Newsgroup Topic Extraction using Probabilistic Inverse Cluster Frequency Term-Cluster Weighting and Growing Neural Gas Clustering

Sigit Adinugroho, *Member, IAENG,* Muh Arif Rahman, and Dahnial Syauqy

***Abstract*—Effectiveness of a topic extraction method is depends on the capability to extract information from a large amount of data. This paper proposes a new solution for selecting a set of topics from text documents as well as a new approach to set the weight of a term in clusters of documents. The process of revealing topics from multiple documents starts with a preprocessing step, which aims to omit the unnecessary portion of the text. After that, the weights of terms in documents are calculated based on the term frequency-inverse document frequency method. Then, feature transformation based on singular value decomposition is employed to build weight for clustering. The clustering process is conducted using the Growing Neural Gas method. Finally, to determine the weights of terms in clusters as a way of selecting topics from clusters, the proposed probabilistic inverse cluster frequency term-cluster method is applied. Experiments show that the framework attains satisfactory results indicated by the average accuracy of 0.8606, 0.7406, 0.4039, and 0.6647 for topics obtained from Binary2, Multi5, Multi7, and Multi10 categories of 20Newsgroup dataset.**

***Index Terms*—topic extraction, growing neural gas clustering, probabilistic inverse cluster frequency term-cluster weighting, feature transformation.**

I. Introduction

WITH the massive use of the Internet and web technology, more and more people start to publish their content online, often in the form of text in emails, instant messages, bulletin board systems, forums, and weblogs [1], [2]. The huge amount of data makes it difficult to analyze what people write online. On the other hand, grasping the topic of contents shared on the Internet might be essential for security management since some people share illegal content or unhealthy information [3]. Therefore, there is a need for topic extraction from text documents. Often, the topic extraction approach involves clustering procedures for collecting documents with similar topics [4].

There are various types of clustering schemes available for arranging data into a set of groups. Based on the way the algorithm groups the data, there are two main categories of the clustering algorithm: hierarchical and non-hierarchical methods. The non-hierarchical algorithm works by splitting data into a set of predefined number of clusters. This particular type of clustering algorithm can be classified into two more specific categories, namely, hard and fuzzy clustering. The former partitions data in such a way that each data point belongs to one cluster only, while the latter may assign more than one cluster membership to a data point with various degrees of memberships. All types of non-hierarchical clustering algorithms require the number of clusters to be defined in advance. This parameter largely influenced the performance of algorithms since the centroids may be placed correctly in data space if only the number of clusters is chosen correctly. On the other hand, the clustering result may be hard to analyze if the selected number of clusters does not represent the actual number of clusters in the data [5]. There have been several attempts to predict the optimal number of clusters; however, results show that the number is hard to estimate [6], [7], [8].

The hierarchical method offers more flexibility in producing clusters by omitting the need to define the number of clusters in the earliest stage of the clustering process. The method forms clusters by sequentially merging or splitting data or a cluster of data from the previous iteration, resulting in a tree-like structure called dendrogram. The splitting or merging step means that the formation of a branch in the upcoming iteration is purely based on the structure from the previous stage, leaving a chance for a data point to participate in forming a cluster in the next iteration. This characteristic is the main disadvantage of hierarchical clustering [9].

The growing neural gas (GNG) algorithm shares some characteristics of partitioning and non-partitioning clustering methods. Similar to the partitioning method, GNG creates a number of groups according to specific criteria. Unlike partitioning clustering, such as K-Means, GNG automatically reveals the number of clusters. GNG is also somewhat similar to hierarchical clustering since the method develops a structure similar to a dendrogram from prototypes or nodes. The growth of the structure can be controlled to reveal the actual cluster. Also, the possibility to add and remove a prototype at any stage means that the nodes' structure does not depend solely on the structure from the previous iteration [9]. It also revealed that GNG converges fast to a low distortion error and has a lower distortion error compared to K-Means, maximum entropy clustering, and self-organizing feature map [5].

Text documents are commonly represented as a vector-space model of term weighting for computation purposes. Measures based on word co-occurrence and a bag of words (BOW) are regularly involved in long texts such as documents and emails [10], [11]. Among all approaches, TF-IDF

Manuscript received February 27, 2020; revised September 11, 2020.

This work was funded by the Faculty of Computer Science, Brawijaya University, under the Internal Research Grant.

S. Adinugroho is with the Department of Informatics, Faculty of Computer Science, Brawijaya University, Malang, Indonesia (corresponding author email:sigit.adinu@ub.ac.id)

M. A. Rahman is with the Department of Informatics, Faculty of Computer Science, Brawijaya University, Malang, Indonesia (email:m_arif@ub.ac.id)

D. Syauqy is with the Department of Computer Engineering, Faculty of Computer Science, Brawijaya University, Malang, Indonesia (email:dahnial87@ub.ac.id)

is the most common approach for building such a model, and its effectivity was already confirmed. In addition to the vanilla version, various improvements were made to suit a particular type of data [12]. However, processing documents based on TF-IDF term weighting are prone to be problematic. One of the problems lies in the nature of term frequency calculation that keeps an account of the frequency of all words in documents, resulting in a sparse matrix [13] of a high dimensional vector-space model. Since the weighting scheme takes account of entire words in the documents, the features show a high degree of redundancy and irrelevance [11].

Building an index for a large number of documents creates a large matrix since the rows of the matrix correspond to the unique terms in all documents as features. It is a standard theory that too many features inhibit the clustering algorithm from generating good clusters since the features provide low discriminant power to differentiate a document from others. One successful strategy for dimension reduction is to involve singular value decomposition (SVD) [14], [15]. This effort has been proved to be effective in clustering documents. A work by Wei et al. [16] shows that dimension reduction using SVD increases both clustering precision and recall compared to TF-IDF weighting. It is also shown that SVD for dimension reduction is able to reduce the effect of polysemy [17].

