**
  for $\forall j = (i + 1) : (i + w)$ **do**
    C1= **concatenate_words** (p_i, p_j)
    append$(X, C1)$ //**lookuptableI* formation*//
  end for
end for
lookuptableII similarly created with *triple_of_words*
//*Searching in *lookuptableI* and *lookuptableII* *//
$S1 = X$
while $(S \neq S1)$ **do**
  $S = S1$
  for $\forall i = 1 : (len(S) - w + 1)$ **do**
    for $\forall j = (i + 1) : (i + w)$ **do**
      $a = freq\,(s_i\,[len\,(s_i) - 1]\,, s_j\,[0])$
      $b = freq\,(s_i\,[len\,(s_i) - 2]\,, s_i\,[len\,(s_i) - 1]\,, s_j\,[0])$
      $c = freq\,(s_i\,[len\,(s_i) - 1]\,, s_j\,[0]\,, s_j\,[1])$
      If $((a > 0)$ **and**$(b > 0)$ **and**$(c > 0)$)
      C2=**concatenate_words** (s_i, s_j)
      append$(S1, C2)$ **end if** //*Forming granule*//
    end for
  end for
end while
End

4 Graph-based topic modeling

After applying the GBWG algorithm, we obtain GoWs where words in each granule are contextually related. However, the GoWs may overlap due to common words and create ambiguity in topic modeling. The graph-based method has been proposed to generate the GoWs with unique words, representing different topics precisely. Words of the GoWs are represented by nodes and each pair of words in a particular GoW is connected by edges.

Figure 3 shows the generated graph using the GoWs, say, $(g1, g2, \ldots, g8)$, obtained by the GBWG algorithm: $g1 =$ (politics, parliament, power), $g2 =$ (politics, party, parliament, man), $g3 =$ (party, power, man, election), $g4 =$ (election, vote, power), $g5 =$ (star, party, vote), $g6 =$ (sport, star, cricket), $g7 =$ (cricket, football, sport, match) and $g8 =$ (football, match, baseball, sport).

4.1 Graph reconstruction

In the graph, the association between the words is measured using PMI and represented as the weight of the edge, described in equation (2).

$$word\_pmi = \log_2\left(\frac{\frac{count(P\cap Q)}{N}}{\frac{count(P)}{N} * \frac{count(Q)}{N}}\right) \tag{2}$$

In equation (2), $count(P)$ and $count(Q)$ denote the frequency of words P and Q, while $count(P \cap Q)$ denotes the frequency of co-occurrence of words P and Q in the corpus and N denotes the total number of nodes in the graph.

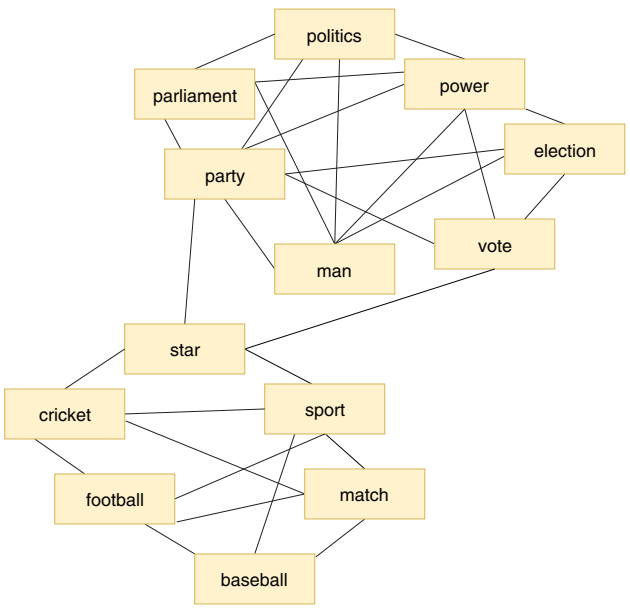


Fig. 3 Graph-based representation of the groups

We determine a threshold (*pmi_th*) to select the nodes (words) which are more relevant corresponding to each GoW, representing different topics. Words in a GoW are strongly associated with having *word_pmi* value in the respective edge greater than the *pmi_th*. The value of *pmi_th* is dependent on the data set and determined by analyzing the histogram representing the corresponding *word_pmi* values. We experimentally consider *pmi_th* value as the midpoint of the monotonically increasing region of the best-fitted curve on the histogram. Figure 4 illustrates the procedure for a particular data set *Buss_ent* (Data set description given in Sect. 6) where midpoint of the positive slope is 0.3, considered as the *pmi_th* value.

It has been observed that there may exist few nodes which are connected with multiple nodes but *word_pmi* value of the respective edges is less than the corresponding *pmi_th*. In this case, we select the nodes (words) connected by the edge(s) of the sub-graph with the highest *word_pmi* value due to their association with multiple words of GoW(s), and included in the reconstructed graph. Otherwise, there is a chance of information loss, resulting in false cases in prediction. We now reconstruct the graph by selecting the significant nodes and corresponding edges.

Figure 5 depicts the reconstructed graph $G1$ from graph G with *pmi_th* value 0.35. Edges (*a,d*), (*d,k*), (*k.l*), (*l,n*), (*n,p*), (*h,j*) and (*f,h*) have *word_pmi* value greater than *pmi_th*. So, the edges along the corresponding nodes are present in graph $G1$. Node *f* is connected with node *g*, *e* and *i* but *word_pmi* values of respective edges are less than *pmi_th* value (0.35). According to the reconstruction method, (*f,e*) edge is retained as its *word_pmi* value is greater than the edges (*f,g*) and (*f,i*). Similarly node *i* is connected with nodes *f* and *j*. But only edge (*i,j*) is retained as it has greater *word_pmi* value than edge (*i,f*).