This study develops a framework that extracts topics from a set of text documents to solve the clustering and term-weighting problems. The framework is based on clustering with a GNG and term selection based on a term-cluster weighting. The involvement of the GNG algorithm is intended to reveal the actual cluster structure of data, especially the number of clusters. On the other hand, probabilistic inverse cluster frequency and term-cluster weight are implemented to replace topic selection based on term-cluster weight. This study also identifies a proper preprocessing technique as well as selecting the right features for topic extraction.

The main contributions of the paper are (1) constructing a framework for topic extraction based on GNG clustering and term-cluster weighting, (2) developing a new scheme for term-cluster weighting, and (3) verifying the effectiveness of the proposed topic extraction framework by doing experiments.

The paper is organized as follows. Section II provides a detailed description of the proposed framework. Experimental settings are explained in Section III, while Section IV presents and discusses the result of conducted tests. Finally, the paper concludes with Section V.

II. Topic Extraction Based on Clustered Term Weight

The proposed framework for topic extraction consists of five main steps: preprocessing, term-document weighting, feature transformation, clustering, and topic extraction. The overall process is depicted in Figure 1. The explanation of each process is provided in the following sections.

A. Preprocessing

Datasets extracted from real-world data are commonly cluttered with irrelevant information. In the field of text processing, it is highly recommended to conduct preprocessing and normalization after reading the text. Specific to newsgroups data, it is necessary to strip all headers except the subject field. Non-alphanumeric characters are deleted from the text, and finally, normalization occurs by converting all cases to lowercase.

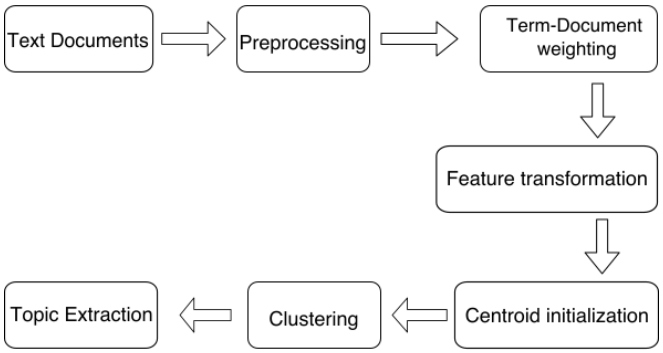

Fig. 1. Flowchart of the proposed framework

The first step to build the term frequency index is extracting all tokens from documents. To achieve that purpose, a set of regular expressions (regex) that recognizes punctuation marks and blank spaces is employed to generate terms since punctuation marks and blank spaces are good separators for words in a sentence. Another form of noise regularly appears in text documents is common words (stop words). In English, words such as of, the, that, is, etc. are considered stop words because they provide a little or no meaning to a sentence. A standard approach to eliminate stop words is to adopt a database of stop words called stoplist and delete all words which exist in the stoplist. This research adapts the stoplist from NLTK (http://www.nltk.org/) library consisting of 179 words. Moreover, the dataset includes a common word ”shall” that presents in 221 documents. Since the word ”shall” is not included in NLTK’s stoplist, it was manually added to the stoplist.

Identical words from a document may appear in different forms due to the use of inflection. It is favorable to use the root word as a standard representation of an inflected word. Two approaches exist to separate affixes from their main word: stemming and lemmatization. Stemming chops off affixes from a word based on some rules, while lemmatization performs morphological analysis using a dictionary. Previous study [18] examines that both methods raise the performance of a clustering method. They also limit the number of words involved in processing. A stemmer performs well for clustering and removes more words, but it only produces stems instead of root words. A stem may not be a valid word, so it is not appropriate for a topic extraction problem since a topic must be a valid word. Therefore, the lemmatization process is more suitable for this case. Our work involves the built-in morphy function in [19].

B. Term-document weighting

An excellent way to eliminate noise words that appear frequently and less meaningful in documents is through stop word removal. On the other hand, less frequent words cannot reveal the meaning of documents either. This type of noise is eliminated by deleting all terms that appear in less than

1% of all documents. [4].

The first step toward building a set of clusters from documents is assigning weight to terms in order to measure their importance. There are various ways to do so, but the standard way is to involve term frequency-inverse document frequency (TF-IDF) [20].In the TF-IDF scheme, the importance of a term in a document is linear with the frequency of occurrence of the term in the document ($TF_{t,d}$); at the same time, penalty is given according to the total number of documents in which the term appears (IDF_t). Mathematically, IDF of term t is calculated as follows:

$$IDF_t = log_2\left(\frac{N}{DF_t}\right) \tag{1}$$

while the TF-IDF weight of term t in document d is

$$TF - IDF_{t,d} = TF_{t,d} \times IDF_t \tag{2}$$

where N is the number of documents in the corpus, DF_t is the number of documents that contain term t, and $TF_{t,d}$ is the frequency of occurrence of term t in document d.

Based on the Equation 2, TF-IDF reaches zero if a term appears in all documents in the corpus. A low value is achieved if a term rarely appears in a document and appears in many documents. However, TF-IDF has high value for a term that appears many times in a document but is rarely found in other documents.

The weights of terms in documents are arranged in a data structure called document-term matrix (DTM) to ease the clustering process. This structure is a matrix composed of terms arranged into rows, while columns are the list of documents in the corpus. A cell of the matrix holds the TF-IDF of a term specified in the corresponding row and column [4].