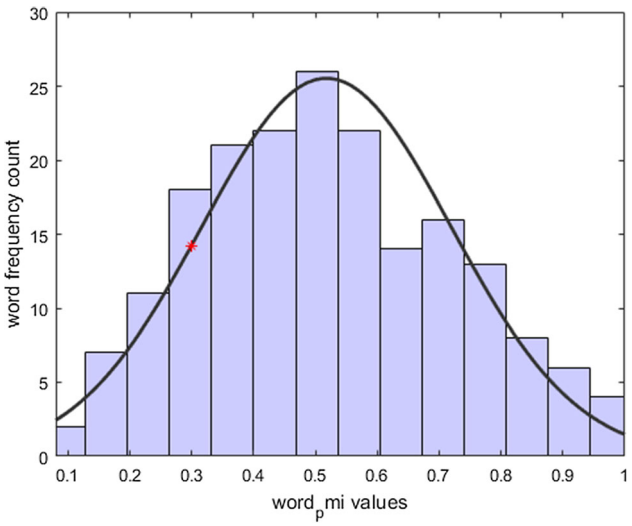


Fig. 4 Histogram of *word_pmi* of *Buss_ent* data set

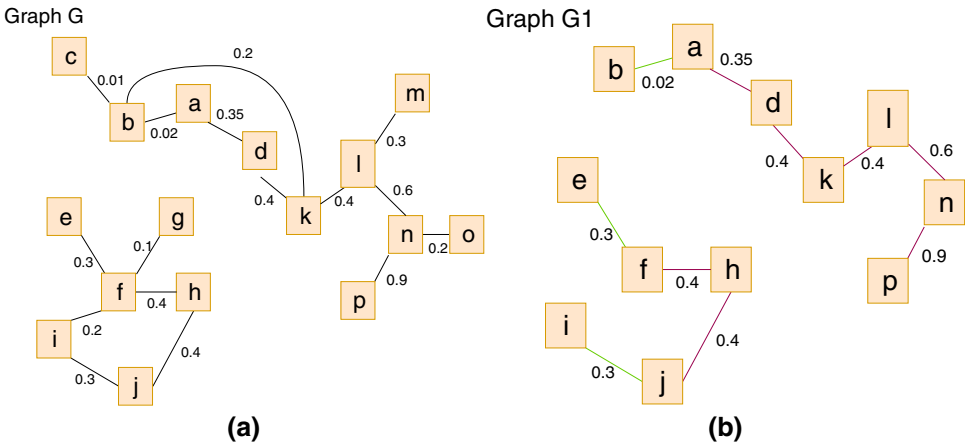


Fig. 5 Graph reconstruction

4.2 Topic generation

Finally, "Louvain graph clustering" algorithm [5] is applied on the reconstructed graph $G1$ to obtain the clusters, representing different topics unambiguously. Each cluster contains unique words related to a particular topic and the result is interpretable due to the graph structure.

The "Louvain graph clustering" [5] algorithm is a community detection algorithm that has been developed based on the density between the nodes of a network. Densely connected nodes belong to the same community or cluster whereas the connection between different communities is sparse. The "modularity" measure is used as the objective function, given in equation (3) for calculating the density of edges inside a community compared to the edges outside the community.

$$Q = (1/2\rho)\sum_{ij}\left[A_{ij} - \left(k_i k_j/2\rho\right)\right]\delta\left(c_i, c_j\right) \tag{3}$$

where, A_{ij} represents the weight of the edge between the nodes i and j.

$k_i = \sum_j A_{ij}$ is the sum of weights of the edges attached to the node i and c_i is the community to which node i is assigned. A Boolean variable $\delta\left(c_i, c_j\right) = 1$ if $c_i = c_j$, 0 otherwise and $\rho = (1/2)\sum_{ij} A_{ij}$.

Figure 6 depicts the finally generated distinct topics, denoted by the formed clusters, after applying the Louvain Graph clustering algorithm for all the data sets with the different number of classes.

Performance improvement has been observed mainly for granularity concept where information has been embedded in the GoWs by using a hierarchical structure, as described in the GBWG algorithm. Louvain Graph clustering algorithm generates a number of clusters automatically, representing topics, and improves the precision of the model because each cluster consists of unique words. GoWs consist of contextually related words and are mapped as nodes of the graph. Connectivity between different GoWs is represented using mutual information in the graph. The graph is a generic representation of the GoWs, obtained by using the GBWG algorithm therefore, effective while integrated with other graph clustering algorithms.

5 Time complexity

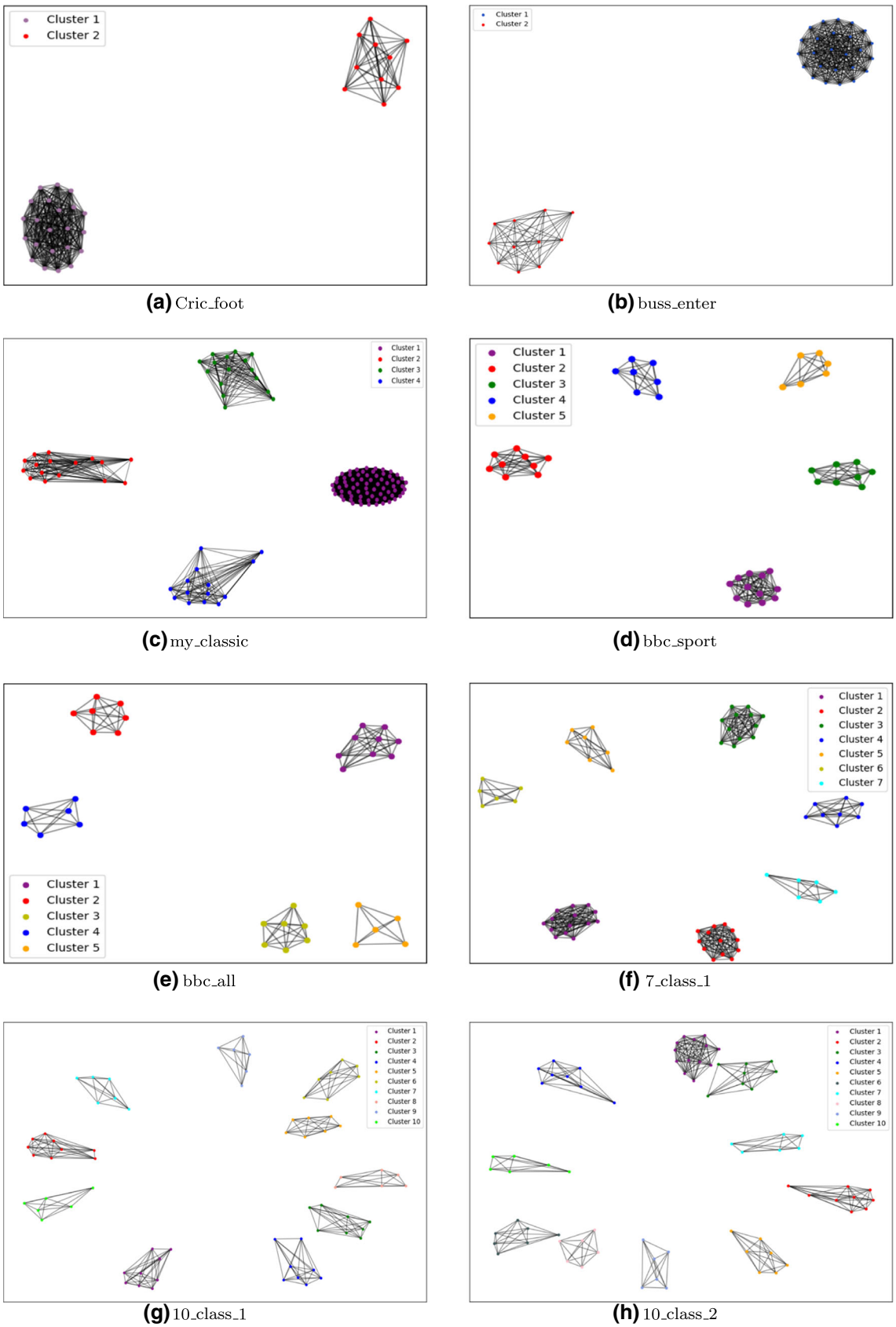
The time complexity of the proposed method has been described, assuming unit time required for the basic operations.

5.1 Time complexity of the GBWG algorithm

- Let *window size* is w and m number of selected words are fed as sequence of input to the first layer. Number of GoWs consisting of pair of words at the first layer is $(m - w + 1)\,{}^{w}C_2$. Say, m unit time is required to scan the input sequence for generating the GoWs of first layer. These pairs are stored in Look-up-Table-I, assuming the worst case that all pair of words are co-occurred.

  Time taken for generating GoWs in the first layer and sorting Look-up-Table-I is $m + \mathrm{O}(m\log m)$ because w is a constant and negligible w.r.t. m.
- Generation of Look-up-Table-II takes $c * (m - 3)$ time, where c is the constant time taken for concatenating each three consecutive words *(triple_of_words)* in the input sequence. Here, also we assume that all *triple_of_words* are co-occurred and sorting takes $\mathrm{O}((m-3)\log(m-3))$ time. So, time taken for generating and sorting Look-up-table-II is $c * (m - 3) + \mathrm{O}(m\log m)$.