C. Feature transformation

A naive implementation of a TF-IDF for building features used during the clustering process tends to form a sparse matrix with a high degree of noise as a result of polysemy and synonymity in text documents [21]. A feature transformation based on SVD reconstructs the sparse DTM matrix into a dense one with significantly fewer features, and provides more discriminant power [14]. SVD decomposes a DTM into three independent components, called U, S, and V. Term vectors from a DTM are transformed into left singular vectors (U), while document vectors are mapped into a right singular vector (V). The complete transformation is written in (3).

$$A = USV \tag{3}$$

To reduce the dimension of features U, S, and V are reduced by using a rank k where $k \leq rank(A)$ so that the A matrix is approximated by the trimmed version of U, S, and V and expressed as follows:

$$A \approx U_k S_k V_k^T \tag{4}$$

where U_k is the $d \times k$ (d denotes the number of documents in the corpus) matrix of left singular vectors, S_k is the $k \times k$ diagonal matrix of k largest singular values ordered in a non-decreasing form, while V_k is the $w \times k$ (w denotes the number of terms) matrix of right singular vectors. The reduced form of A used for clustering purposes is expressed in (5)

$$A_k = S_k V_k^T \tag{5}$$

D. GNG clustering

The GNG developed by Fritzke [22] is a topology learning algorithm that aims to uncover the topological structure of a high-dimensional data distribution. It combines the Growing Cell Structures [23] for node addition and the Competitive Hebbian Learning (CHL) method [24] to generate network topology incrementally. The topology is represented as a self-organizing neural network consisting of a set of units or prototypes (A) and a set of edges (N) connecting pairs of units. Each unit $c \in A$ has a weight ($w_c \in \mathbb{R}^n$) that represents the position of the unit in data space.

Initial topology is composed of two random units, a and b, and their corresponding weight, w_a and w_b. Those nodes are connected with a zero-age edge, and their error values are set to 0. For each data point x, the nearest node s and the second one t are located. Then, the ages of all edges emitted from s are raised by one. Also, the error of the winning node is increased by the squared distance between w_s and x as written in (6):

$$error_s = error_s + \|w_s - x\|^2 \tag{6}$$

The topological update to the structure happens only at the winning node and all nodes connected to it. Their weights are updated toward x by the factor of e_w and e_n, where $e_w, e_n \in [0, 1]$.

$$\begin{aligned} w_s &= w_s + e_w(w_s - x) \\ w_n &= w_n + e_n(w_n - x), \forall n \in neighbor(s) \end{aligned} \tag{7}$$

If s and t are neighboring nodes (meaning that they are connected by an edge), set the age of the connecting edge to 0; otherwise, create an edge to connect them.

Topology pruning occurs in each iteration if an edge has the age larger than a_{max}. If the pruning produces lone nodes (nodes without any neighbor), the nodes are removed. Nevertheless, topology growth only takes place if the current iteration is a multiple of λ. In this case, a new node is inserted by the following procedures:

- Locate the node l, which has the largest error.
- Select all nodes connected to l, and then find the node v with the highest error.
- Put a new node r between l and v. The weight of r (w_r) is calculated by using the following equation:

$$w_r = \frac{(w_l + w_v)}{2} \tag{8}$$

- Connect l and r, v and r, and then remove the connection between l and v.
- Set error value for l, v, and r according to (9):

$$\begin{aligned} e_l &= \alpha e_l \\ e_v &= \alpha e_v \\ e_r &= e_l \end{aligned} \tag{9}$$

At the end of each iteration, error values of all nodes are decreased by multiplying the old values with a constant d.

Finally, stopping conditions are checked if a further iteration is needed. Some possible terminating conditions are the maximum number of nodes or performance evaluations based on several measures.

E. Topic extraction using probabilistic inverse cluster frequency term-cluster weighting

Topics from a cluster can be represented by the center of the cluster based on a weighting measure. One of the measures is term-cluster weighting, which consists of two measures: term frequency and inverse cluster frequency (ICF)[25]. Similar to IDF, ICF of term j is derived from the number of clusters where term j appears (c_j) and the number of clusters (C) as follows:

$$ICF_j = log\frac{C}{cf_j} \tag{10}$$

The weight of term j in cluster i is defined as

$$w_{ij} = tf_{ij}ICF_j \tag{11}$$

where tf_{ij} is the frequency of term j in cluster i.

Similar to the IDF formula, ICF is built under the assumption that the number of clusters in which the term appears (cf_j) is very small compared to the number of clusters. This may not hold true in the dataset used in various research, and there is a possibility that a term appears in more than 1 cluster. Therefore, the probabilistic inverse cluster frequency was coined to substitute ICF. The proposed topic weighting is written as follows:

$$w_{ij} = tf_{ij}log\frac{C - cf_j}{cf_j} \tag{12}$$

It is worth mentioning that GNG clustering produces final centroids that are not necessarily an existing data point in the dataset. Therefore, terms for topic candidates are selected from data points with minimum distance from the actual centroids. A term provides more strength to represent a topic if it is available in a cluster with high frequency but infrequently appears in other clusters. A topic of cluster T_i is represented by an array of terms with maximum weight in the cluster:

$$T_i = (term_{i1}, term_{i2}, term_{i3} \dots term_{in}) \tag{13}$$

F. Performance evaluation

Several validity measures for clustering results and topic extraction are involved to validate the performance of the proposed methods. Those measures are the silhouette coefficient and topic accuracy.

The silhouette coefficient [26] measures how similar data are arranged in a cluster (cohesion) and how separated the clusters are (separation). The silhouette coefficient is calculated for each object in the overall dataset based on distance metrics, such as Euclidean distance or Manhattan distance.

After a result is acquired using a clustering method, let us take an object i assigned in cluster A by the clustering algorithm. Then a_i can be calculated as the average distance of i to other data in cluster A. From the clustering result, C can be defined as clusters other than A and $d(i, C)$ as the average distance from i to each data point in C. After that, b_i is denoted as the minimum $d(i, C)$ for all $C \neq A$. Finally, the silhouette score of i, s_i is computed as follows:

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}} \tag{14}$$

Note that if a singleton cluster (a cluster with only one object) appears, the silhouette coefficient for the object is set to zero. The possible value for a silhouette coefficient ranges from -1 to 1. The overall silhouette coefficient for a clustering result is the mean of silhouette scores of all data.