  Therefore for first layer (L=1) total time taken is:



(a) Cric_foot **(b)** buss_enter

(c) my_classic **(d)** bbc_sport

(e) bbc_all **(f)** 7_class_1

(g) 10_class_1 **(h)** 10_class_2

Fig. 6 Generated community structures for all the data sets after applying Louvain Graph clustering

$$m + O(m\log m) + c * (m-3) + O(m\log m) \simeq (c+1) * m + 2 * O(m\log m).$$

- For layers $L > 1$,
  1. CWPG searching time in Look-up-Table-I is: $O(\log((m-w+1)\,{}^{w}C_2))$.
  2. LSW-triple searching time in Loop-up-Table-II is : $O\log(m-2)$.
  3. RSW-triple searching time in Loop-up-Table-II is : $O\log(m-2)$.

     So the total searching time for satisfying two conditions is:

     $O(\log((m-w+1)\,{}^{w}C_2)) + 2 * O(\log(m-2)) \simeq 2 * O(\log(m)) + O(\log(m))$; assuming ${}^{w}C_2$ is constant and negligible compare to m $\simeq 3 * O(\log(m)) = O(\log(m))$
- For L number of layers, worst case time complexity of GBWG algorithm is :

  first layer + $(L-1)$ layers of the hierarchical structure $\simeq m + 2 * O(m\log m) + (L-1) * O(\log m) = O(m\log m)$

  So time complexity of the proposed GBWG algorithm linearly depends on m.

5.2 Time complexity of graph modeling

- Let number of GoWs is K obtained after termination of GBWG algorithm and n is the maximum number of words in a GoW where $n << m$. Worst case time requirement for generating the graph is $O(Kn)$.
- Time required for calculating *word_pmi* value at each edge is constant and say, η unit time.
- Number of edges present in the graph is $K(n-1)$. Time required for selecting the edges based on *pmi_th* value is unity, considered as a comparison operation, executed for each edge. After removing the insignificant edges, remaining edges present in the reconstructed graph is less than or equal to $K(n-1)$.
- So the graph reconstruction time is $O(K(n-1) * \eta)$, i.e., $O(Kn)$ (η is negligible).

The graph clustering algorithm (Louvain Graph Clustering) uses the greedy optimization approach and time complexity is $O(Kn\log Kn)$. Total time complexity of the proposed method is:

$$\text{Time\_complexity} = O(m\log m) + O(Kn\log Kn)$$

6 Results and discussions

News articles from BBC News, i.e., "BBC sport" [23], "BBC" [23] and "Classic4" [8] are benchmark data sets considered for experimentation. Both "BBC" and "BBC sport" data sets include news of five different areas during 2004-2005. "Classic4" data set consists of four different document collections: CACM, CISI, CRAN, and MED. In our work, initially we have started with the data sets having less number of classes (2-class, 4-class, 5-class). For this, we have considered the existing corpus: 1. BBC news articles consisting of two data sets, i.e., "BBC Sport" and "BBC," each having 5 topics or classes and 2. "Classic 4" data set which consists of 4 topics. We measure the performance of the proposed model by varying the number of classes. For that we create data sets with 2 classes, considering documents ("cricket" and "football") from "BBC Sport" and also ("business" and "entertainment") from "BBC." Performance of the proposed model is observed by creating data sets with a higher number of classes (7-class, 10-class) by including topics from both the 5-class and 4-class data sets. We prepare eight different subsets of data using the benchmark data sets as given in Table 1.

The results of the proposed method are provided in two aspects (i) Cluster Quality Measurement and (ii) Classification Performance Measurement. Here, it is worth mentioning that since the proposed method uses a graph clustering algorithm, we provide additional two measures (modularity and coverage) which are not applicable for evaluating other methods, considered for comparisons with our method. Topic coherence [55] measurement has also been given to understand the semantic similarity of the words within a generated topic.

6.1 Cluster quality measurement

We evaluate quality of clusters based on five metrics, i.e : (i) Entropy, [33], (ii) Perplexity, [49], (iii) Purity, [15] (iv) Modularity, [1] and (v) Coverage. [1].

The modularity and coverage are evaluated as given in Table 2 for the proposed method considering eight different data sets. Modularity close to one indicates that the strength of the clusters to identify the topics is very high. The coverage value is one for each data set implies that intra-cluster similarity is very high. So in both aspects, the proposed method satisfies the criteria of quality clustering. It has been observed that the purity of clusters improves after applying graph modeling. This is obvious because each cluster now contains unique words, resulting in a more accurate topic generation.

Table 1 Description of data sets

Custom made data set name	Number of class	Original data set	Data set include areas	Number of documents	Total size	Average length of document
Cric_foot	2	BBC Sport	Cricket, football	389	809.4 kB	345.65
Buss_ent	2	BBC	Business, entertainment	896	1.4 MB	322.35
My_classic	4	Classic4	Cacm, cisi, cran, med	7095	5.3 MB	105.68
Sport_all	5	BBC Sport	Cricket, football, rugby, tennis, athletics	737	1.5 MB	334.69
News_all	5	BBC	Business, entertainment, politics, sport, tec	2225	5.1 MB	378.44
7_class_1	7	BBC Sport + BBC	Business, entertainment, cricket, football, politics, tech, tennis	2219	5.1 MB	380.93
10_class_1	10	BBC Sport +BBC+ Classic4	Business, cacm, cisi, cran, cricket, entertainment, football, med, pol, tech	9220	10.3 MB	171.84
10_class_2	10	BBC Sport +BBC+ Classic4	Athletics, business, cacm, cran, cricket, entertainment, football, pol, rugby, tennis	6674	7.4 MB	173.54

Table 2 Modularity and Coverage values

Data sets	Modularity	Coverage
Cric_foot	0.8167	1.0
Buss_ent	0.8234	1.0
My_classic	0.6735	1.0
Sport_all	0.9423	1.0
News_all	0.9657	1.0
7_class_1	0.9081	1.0
10_class_1	0.9731	1.0
10_class_2	0.9531	1.0

Though our method is unsupervised, we compare with recent semi-supervised methods such as, i.e., GLDA [28], CorEx [21] and supervised method LLDA [52]. Table 3 provides a comparison with unsupervised (LSA, LDA, HDP), semi-supervised (GLDA, CorEx), and supervised (LLDA) methods in terms of purity while Table 4 in terms of entropy and perplexity. Semi-supervised method GLDA performs best in terms of entropy and perplexity for all the data sets with labeled as well as unlabeled instances, in the presence of known seed words. Our method is slightly better than another semi-supervised method CorEx for all the data sets. However, the outcome of CorEx depends on random initialization and one may restart the CorEx several times and choose the one that explains the maximum correlation. We notice that the proposed method beats the state-of-the-art unsupervised methods for all the data sets, except one. Only for "My_classic" data set, LDA method outperforms our method in terms of entropy and perplexity. The supervised LLDA method results in comparatively