A topic accuracy metric is an essential tool for determining the quality of the extracted topic [25]. It calculates the mean of n highest weight of terms that represent a topic. The accuracy of term j in cluster i, $A(i, j)$ is calculated in (15):

$$A(i, j) = \frac{w(i, j)}{\sum_{k=i}^{C} w(k, j)} \tag{15}$$

where $w(i, j)$ is the weight of term j in cluster i and C is the number of clusters. The overall accuracy of all topics in all clusters is shown in (16):

$$\text{Overall topic accuracy} = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{n}\sum_{j=i}^{n} A(i, j) \tag{16}$$

III. Experiments

Performance evaluations of the proposed method were examined on a public text dataset called 20Newsgroup. More specifically, the experiments were conducted using a specific version of the dataset called 1828 version by Jason Rennie [1]. The dataset omits duplicates and all headers other than "From" and "Subject" from the original version. A slice of the dataset is selected and divided into four categories [27] as shown in Table I.

The goal of the first experiment was to find the most appropriate SVD rank for each dataset to produce ideal clusters, so silhouette coefficients were employed to measure the quality of cluster results. In each dataset, the rank was set to value ranging from 2 to 500. Then, the silhouette coefficient and the number of clusters formed in each rank were recorded. All GNG parameters were set to fixed values, as shown in Table II.

The second experiment was intended to identify the most suitable GNG parameters for each dataset. This goal was done by measuring the silhouette coefficient for various maximum nodes and maximum ages combination by applying exhaustive search. The maximum number of nodes ranges from 1-100 in one increment, while the maximum age ranges from 1-100 in one increment. e_n and e_w were set to the fixed numbers, as shown in Table II.

A comparison to other clustering algorithms was performed to identify if GNG is an appropriate algorithm to cluster the newsgroup dataset. K-Means and Agglomerative clustering with four linkage methods are examined based on their silhouette coefficients.

The last experiment was aimed to identify the accuracy of extracted terms as well as to compare the proposed term weighting scheme with the existing one. In this scenario, the

[1]http://qwone.com/ jason/20Newsgroups/

TABLE I
THE DISTRIBUTION OF THE DATASET

Category	Newsgroup	Number of documents per topic	Total documents
Binary2	talk.politics.mideast	940	1715
	talk.politics.misc	775	
Multi5	comp.graphics	973	4888
	rec.motorcycles	994	
	rec.sport.baseball	994	
	sci.space	987	
	talk.politics.mideast	940	
Multi7	alt.atheism	799	6612
	comp.sys.mac.hardware	961	
	misc.forsale	972	
	rec.autos	990	
	rec.sport.hockey	999	
	sci.electronics	981	
	talk.politics.guns	910	
Multi10	alt.atheism	799	9580
	comp.sys.mac.hardware	961	
	misc.forsale	972	
	rec.autos	990	
	rec.sport.hockey	999	
	sci.electronics	981	
	talk.politics.guns	910	
	sci.crypt	991	
	sci.med	990	
	sci.space	987	

TABLE II
GNG PARAMETERS FOR RANK TESTING

Parameter	Value
Maximum number of nodes	50
Maximum age	5
e_n	0.0006
e_w	0.05

quality of topics generated by term-cluster weighting and the proposed method were compared based on the optimal parameters found in the previous observation. The metric used for measuring the quality of extracted topics was topic accuracy.

IV. EVALUATION RESULTS

A. Selecting the proper rank

Our goal to assess the most appropriate rank for each category was achieved by running the GNG clustering with a rank value ranging from 2 to 500. Cluster quality was measured as well as the number of clusters only if the number of clusters was more than 1. This approach is intended to avoid the formation of a singleton cluster. Figure 2 shows the silhouette coefficient concerning the rank for each category. It is shown that all categories have a similar trend: the low rank attains the highest coefficient, and as the rank grows, the probability of obtaining a low coefficient increases.

The highest silhouette coefficient for Binary2 is 0.749384 when the rank is set to 2. In this setting, the number of clusters found is 2. According to the definition of the silhouette coefficient, this clustering result is ideal since the coefficient is relatively close to 1, and the number of revealed clusters is the same as the actual number of clusters. As shown in Figure 2, the coefficient plummets as the rank is incremented and reaches a steady condition when the rank is greater than 20. The figure only reveals the first 60 ranks since the values beyond that threshold have a similar coefficient lower than 0.1.

Multi5 dataset shows a similar tendency: it has the highest silhouette coefficient in the first few ranks and significantly decreases after that. The rank with the maximum coefficient is 5. However, the cohesion of data in a cluster and separation between clusters is relatively low as indicated by the coefficient of 0.393371, which is far away from 1. In this rank, the number of clusters found is 3.

The lowest silhouette coefficient in the whole dataset is obtained in Multi7 category. In this category, the highest coefficient is 0.170486, obtained in rank 13. Even though the number indicates cluster cohesion and separation that are far from ideal, the number of extracted clusters is 5, which is quite close to the actual number of clusters.

GNG also succeeds in unveiling the ideal form of a cluster in Multi10 dataset marked by the high silhouette coefficient, which is 0.610698. It is obtained when the rank is 2. Despite the high silhouette coefficient, the number of clusters found is not close to the actual one. At the highest silhouette, the number of clusters found is 2. The reason why the number of clusters found is far lower than the real one is the cluster overlap, as explained in [28].

B. Identifying optimal GNG parameters

In order to select the right GNG parameters, maximum nodes and maximum age, the silhouette coefficient is evaluated for each combination of those two parameters based on the rank chosen in the previous step. To do that, an exhaustive search method is implemented to determine silhouette for every possible permutation of maximum age and nodes.

The maximum silhouette coefficient discovered in Binary2 is 0.781251, with maximum nodes being six and the maximum age of three. In this configuration, two clusters are found that matches the actual number of newsgroup category in Binary2. Variations of maximum age and nodes around the optimal one converge to similar silhouette coefficient while maintaining the same number of clusters found, as shown in Table III. It is also shown in Table IV that the five highest silhouette coefficients require few nodes and age.

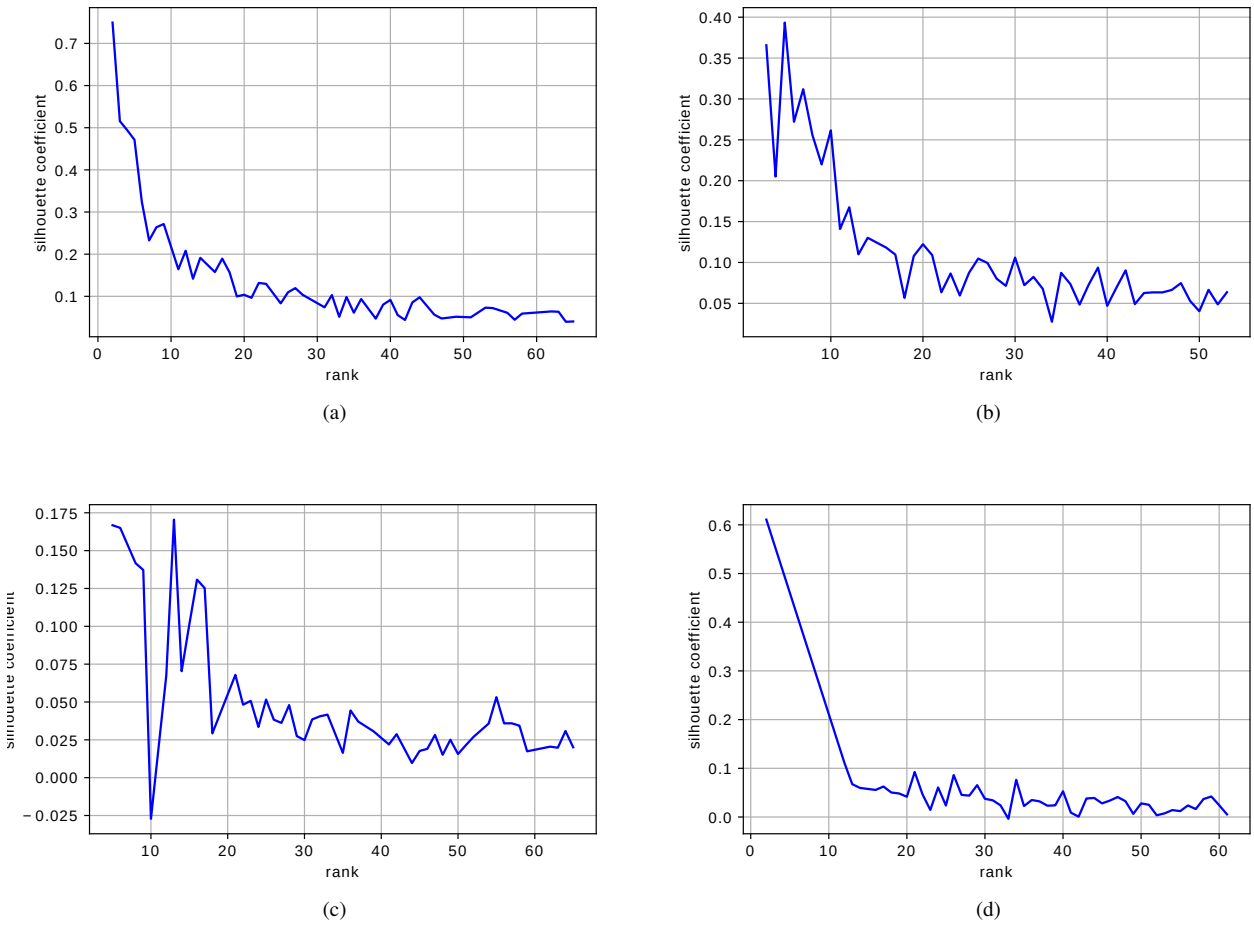

Fig. 2. Silhouette coefficients for first 50 rank with nonzero coefficient in each category: **a** Binary2, **b** Multi5, **c** Multi7, and, **d** Multi10

The reason why only a small number of nodes are needed is that the structure of the Binary2 dataset is not complicated. Thus six nodes are sufficient to form two connected graphs that represent the structure of the clusters. It is interesting to see that clusters formed by GNG have a better structure since its silhouette coefficient is close to one. On the other hand, clusters formed by the actual labels only achieve the silhouette score of 0.07498. The score that is close to zero indicates overlapping clusters since Binary2 comes from sub-categories in Newsgroup dataset, which is talk.politics.

TABLE III
SOME COMBINATIONS OF MAX AGE AND MAX NODES FOR BINARY2. THE OPTIMAL PAIR IS IN BOLD

Max Age	Max Nodes	#cluster	Silhouette coefficient
2	5	2	0.779566
2	6	2	0.77704
2	7	2	0.778495
3	5	2	0.776932
3	**6**	**2**	**0.781251**
3	7	2	0.779876
4	5	2	0.779432
4	6	2	0.768968
4	7	2	0.780975

Multi5 category demands significantly more nodes to achieve the highest silhouette score. This particular dataset requires maximum nodes of 22 and maximum age of 3 to produce 5 clusters and the silhouette score of 0.505109. More nodes are involved in this dataset since more nodes are needed to form more connected graphs that represent cluster formation in the dataset. Compared to the clusters based on actual labels in Multi5, GNG composes better structure since its silhouette coefficient is 58.48% higher that of actual label cluster. Table V provides evidence that all variations of maximum nodes and age in the same number of clusters and similar silhouette coefficients.