Table 3 Comparison of Proposed method with other methods based on purity

Data sets	Purity						
	LSA	LDA	HDP	GLDA	COREX	LLDA	Proposed method
Cric_foot	0.8273	0.6976	0.5380	0.6107	0.9309	0.7142	0.9473
Buss_ent	0.8195	0.7804	0.7500	0.5413	0.9347	0.7460	0.9842
My_classic	0.6170	0.6070	0.4420	0.4380	0.9840	0.6571	0.9785
Sport_all	0.8276	0.7180	0.4648	0.4329	0.9000	0.7222	0.9690
News_all	0.8308	0.7008	0.5566	0.3658	0.9333	0.7851	0.9906
7_class_1	0.8428	0.7336	0.5959	0.4183	0.9163	0.7875	0.9802
10_class_1	0.7097	0.6361	0.4916	0.3810	0.9291	0.7012	0.9930
10_class_2	0.6840	0.6265	0.5090	0.3803	0.9363	0.7186	0.9693

Table 4 Comparison of Proposed method with other methods based on entropy and perplexity

Data sets	Entropy							Perplexity						
	LSA	LDA	HDP	GLDA	COREX	LLDA	Proposed method	LSA	LDA	HDP	GLDA	COREX	LLDA	Proposed method
Cric_foot	0.2387	0.0768	0.0731	0.0167	0.0526	0.3341	0.0448	1.1799	1.0546	1.0519	1.0116	1.0371	1.2605	1.0315
Buss_ent	0.2251	0.1653	0.1935	0.0061	0.0804	0.2318	0.0535	1.1688	1.1213	1.1435	1.0042	1.0617	1.1742	1.0377
My_classic	0.2259	0.0952	0.1288	0.0088	0.1482	0.2425	0.2732	1.1695	1.0682	1.0933	1.0061	1.1081	1.1830	1.2084
Sport_all	0.6285	0.3215	0.2084	0.0258	0.1858	0.5367	0.1074	1.5459	1.2496	1.1554	1.0180	1.1374	1.4506	1.0772
News_all	0.3605	0.1815	0.3050	0.0076	0.1268	0.3815	0.0513	1.2838	1.1340	1.2354	1.0052	1.0918	1.3026	1.0361
7_class_1	0.5273	0.2578	0.4185	0.0305	0.1245	0.4565	0.0346	1.4412	1.1956	1.3365	1.0213	1.0901	1.3722	1.0242
10_class_1	0.4993	0.1805	0.3466	0.0179	0.2543	0.4956	0.0568	1.4135	1.1332	1.2715	1.0124	1.1927	1.4099	1.0401
10_class_2	0.5141	0.2167	0.4122	0.0174	0.2445	0.6129	0.1213	1.4281	1.1620	1.3307	1.0121	1.1846	1.8346	1.0877

Table 5 Comparison of classification Performance with LSA, LDA, HDP and Proposed Method

Data sets	LSA				LDA				HDP				Proposed Method			
	Precision	Recall	F1-score	Accuracy (%)	Precision	Recall	F1-score	Accuracy (%)	Precision	Recall	F1-score	Accuracy (%)	Precision	Recall	F1-score	Accuracy (%)
Cric_foot	0.7000	0.6500	0.6266	65	0.9675	0.9600	0.9600	96	0.5980	0.5500	0.4800	55	0.9545	0.9500	0.9498	95
Buss_ent	0.7700	0.6000	0.5238	60	0.9724	0.9200	0.9158	92	0.9000	0.90	0.9000	90	0.8847	0.8500	0.8465	85
My_classic	0.1539	0.3000	0.2034	30	0.6187	0.5500	0.5570	55	0.1964	0.3000	0.1825	30	0.6488	0.5700	0.5637	57
Sport_all	0.2100	0.3200	0.2300	32	0.8129	0.6800	0.6530	68	0.4495	0.3800	0.3488	38	0.6899	0.6000	0.6023	60
News_all	0.7300	0.4200	0.4285	42	0.6317	0.5200	0.5300	52	0.2751	0.2600	0.1866	26	0.7404	0.6200	0.6229	62
7_class_1	0.4789	0.2714	0.2464	27	0.5445	0.4714	0.4253	47.14	0.1600	0.1857	0.1390	18.57	0.6392	0.5142	0.5163	51.42
10_class_1	0.2939	0.2700	0.2100	27	0.3480	0.4000	0.3300	40	0.1636	0.2300	0.1532	23	0.5346	0.4932	0.4866	49.32
10_class_2	0.1067	0.2100	0.1359	21	0.4862	0.4700	0.4425	47	0.3582	0.3000	0.2420	30	0.5047	0.4800	0.4533	48

Table 6 Comparison of Classification Performance with semi-supervised GLDA, CorEx, supervised LLDA and Proposed Method

Data sets	GLDA				COREX				LLDA				Proposed Method			
	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy
Cric_foot	0.9166	0.9000	0.8989	0.9000	0.7770	0.6000	0.5238	0.6000	0.9166	0.9000	0.8989	0.9000	0.9545	0.95	0.9498	0.9500
Buss_ent	0.5000	0.5000	0.4500	0.5000	0.8125	0.8000	0.7979	0.8000	0.9166	0.9000	0.8989	0.9000	0.8847	0.8500	0.8465	0.8500
My_classic	0.3863	0.4250	0.3518	0.4250	0.6453	0.375	0.3275	0.3750	0.7925	0.5500	0.5542	0.5500	0.6488	0.5700	0.5637	0.5700
Sport_all	0.3411	0.2400	0.2053	0.2400	0.6551	0.4400	0.4500	0.4400	0.3682	0.3410	0.3400	0.3410	0.6899	0.6000	0.6023	0.6000
News_all	0.3244	0.3000	0.3034	0.3000	0.6928	0.4400	0.4575	0.4400	0.6980	0.4800	0.4677	0.4800	0.7404	0.6200	0.6229	0.6200
7_class_1	0.4526	0.4320	0.4100	0.4320	0.7253	0.5142	0.5597	0.5142	0.6836	0.6428	0.6138	0.6428	0.6392	0.5142	0.5163	0.5142
10_class_1	0.3229	0.3000	0.2597	0.3000	0.5369	0.3800	0.3650	0.3800	0.7784	0.66	0.6601	0.66	0.5346	0.4932	0.4866	0.4932
10_class_2	0.3417	0.3100	0.2900	0.3100	0.4110	0.3600	0.3277	0.3600	0.7588	0.6200	0.6354	0.6200	0.5047	0.4800	0.4533	0.4800

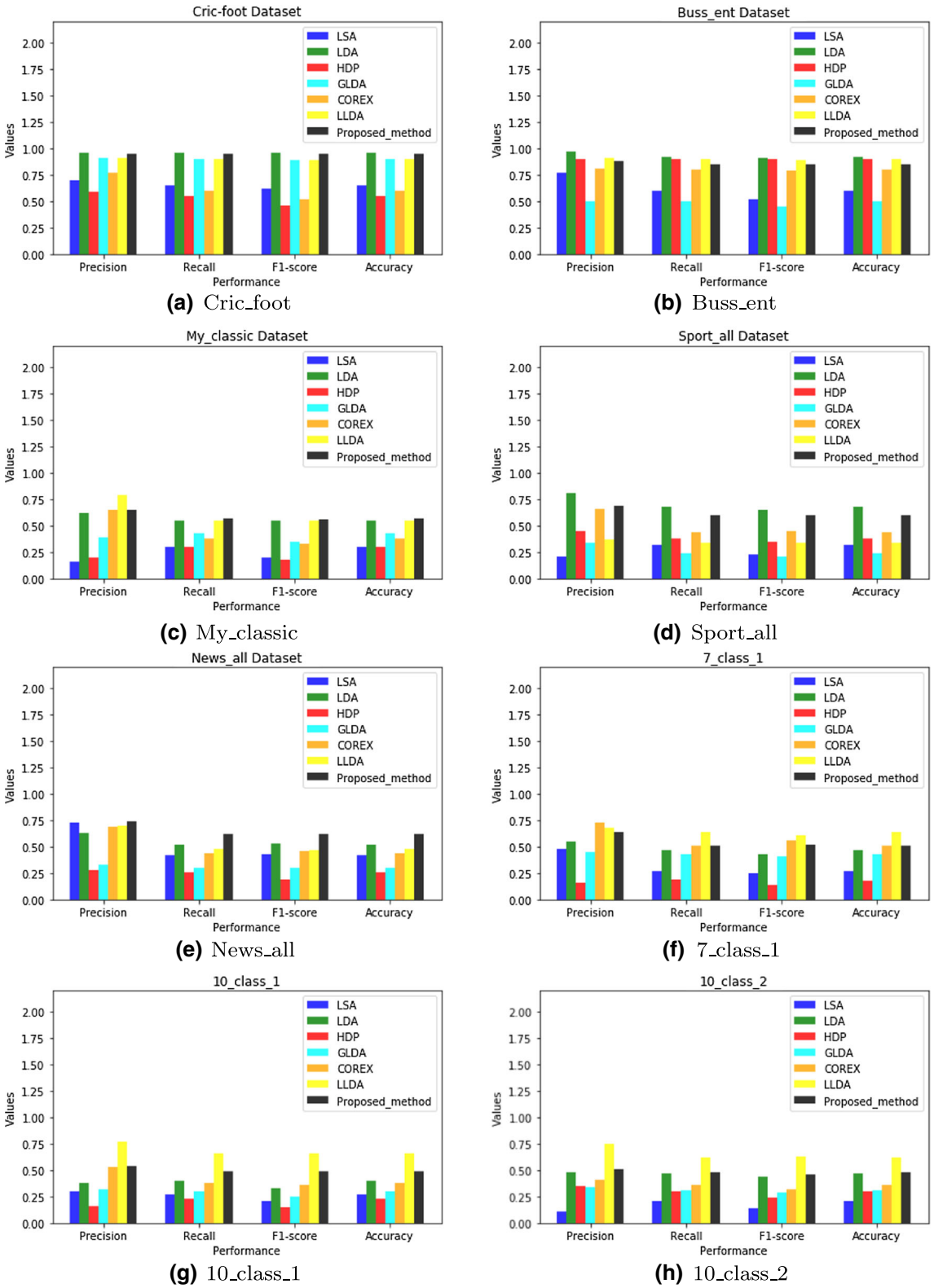
lowest performance due to the assignment of class labels to the documents with almost zero uncertainty. We observe that in terms of purity, the proposed method outperforms others (unsupervised, semi-supervised, and supervised) for all the data sets. This result is due to the presence of unique words in the clusters obtained using the proposed method.

6.2 Classification performance measurement

The accuracy is calculated using ten-fold cross-validation technique by dividing the data sets into "training data" and "test data." Randomly 25% of each data set is chosen as "test" input and the rest as "training" data. The model is built using the training data sets. The test data is preprocessed and we calculate the probability of unique words of the test data in different clusters. We sum the probability w.r.t. each cluster and the test data represents the topic corresponding to the cluster having maximum probability value.

Precision [14], Recall [14], F1-score, and Accuracy are calculated and compared with LSA, LDA, and HDP methods as shown in Table 5. The proposed method provides a statistically comparable result for the data sets with a moderate number of class labels while significant improvement has been observed for the data sets having a large number of classes. In Table 6, performance has been compared between the proposed method and semi-supervised GLDA, CorEx, and supervised LLDA methods. Our method beats the semi-supervised methods, GLDA, and CorEx for all the data sets in terms of accuracy, recall, and F1-score. In terms of Precision, the proposed method outperforms semi-supervised GLDA for all the data sets and gives comparable results with CorEx. The supervised LLDA method outperforms the proposed method for the data sets with a higher number of class labels. This observation is quite obvious due to the availability of prior knowledge. Classification Performance of different methods is shown in Fig. 7 for all the data sets.

The GBWG algorithm extracts contextual information from the documents by generating GoWs using a hierarchical structure. In the proposed framework, the GoWs identify the topics by analyzing the observed documents and infer the hidden topic structure. This can be thought of as "reversing" the generative process. Latent Dirichlet allocation (LDA) is a statistical topic model and described as a generative process. This generative process defines a joint probability distribution over both the observed and hidden random variables. It defines a topic to be a distribution over a fixed vocabulary and randomly chooses a distribution over topics. Choosing distribution over topic is a difficult task and approximates the distribution by estimating the random variables. Although the LDA and its variants (LLDA, supervised) achieve better or comparable



(a) Cric_foot **(b)** Buss_ent

(c) My_classic **(d)** Sport_all

(e) News_all **(f)** 7_class_1

(g) 10_class_1 **(h)** 10_class_2

Fig. 7 Comparison of classification performance by all the methods for all the data sets

Table 7 Results for Wilcoxon Rank-sum test between the proposed method and other methods

	Data sets	LSA			LDA			HDP			GLDA			CorEx			LLDA		
		p value	*h*	Sig	*p* value	*h*	Sig	*p* value	*h*	Sig	*p* value	*h*	Sig	*p* value	*h*	Sig	*p* value	*h*	Sig
Proposed Method	Cric_foot	2^{5}	1	+	8.8440e^{− 5}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+
	Buss_ent	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+
	My_classic	2.5621e^{− 34}	1	+	2.2609e^{− 33}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	9.5349e^{− 32}	1	+
	Sport_all	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+
	News_all	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+
	7_class_1	2.5621e^{− 34}	1	+	2.6403e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+
	10_class_1	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	0.34	0	-	2.5621e^{− 34}	1	+
	10_class_2	2.5621e^{− 34}	1	+	1.3596e^{− 24}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+	2.5621e^{− 34}	1	+

performance, especially for multi-class data sets compared to the proposed GBWG algorithm, however, the GBWG algorithm has its own advantages which include nonparametric nature and over-fitting of data does not arise.

Wilcoxon Rank-sum Test [69] is a nonparametric alternative to the two-sample *t* test, based on the order of the observations from the two samples. In our work, Wilcoxon Rank-sum Test is evaluated to show the significance of the results obtained by the proposed method and the other state-of-the-art methods. Value of *h* is 1, indicates a rejection of the null hypothesis, and the value of h is 0 indicates a failure to reject the null hypothesis at the 5% significance level. In Table 7, the symbols "+" and "−" indicates the significance and non-significance of the performance of the proposed method with respect to the state-of-the-art methods. Table 7 shows that in all the cases (except one) the p values are less than that of 0.05 significance level value and h value is 1. Both the p value and h indicate the rejection of the null hypothesis of equal medians at the default 5% significance level. Only for data set 7_CLass_1, the p value is 0.34 (greater than significance level), and h value is 0 which indicates that there is not enough evidence to reject the null hypothesis.