TABLE IV
TOP 5 COMBINATIONS OF MAXIMUM AGE AND NODES FOR BINARY2

Max Age	Max Nodes	#cluster	Silhouette coefficient
3	6	2	0.781251
4	7	2	0.780975
5	7	2	0.7809
3	8	2	0.780181
3	7	2	0.779876

Table VI provides unstable results shown by variations of the silhouette coefficient and the number of clusters found in Multi7 dataset. The reason for this phenomenon is cluster overlapping that has been analyzed in the previous research [28]. Another proof of cluster overlap is a low silhouette coefficient of 0.19685 computed from clusters generated by the actual labels. The best parameter in GNG leads to six

TABLE V
TOP 5 COMBINATIONS OF MAXIMUM AGE AND NODES FOR MULTI5

Max Age	Max Nodes	#cluster	Silhouette coefficient
3	22	5	0.505109
3	24	5	0.505109
7	15	5	0.499309
6	17	5	0.498585
6	19	5	0.494129

clusters with the silhouette coefficient of 0.180004. Since some clusters overlap in the dataset, the number of clusters found does not match the actual number of clusters in the dataset.

TABLE VI
TOP 5 COMBINATIONS OF MAXIMUM AGE AND NODES FOR MULTI7

Max Age	Max Nodes	#cluster	Silhouette coefficient
5	52	6	0.180004
5	51	6	0.179091
5	50	5	0.170486
5	49	5	0.16986
3	25	5	0.163494

GNG performs well in Multi10 dataset, as demonstrated in Table VII. The far from zero silhouette coefficient argues that proper formation of clusters is exposed by GNG. However, the number of clusters disclosed by GNG does not fit the actual labels. GNG discovers two clusters, while the actual labels separate the data into ten clusters. By investigating the silhouette coefficient calculated from the labels, we figured out that cluster overlap occurred in the dataset since the coefficient is as low as -0.07404. It is also easy to spot that several topics in Multi10 belong to a similar category, and therefore some topics share common words that in the end creates cluster overlap. In conclusion, it is more favorable to have two clusters, rather than ten.

TABLE VII
TOP 5 COMBINATIONS OF MAXIMUM AGE AND NODES FOR MULTI10

Max Age	Max Nodes	#cluster	Silhouette coefficient
3	11	2	0.672618
3	36	2	0.659165
3	50	2	0.659165
3	46	2	0.659165
3	45	2	0.659165

C. Comparison with other clustering algorithms

In order to evaluate the performance of the GNG algorithm to automatically group newsgroup data, a comparison with other clustering methods is conducted. Two algorithms are involved in the testing: K-Means and Agglomerative Clustering with four linkage criterion, namely ward, complete, average, and single linkage. It is important to note that Agglomerative Clustering generates a tree-like structure that represents a hierarchy of clusters; thus, cluster labels are hard to determine since they depend on the levels at the tree. To deal with that issue, we set the agglomerative clustering method to stop creating the tree structure after finding k clusters, where k is the number of unique labels in each dataset. In this case, agglomerative clustering behaves like a flat clustering.

Table VIII highlights the performance for each clustering algorithm represented by the silhouette coefficient, and the best result is printed in bold. In Binary2 dataset, all algorithms perform as expected on this well-structured data indicated by identical silhouette coefficients. However, there is an exception for Agglomerative clustering with a single linkage that performs poorly because it forms near singleton cluster where only one data is labeled differently from the remaining data.

A similar tendency is shown in Multi5 dataset, where almost all algorithms' performance are alike, except Agglomerative with single linkage. K-Means outperforms other algorithms, but the gap with GNG is unnoticeable by mere -0.447%. The gap between K-Means and Agglomerative methods becomes wider, -4.156%, -10.071%, -3.925% and -118.294 for ward, complete, average, and single linkage, respectively. Similar to the previous dataset, single linkage forms a cluster where almost all data points are the member of the first cluster, and the four remaining clusters only have one member each.

GNG's performance cannot cope with those of K-Means and Agglomerative clustering with ward and average linkage in Multi7 dataset. The overlapping cluster problem makes it difficult for GNG to reveal the actual number of clusters in that dataset.

GNG outruns its rivals in Multi10 dataset with notable disparity. Its silhouette coefficient is 28.155% higher compared to its closest competitor, K-Means. Unlike Multi5, cluster overlapping in Multi10 exposes the actual structure of the clusters as previously discussed in SectionIV-B.

To summarize the performance of clustering algorithms, GNG performs at a satisfactory level for most of the dataset, except Multi7. GNG also excels at revealing the actual cluster structure in Multi10 dataset, where others are failed to show the same capability. Agglomerative clustering with single linkage is the most inferior method since it establishes clusters where almost all data are grouped into a single cluster, and the rest are scattered in the remaining clusters.

D. Accuracy of extracted topics

Based on the ranks found in Section IV-A, and maximum nodes and ages found in Section IV-B, the accuracy of the extracted topic in each category is compared using the proposed probabilistic inverse cluster frequency term-cluster weighting and the original term-cluster weighting. The aim is to understand whether the number of topics affects accuracy. Topic accuracy is then calculated for 1 - 10 topics in each category.