The proposed method is compared with two recent topic modeling works: Logistic LDA [37] and Discriminative topic mining via category-name guided text embedding [42] in terms of both Cluster quality measurement and Classification accuracy. Results are given in table 8. The proposed method outperforms both the methods (Logistic LDA and Discriminative) in terms of purity of topics, except for the two-class data set Buss_ent. In terms of entropy and perplexity, the proposed method gives comparable results with the discriminative method but not the Logistic LDA. In terms of Precision, Recall, F1-score, and Accuracy, the proposed method outperforms the Logistic LDA in all the data sets. However, with the increasing number of classes, the discriminative method gives better performance than the proposed method. In this context, it is notable that unlike the proposed method, both the Logistic LDA and Discriminative method needs human intervention. Both need number of topics as a priori knowledge and the discriminative method even needs the name of the input folders (i.e., name of topics) as an existing word in the corpus.

In Fig. 8, the ROC (Receiver operating characteristics) curves [17] for each data set reveal the prediction performance of the proposed method. Here, the detailed analysis of the ROC curves for multi-class data sets is presented. Figure 8a, b shows that the correct prediction of class 1 is slightly better than class 2, though both the classes have good prediction accuracy. In my_classic data (Fig. 8c), class 4 is correctly classified with around 50% prediction accuracy, whereas other classes perform better accuracy,

Table 8 Comparison of Proposed method with Logistic LDA and Discriminative topic mining method

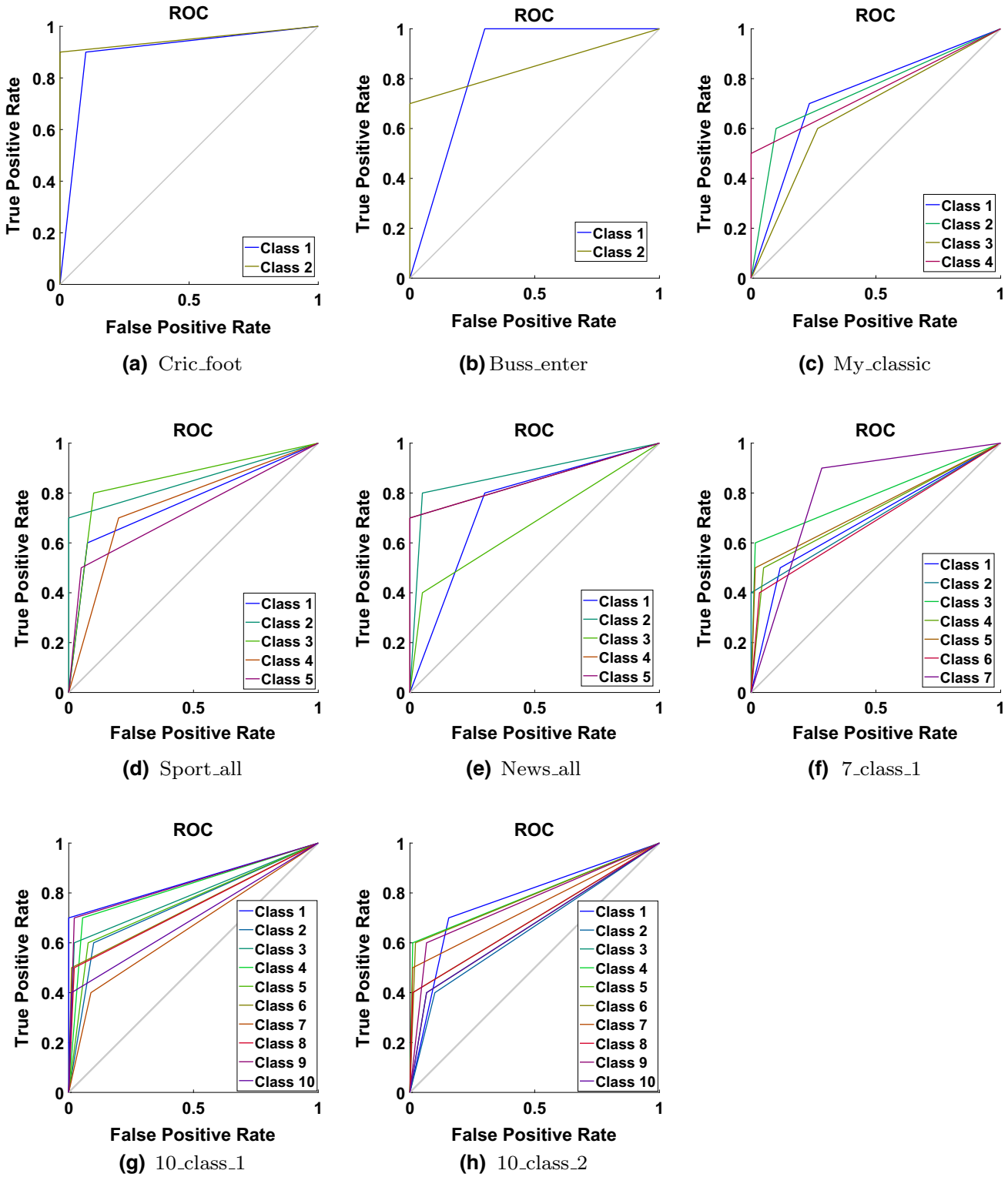
Data set	Method	Cluster Quality Measurement			Classification Accuracy			
		Entropy	Perplexity	Purity	Precision	Recall	F1-score	Accuracy (%)
Cric_foot	Logistic LDA	0.0067	1.0040	0.9660	0.7600	0.5500	0.4357	55
	Discriminative method	0.1142	1.0823	0.9000	0.9000	0.9000	0.9000	90
	Proposed method	0.0448	1.0315	0.9473	0.9500	0.9500	0.94	95
Buss_ent	Logistic LDA	0.0073	1.0050	0.5000	0.7700	0.6000	0.5200	60
	Discriminative method	0.0771	1.0548	0.8670	0.9594	0.9500	0.9498	95
	Proposed method	0.0535	1.0377	0.9842	0.8800	0.8500	0.8400	85
My_classic	Logistic LDA	0.0374	1.0262	0.6956	0.5600	0.3500	0.2695	35
	Discriminative method	0.0132	1.0091	0.9000	0.5714	0.3750	0.3098	37
	Proposed method	0.2732	1.2084	0.9785	0.6400	0.5700	0.5600	57
Sport_all	Logistic LDA	0.0087	1.0060	0.4400	0.4486	0.3200	0.2462	32
	Discriminative method	0.1220	1.0882	0.8260	o.6435	0.6399	0.5910	63
	Proposed method	0.1074	1.0772	0.9690	0.6800	0.6000	0.6000	60
News_all	Logistic LDA	0.0011	1.0007	0.5733	0.5657	0.4200	0.3858	42
	Discriminative method	0.0125	1.0087	0.7333	0.4816	0.5200	0.4882	52
	Proposed method	0.0513	1.0360	0.9906	0.7400	0.6200	0.6200	62
7_class_1	Logistic LDA	0.0140	1.0101	0.4854	0.4472	0.3000	0.2531	30
	Discriminative method	0.0538	1.0379	0.8190	0.6172	0.6142	0.5885	61
	Proposed method	0.0346	1.0242	0.9802	0.6300	0.5100	0.5100	51.42
10_class_1	Logistic LDA	0.0030	1.0020	0.4836	0.4117	0.2500	0.2290	25
	Discriminative method	0.0514	1.0362	0.8200	0.6290	0.5400	0.5051	54
	Proposed method	0.0568	1.0401	0.9930	0.5300	0.4900	0.4800	49.32
10_class_2	Logistic LDA	0.0247	1.0174	0.4537	0.4394	0.2500	0.1804	25
	Discriminative method	0.0638	1.0452	0.7933	0.7981	0.6200	0.6105	62
	Proposed method	0.1213	1.0877	0.9693	0.5000	0.4800	0.4500	48

which is more than around 60%. It is depicted in Fig. 8d that in a five-class data set sport_all, samples of class 3 are correctly predicted with around 80% accuracy. Samples of class 1 and class 2 show relatively low accuracy, around 50%. In News_all data (Fig. 8e), all the class samples are correctly classified with 70% prediction accuracy, except class 3. For 7_class_1 data, class 7 achieves 90% accuracy whereas class 2 and class 6 have comparatively lesser accuracy than other classes. Similarly, it is clear from 10_class_1 and 10_class_2 data sets (Fig. 8g, h)that with a higher number of classes, the average accuracy is better to compare to less number of classes. We, therefore, infer that on average the proposed method gives a statistically significant performance incorrectly predicting the class labels for multi-class data sets. The terms of the large corpus are well distributed among the class labels by the proposed method.

6.3 Topic coherence measurement

Topic coherence [55, 61] is the measurement of degree of semantic similarity between words in a topic. This measurement can be done in two different ways namely, Intrinsic: which do not use any external source or task from the data set, and Extrinsic: which uses external statistics to evaluate the topics.

We have calculated the UCI topic coherence measurement scores for all the data sets to measure the coherence of the topics, generated by the proposed method both in intrinsic and extrinsic ways. In the extrinsic approach, the "Wikipedia data set" and the "20 news group" data sets are used as the external data sets to calculate the extrinsic UCI score. The average coherence score of all the topics in the data set has been calculated and shown in table 9. It has been observed that the coherence score for all the data sets is close to or greater than one which is compatible with other methods [46]. Therefore, the words in a topic are semantically similar and topic distributions are semantically coherent.



(a) Cric_foot **(b)** Buss_enter **(c)** My_classic

(d) Sport_all **(e)** News_all **(f)** 7_class_1

(g) 10_class_1 **(h)** 10_class_2

Fig. 8 ROC curve for different data sets using proposed method

Table 9 Coherence score of different data sets

Data set	UCI Score for proposed method		
	Extrinsic score: Wikipedia as external data	Extrinsic score: 20 newsgroup as external data	Intrinsic score
Cric_foot	1.009	1.185	0.986
Buss_ent	0.728	0.868	1.006
My_classic	1.117	1.291	1.524
Sport_all	0.668	0.810	0.855
News_all	0.668	0.810	0.883
7_class_1	0.772	0.915	1.022
10_class_1	0.882	1.037	1.180
10_class_2	0.572	0.700	0.730

7 Conclusion

The granularity concept used in layered architecture is highly effective to extract the contexts of the training documents. Unlike the deep architecture where the number of nodes and layers is determined heuristically, our method is free from parameterizing effect, and architecture is determined automatically. The information has been transferred hierarchically to combine the GoWs therefore, computationally efficient. Ambiguity in identifying the topics has been dealt with using graph-based modeling, with its ability of interpretation and scaling.

The performance of the proposed method is analyzed in terms of cluster quality and classification accuracy. The proposed unsupervised method performs better than state-of-the-art unsupervised and semi-supervised methods and gives comparable results with a supervised approach. It has been observed that the removal of ambiguity results in better performance in terms of cluster quality and classification accuracy. So in the paper, we justify the application of the graph-based approach on a coarse group of words to achieve better performance than applying GBWG algorithm only.

Compliance with ethical standards

Conflict of 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.

References

1. Almeida H, Guedes D, Meira W, Zaki MJ (2011) Is there a best quality metric for graph clusters? In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 44–59
2. Bafna P, Shirwaikar S, Pramod D (2019) Task recommender system using semantic clustering to identify the right personnel. VINE J Inf Knowl Manag Syst 2:181–199
3. Blagojević M, Micić Ž (2013) A web-based intelligent report e-learning system using data mining techniques. Comput Electr Eng 39(2):465–474
4. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
5. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008
6. Cai D, He X, Han J (2007) SRDA: an efficient algorithm for large-scale discriminant analysis. IEEE Trans Knowl Data Eng 20(1):1–12
7. Chen S-Y, Hung Y-C, Hung Y-H, Chien-Hsun W (2016) Application of a recurrent wavelet fuzzy-neural network in the positioning control of a magnetic-bearing mechanism. Comput Electr Eng 54:147–158
8. classic4 dataset. http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/
9. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391–407
10. Dieng AB, Wang C, Gao J, Paisley JW (2016) Topicrnn: a recurrent neural network with long-range semantic dependency. CoRR. arXiv:1611.01702
11. Dörpinghaus J, Schaaf S, Jacobs M (2018) Soft document clustering using a novel graph covering approach. BioData Min 11(1):1–20
12. Duan T, Lou Q, Srihari SN, Xie X (2019) Sequential embedding induced text clustering, a non-parametric bayesian approach. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 68–80
13. Duan T, Pinto JP, Xie X (2019) Parallel clustering of single cell transcriptomic data with split-merge sampling on Dirichlet process mixtures. Bioinformatics 35(6):953–961
14. Egghe L (2008) The measures precision, recall, fallout and miss as a function of the number of retrieved documents and their mutual interrelations. Inf Process Manag 44(2):856–876
15. Evaluation of clustering (2017). https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html
16. Fang YC, Parthasarathy S, Schwartz F (2001) Using clustering to boost text classification. In: ICDM workshop on text mining (TextDM'01). Citeseer
17. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874
18. Fei J, Rui T, Song X, Zhou Y, Zhang S (2018) More discriminative convolutional neural network with inter-class constraint for classification. Comput Electr Eng 68:484–489
19. Feldman R, Sanger J (2006) Text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, New York
20. Fernández J, Antón Vargas JA, Villuendas-Rey Y, Cabrera-Venegas JF, Chávez Y, Argüelles-Cruz AJ (2016) Clustering techniques for document classification. Res Comput Sci 118:115–125
21. Gallagher RJ, Reing K, Kale D, Steeg GV (2017) Anchored correlation explanation: Topic modeling with minimal domain knowledge. Trans Assoc Comput Linguist 5:529–542
22. Gomez JC, Moens M-F (2012) PCA document reconstruction for email classification. Comput Stat Data Anal 56(3):741–751
23. Greene D, Cunningham P (2006) Practical solutions to the problem of diagonal dominance in kernel document clustering.