Table IX shows the topic accuracy for Binary2 dataset. For the first two topics, the term-cluster weighting achieves a perfect silhouette score, while the proposed method scores 0.8861 and 0.8776. The lower silhouette scores are caused by high cluster frequency. The proposed method found terms that appear in both clusters, while the legacy method extracted terms that exist exclusively in a single cluster. In contrast, the proposed method has a tendency to find terms that frequently appear in documents of a cluster, whereas

TABLE VIII
CLUSTER EVALUATION FOR VARIOUS CLUSTERING ALGORITHMS

Dataset	GNG	K-Means	Agglomerative clustering			
			Ward	Complete	Average	Single
Binary2	**0.781251**	0.78018	0.780791	0.78018	0.**781251**	0.425999
Multi5	0.505109	**0.507378**	0.486289	0.45628	0.487461	-0.09282
Multi7	0.180004	**0.247191**	0.201012	0.079253	0.194697	-0.12984
Multi10	**0.672618**	0.524424	0.495987	0.500402	0.410048	0.481782

the term-cluster weighting is likely to find a term that less frequently presents in a cluster. For instance, the term with the highest proposed weight in the first cluster, "armenian", is available in both clusters, with the frequency of 2072 in the first cluster and 27 in the second one. For comparison, the term with the highest term-cluster weight is "ohanus" which occurs 138 times in the first cluster but does not exist in the second cluster. As more terms are taken from clusters, it is more difficult to find a term that exclusively belongs to a single cluster. That is why the accuracy of the term-cluster weighting gradually declines. On average, a topic from the proposed weighting shows up at 1.8 clusters and found 1018.65 times in its designated cluster, while a topic from the term-cluster one emerges from 1.45 clusters, each with 1018.65 appearances. The general performance comparison is made by averaging the silhouette coefficient for ten topics gathered by both methods. The proposed method achieved a 31.52% higher silhouette coefficient compared to the term-cluster weighting scheme.

TABLE IX
COMPARISON OF ACCURACY ON BINARY2

Number of terms	Proposed weighting	Term cluster
1	0.8861	1.0000
2	0.8776	1.0000
3	0.8612	0.8333
4	0.8781	0.7500
5	0.8718	0.7000
6	0.8612	0.5833
7	0.8596	0.5000
8	0.8543	0.4375
9	0.8335	0.3889
10	0.8223	0.3500
Average	0.8606	0.6543

In the Multi5, the term-cluster weighting exceeds the performance of the proposed one by 17.59% as highlighted by Table X. It is likely caused by more term frequency from topics gathered by the term-frequency method and higher cluster frequency from topics extracted by the proposed method. Statistically, a topic appears in 3.52 clusters with a term frequency of 688.74 when the proposed method is used. From the term-cluster method, a term has a cluster frequency of 2.82 and a term frequency of 247.28. Those measures are averaged from 10 topics.

The silhouette coefficient from Multi7 dataset is even lower for the proposed method. Table XI concludes that the silhouette of the proposed method is 45.80% lower compared to the term-cluster method. As we discussed before, the proposed method is focused on finding terms with higher term frequency. Since the number of the actual cluster grows, it is more difficult to find higher term frequency. Besides, cluster overlapping adds another complication to find low cluster frequency,i.e, finding a term that rarely exists in multiple clusters.

TABLE X
COMPARISON OF ACCURACY ON MULTI5

Number of terms	Proposed weighting	Term cluster
1	0.9070	0.9731
2	0.8868	0.9209
3	0.8355	0.9296
4	0.7784	0.8889
5	0.7379	0.8854
6	0.7187	0.8566
7	0.6678	0.8400
8	0.6420	0.8244
9	0.6252	0.8002
10	0.6062	0.7889
Average	0.7406	0.8708

TABLE XI
COMPARISON OF ACCURACY ON MULTI7

Number of terms	Proposed weighting	Term cluster
1	0.5086	0.7023
2	0.4534	0.6602
3	0.4415	0.6430
4	0.4169	0.6053
5	0.4219	0.6085
6	0.3964	0.5953
7	0.3663	0.5525
8	0.3568	0.5314
9	0.3412	0.5065
10	0.3360	0.4843
Average	0.4039	0.5889

Similar to Binary2 dataset, the proposed method outperforms the term-cluster method in Multi10 category. In this particular dataset, the average cluster frequency from ten topics obtained by the proposed and term-cluster method is similar, 1.85 and 1.65, respectively. For term frequency, the proposed method achieves far higher value, 2443.35, compared to 155.15 from the term-cluster method. These factors lead to a far higher silhouette coefficient from the proposed method as shown in Table XII.

TABLE XII
COMPARISON OF ACCURACY ON MULTI10

Number of terms	Proposed weighting	Term cluster
1	0.8054	0.5000
2	0.7232	0.5000
3	0.7079	0.5000
4	0.6668	0.5000
5	0.6412	0.4000
6	0.6201	0.3333
7	0.6080	0.2857
8	0.6128	0.2500
9	0.6197	0.2222
10	0.6421	0.2000
Average	0.6647	0.3691

All categories in the accuracy testing show identical behavior. As more keywords are extracted from clusters, the accuracy of the topic worsens. The explanation for this phenomenon is that the extracted topics have more probability of being a member of more than one cluster if more topics are extracted. This is caused by more terms are located on more than one cluster, leading to a drop in accuracy.

V. Conclusion

This paper developed a framework for extracting topics from text data based on GNG clustering and probabilistic inverse cluster frequency term-cluster weighting metric. Experiments on the 20Newsgroup dataset verified the performance of the framework and the proposed term-weighting. First, the approach to involve GNG resulted in ideal document clusters in most dataset categories. Then, the number of optimal ranks for each category was observed to be 2, 5, 13, and 2 for Binary2, Multi5, Multi7, and Multi10, respectively. The most appropriate pair of maximum nodes and ages was 6 and 3 for Binary2, 22 and 3 for Multi5, 52 and 5 for Multi7, and 11 and 3 for Multi10. Finally, the term accuracy was calculated in order to measure the accuracy of the extracted topics. The average accuracy of the framework was 0.8606, 0.7406, 0.4039, and 0.6647 for Binary2, Multi5, Multi7, and Multi10, respectively.