In: Proceedings of 23rd international conference on machine learning (ICML'06). ACM Press, pp 377–384
24. Hingmire S, Chougule S, Palshikar GK, Chakraborti S (2013) Document classification by topic labeling. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, pp 877–880
25. Hirsch L, Di Nuovo A (2017) Document clustering with evolved search queries. In: 2017 IEEE congress on evolutionary computation (CEC). IEEE, pp 1239–1246
26. Huang R, Guan Yu, Wang Z, Zhang J, Shi L (2012) Dirichlet process mixture model for document clustering with feature partition. IEEE Trans Knowl Data Eng 25(8):1748–1759
27. Indurkhya N, Damerau FJ (2010) Handbook of natural language processing. Chapman and Hall/CRC, Boca Raton
28. Jagarlamudi J, Daumé III H, Udupa R (2012) Incorporating lexical priors into topic models. In: Proceedings of the 13th conference of the European chapter of the association for computational linguistics, EACL '12, pp 204–213, Stroudsburg, PA, USA. Association for Computational Linguistics
29. Jain VK, Kumar S, Fernandes SL (2017) Extraction of emotions from multilingual text using intelligent text processing and computational linguistics. J Comput Sci 21:316–326
30. Jan B, Farman H, Khan M, Imran M, Islam I, Ahmad A, Ali S, Jeon G (2017) Deep learning in big data analytics: a comparative study. Comput Electr Eng 12
31. Jelodar H, Wang Y, Yuan C, Feng X, Jiang X, Li Y, Zhao L (2019) Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimed Tools Appl 78(11):15169–15211
32. Karaa WBA, Ashour AS, Sassi DB, Roy P, Kausar N, Dey N (2016) Medline text mining: an enhancement genetic algorithm based approach for document clustering. In Applications of intelligent optimization in biology and medicine. Springer, pp 267–287
33. Karypis MSG, Kumar V, Steinbach M (2000) A comparison of document clustering techniques. In: KDD workshop on text mining
34. Kim S-W, Gil J-M (2019) Research paper classification systems based on TF-IDF and LDA schemes. Hum Centric Comput Inf Sci 9(1):30
35. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1746–1751. Association for Computational Linguistics
36. Kong J, Scott A, Goerg GM (2016) Improving semantic topic clustering for search queries with word co-occurrence and bigraph co-clustering. Google Inc, Mountain View
37. Korshunova I, Xiong H, Fedoryszak M, Theis L (2019) Discriminative topic modeling with logistic LDA. In: Advances in neural information processing systems, pp 6770–6780
38. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence
39. Liu L, Liu K, Cong Z, Zhao J, Ji Y, He J (2018) Long length document classification by local convolutional feature aggregation. Algorithms 11(8):109
40. Liu Y, Niculescu-Mizil A, Gryc W (2009) Topic-link LDA: joint models of topic and author community. In: Proceedings of the 26th annual international conference on machine learning, ICML '09. ACM, New York, NY, USA, pp 665–672
41. Madsen RE, Kauchak D, Elkan C (2005) Modeling word burstiness using the Dirichlet distribution. In: Proceedings of the 22nd international conference on machine learning, pp 545–552
42. Meng Y, Huang J, Wang G, Wang Z, Zhang C, Zhang Y, Han J (2020) Discriminative topic mining via category-name guided text embedding. In: Proceedings of the web conference 2020, pp 2121–2132
43. Meng Y, Zhang Y, Huang J, Zhang Y, Zhang C, Han J (2020) Hierarchical topic mining via joint spherical tree and text embedding. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1908–1917
44. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. J Big Data 2(1):1
45. Neal RM (2000) Markov chain sampling methods for Dirichlet process mixture models. J Comput Graph Stat 9(2):249–265
46. Pasquali AR (2016) Automatic coherence evaluation applied to topic models
47. Pavlopoulos GA, Promponas VJ, Ouzounis CA, Iliopoulos I (2014) Biological information extraction and co-occurrence analysis. In: Biomedical literature mining, pp 77–92. Springer
48. Petz G, Karpowicz M, Fürschuß H, Auinger A, Stříteský V, Holzinger A (2013) Opinion mining on the web 2.0—characteristics of user generated content and their impacts. In: Holzinger A, Pasi G (eds) Human-computer interaction and knowledge discovery in complex, unstructured, big data. Springer, Berlin, pp 35–46
49. Popel M, Mareček D (2010) Perplexity of n-gram and dependency language models. In: Sojka P, Horák A, Kopeček I, Pala K (eds) Text, speech and dialogue. Springer, Berlin, pp 173–180
50. Porteous I, Newman D, Ihler A, Asuncion A, Smyth P, Welling M (2008) Fast collapsed gibbs sampling for latent dirichlet allocation. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '08. ACM, New York, USA, pp 569–577
51. Power R, Chen J, Karthik T, Subramanian L (2010) Document classification for focused topics. In: 2010 AAAI spring symposium series
52. Ramage D, Hall D, Nallapati R, Manning CD (2009) Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 conference on empirical methods in natural language processing: volume 1, EMNLP '09. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 248–256
53. Rangrej A, Kulkarni S, Tendulkar AV (2011) Comparative study of clustering techniques for short text documents. In: Proceedings of the 20th international conference companion on World wide web, pp 111–112
54. Rapečka A, Dzemyda G (2015) A new recommendation model for the user clustering-based recommendation system. Inf Technol Control 44(1):54–63
55. Röder M, Both A, Hinneburg A (2015) Exploring the space of topic coherence measures. In: Proceedings of the eighth ACM international conference on Web search and data mining, pp 399–408
56. Schaeffer SE (2007) Graph clustering. Comput Sci Rev 1(1):27–64
57. Siivola V, Pellom BL (2005) Growing an n-gram language model. In: Proceedings of 9th European conference on speech communication and technology, pp 1309–1312
58. Solka JL et al (2008) Text data mining: theory and methods. Stat Surv 2:94–112
59. Sontag D, Roy D (2011) Complexity of inference in latent dirichlet allocation. In: Advances in neural information processing systems, pp 1008–1016
60. Stanchev L (2016) Semantic document clustering using a similarity graph. In: 2016 IEEE tenth international conference on semantic computing (ICSC). IEEE, pp 1–8
61. Stevens K, Kegelmeyer P, Andrzejewski D, Buttler D (2012) Exploring topic coherence over many models and many topics.

In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp 952–961
62. Sun X (2014) Textual document clustering using topic models. In: 2014 10th International conference on semantics, knowledge and grids. IEEE, pp 1–4
63. Suo Q, Ma F, Canino G, Gao J, Zhang A, Veltri P, Agostino G (2017) A multi-task framework for monitoring health conditions via attention-based recurrent neural networks. In: AMIA annual symposium proceedings, vol 2017, p 1665. American Medical Informatics Association
64. Tang P, Wang H (2017) Richer feature for image classification with super and sub kernels based on deep convolutional neural network. Comput Electr Eng 62:499–510
65. Theodosiou T, Darzentas N, Angelis L, Ouzounis CA (2008) Pured-MCL: a graph-based pubmed document clustering methodology. Bioinformatics 24(17):1935–1941
66. Tian F, Gao B, He D, Liu T-Y (2016) Sentence level recurrent topic model: letting topics speak for themselves. arXiv preprint arXiv:1604.02038
67. Tong Z, Zhang H (2016) A text mining research based on LDA topic modelling. In: Proceedings of the sixth international conference on computer science, engineering and information technology (CCSEIT), pp 21–22
68. Teh YW, Jordan M, Beal MJ, Blei DM (2006) Hierarchical dirichlet processes. J Am Stat Assoc 101:1566–1581
69. Wilcoxon F, Katti SK, Wilcox RA (1970) Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. Sel Tables Math Stat 1:171–259
70. Wu HC, Luk RWP, Wong KF, Kwok KL (2008) Interpreting TF-IDF term weights as making relevance decisions. ACM Trans Inf Syst 26(3):13:1–13:37
71. Xie P, Xing EP (2013) Integrating document clustering and topic modeling. arXiv preprint arXiv:1309.6874
72. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489
73. Yin J, Wang J (2016) A model-based approach for text clustering with outlier detection. In: 2016 IEEE 32nd international conference on data engineering (ICDE). IEEE, pp 625–636
74. Yu G, Huang R, Wang Z (2010) Document clustering via dirichlet process mixture model with feature selection. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 763–772