Further research should be directed toward a more comprehensive understanding of topics. In this research, topics are only represented by an array of terms. Understanding a topic would be easier if the terms were linked to form a sentence. One possible way is to use the natural language processing method to grasp the relation between terms and knowledge extraction from the Internet since some terms are popular words still not registered in English dictionaries.

References

[1] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 171–180.

[2] K. Naruse and M. Kubo, "Lognormal distribution of bbs articles and its social and generative mechanism," in *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, Dec 2006, pp. 103–112.

[3] J. Zeng, S. Zhang, C. Wu, and X. Ji, "Modelling topic propagation over the Internet," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 15, no. 1, pp. 83–93, 2009.

[4] A. Karl, J. Wisnowski, and W. H. Rushing, "A practical guide to text mining with topic extraction," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 5, pp. 326–340, Sep. 2015.

[5] A. Qin and P. Suganthan, "Robust growing neural gas algorithm with application in cluster analysis," *Neural Networks*, vol. 17, no. 8, pp. 1135 – 1148, 2004, new Developments in Self-Organizing Systems.

[6] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 727–734.

[7] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, ser. NIPS'03. Cambridge, MA, USA: MIT Press, 2003, pp. 281–288.

[8] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*. Calcutta, India, 1999, pp. 137–143.

[9] M. Daszykowski, B. Walczak, and D. L. Massart, "On the optimal partitioning of data with k-means, growing k-means, neural gas, and growing neural gas," *Journal of chemical information and computer sciences*, vol. 42, no. 6, pp. 1378–1389, 2002.

[10] K. Abdalgader, "Clustering Short Text using a Centroid-Based Lexical Clustering Algorithm." *IAENG International Journal of Computer Science*, vol. 44, no. 4, pp. 523–536, 2017.

[11] J.-M. Yang, Z.-Y. Liu, and Z.-Y. Qu, "Clustering of words based on relative contribution for text categorization," *IAENG International Journal of Computer Science*, vol. 40, no. 3, pp. 207–219, 2013.

[12] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, Jan. 2003.

[13] X. Tian and W. Tong, "An improvement to tf: Term distribution based term weight algorithm," in *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 1, April 2010, pp. 252–255.

[14] R. Albright, "Taming text with the svd," *SAS Institute Inc*, 2004.

[15] B. Gautam and S. Dipesh, "Document Clustering Through Non-Negative Matrix Factorization: A Case Study of Hadoop for Computational Time Reduction of Large Scale Documents," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2010 , IMECS 2010*, vol. 2180, Hong Kong, 17-19 Mar. 2010, pp. 570–575.

[16] C.-P. Wei, C. C. Yang, and C.-M. Lin, "A latent semantic indexing-based approach to multilingual document clustering," *Decision Support Systems*, vol. 45, no. 3, pp. 606 – 620, 2008, special Issue Clusters.

[17] J. R. Bellegarda, J. W. Butzberger, Y.-L. Chow, N. B. Coccaro, and D. Naik, "A novel word clustering algorithm based on latent semantic analysis," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1, May 1996, pp. 172–175 vol. 1.

[18] P. Han, S. Shen, D. Wang, and Y. Liu, "The influence of word normalization in English document clustering," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2, May 2012, pp. 116–120.

[19] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[20] T. D. Nguyen and M.-Y. Kan, "Keyphrase extraction in scientific publications," in *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, D. H.-L. Goh, T. H. Cao, I. T. Sølvberg, and E. Rasmussen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 317–326.

[21] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[22] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in neural information processing systems*, 1995, pp. 625–632.

[23] B. Frizke, "Crowing cell structure-a self organizing network for unsuprevised and supervised learning," *Neural Networks*, vol. 7, pp. 1441–1460, 1994.

[24] T. Martinetz, "Competitive hebbian learning rule forms perfectly topology preserving maps," in *ICANN '93*, S. Gielen and B. Kappen, Eds. London: Springer London, 1993, pp. 427–434.

[25] H. Ayad and M. Kamel, "Topic Discovery from Text Using Aggregation of Different Clustering Methods," in *Advances in Artificial Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 161–175.

[26] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.

[27] L. Muflikhah and B. Baharudin, "Document Clustering Using Concept Space and Cosine Similarity Measurement," in *2009 International Conference on Computer Technology and Development*, vol. 1, Nov. 2009, pp. 58–62.

[28] S. Adinugroho, Y. A. Sari, M. A. Fauzi, and P. P. Adikara, "Optimizing k-means text document clustering using latent semantic indexing and pillar algorithm," in *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, Aug 2017, pp. 81–85.

Sigit Adinugroho (M'08) Sigit Adinugroho got his bachelor's degree in computer science from Brawijaya University, INDONESIA. He earned his master's degree from Uppsala University, SWEDEN, majoring in Computer Science.

After graduating from Uppsala University, he joined the Faculty of Computer Science, Brawijaya University, as a junior lecturer in the Informatics Department. During his career as a lecturer, he has published several works in journals and conference proceeding. His main interests are machine learning, data mining, and computer vision. He also published a book on data mining.

Muh Arif Rahman Muh Arif Rahman got his bachelor's degree from the Mathematics Department, Institut Teknologi Sepuluh Nopember, INDONESIA. After that, he pursued his master's degree at the Faculty of Computer Science, Universitas Indonesia, INDONESIA.

During his career as a senior lecturer at the Faculty of Computer Science, Brawijaya University, he has published several research articles in journals and conference proceedings. His main research areas are machine learning and computer vision.

Dahnial Syauqy Dahnial Syauqy got his bachelor's degree from the Electronics Department, Faculty of Engineering, Brawijaya University, INDONESIA. He earned a double master's degree from Brawijaya University and National Central University, TAIWAN.

His main research interests are control systems, electronics, and biomedical engineering. During his career as a junior lecturer at the Faculty of Computer Science, Brawijaya University, he has published several research articles in journals and conference proceedings.