Application of Community Detection Technique in Text Mining

Shashank Dubey, Abhishek Tiwari and Jitendra Agrawal

1 Introduction

The data mining is a technique that enables us to evaluate or analyze the data automatically. This ability of data mining offers to analyze the significant amount of data in less amount of human effort consumption [1]. In this work, the data mining technique is utilized for text mining. The text mining is a domain where the data mining algorithms and techniques are applied for analyzing the text data [2]. The text mining techniques includes the various phases of data analysis such as pre-processing, feature computation, implementation of data mining algorithm on feature set recovered, and finally the evaluation of performance on the basis of application [3].

In this context, various techniques of data mining techniques are developed that are claimed to provide the accurate and efficient data analysis. But the text data is not completely separable from each other; a partial similarity always exists among various subjects of data or different domains of data. Therefore in order to understand the similarity and the differences among two given text documents, they are needed to be evaluated. In addition to that the text is an unstructured kind of data which is not available in pre-labeled format. Thus, making accurate classification technique for the text is also a complicated task [4].

In order to deal with the considered issues and text mining challenges, a new approach of text mining is proposed in this work. That automatically analyzes the data and discovers the possible similar groups in data. This automatic recovery of

S. Dubey (✉) · A. Tiwari
Information Technology, Mahakal Institute of Technology, Ujjain, India
e-mail: shashankdubey9@gmail.com

A. Tiwari
e-mail: abhi.tiwari23@gmail.com

J. Agrawal
Computer Science and Engineering, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India
e-mail: jitendra@rgtu.net

R. K. Shukla et al. (eds.), *Data, Engineering and Applications*,
https://doi.org/10.1007/978-981-13-6347-4_4

data groups is termed here as the community detection in the text mining. In addition to that to make understandable the similarity in interclusters [5], the visualization technique is used. In addition to that sometimes the two clusters can share some kind of common data, thus it is also considered to evaluate the common amount and instances of data among multiple groups. This section provides the basic overview of the proposed work. The next section describes the complete system modeling and the phases of the system.

2 Proposed Work

This section provides the detailed discussion and understanding about the proposed methodology. The methodology involves the solution development components, their functions, and the evaluation processes.

2.1 System Overview

The proposed work is motivated to perform text mining using the new technique which promises to not only provide accurate cluster analysis but also provide the relativity among the available clusters on to another. Basically, the cluster technique is an unsupervised manner of data mining. The unsupervised learning needs to provide the predefined patterns as input for learn about the patterns. These algorithms are developed in such manner by which the algorithm self-evaluates the data and finds the similarity or differences among them to prepare the similar kind of data objects or instances.

The text mining technique is now in these not only helpful for performing the classical task such as digital document classification and categorization. That is also much effectively used for various other applications such as finding the trending topics in social media, obtaining the positive and negative reviews of company products and services, emotion mining, and similar others [6]. Therefore, the text mining using the unsupervised learning is a subject of interest in this work. The proposed text mining technique is motivated from a community detection approach. Basically, the community detection approach is a problem-solving method which uses the graph theory for finding the similar group of objects in a significant amount of data.

In this work, the key method utilizes the text data as input and analyzes the text using distance finding method. Finally, the relativity among the data is demonstrated using the graph method. The main issue during this process is to regularize the size of text data and the recovery of text features. These features are the properties by which the similarity or difference among two data instances is measured. This section provides the overview of the proposed model, and the next section provides the detailed description of model formulation.

2.2 Methodology

The process used in the proposed methodology for community detection in text data is demonstrated in Fig. 1. In this diagram, the block contains the processes and the edge of blocks represents the flow of processed data from one process to another.

Input text data: That is an initial input to the system. The experimenting user can use any kind of text data using this input provision. But the limitation is only that the input data is needed to be combined in a single text document or a single directory. The system read this input document or set of document to utilize in further process. In order to collect the various domains or subjective data, here the social media text is used as input for experimentation purpose—more specifically the Twitter conversion (post) is used in a single document.

Data pre-processing: The pre-processing of data is one of the essential phases of data mining. The pre-processing technique is used to improve the quality of data. Therefore, the noisy attributes and instances are identified and removed from the input datasets. In the context of text document, the noise is recognized as the unused data, words, symbols which appeared in document or text blocks and not having much importance in document orientation discovery. Therefore, the proposed system includes two phases of data pre-processing.

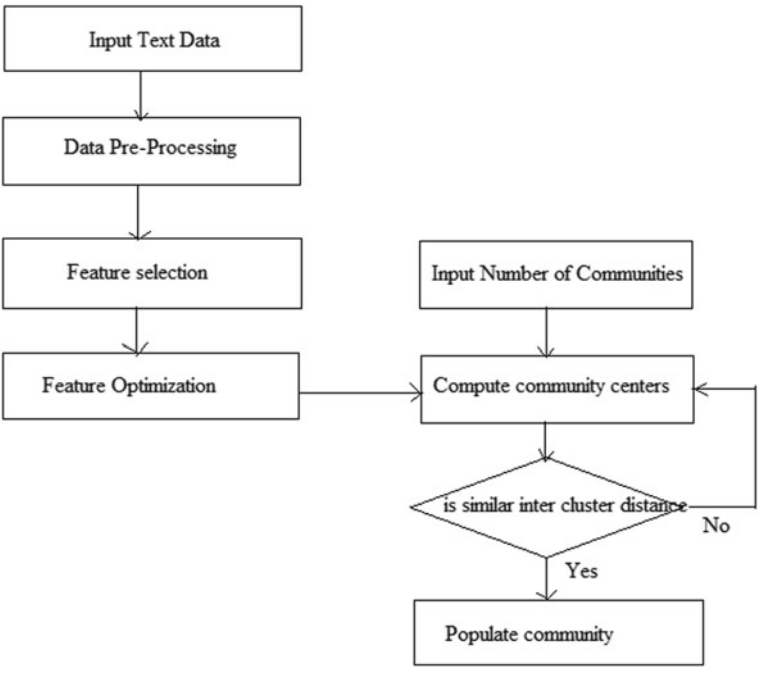


Fig. 1 Proposed methodology

1. Removing the symbols from the social media text blocks, these symbols include the special characters and unwanted symbols. In order to define this task, a user-defined list of unwanted symbols is prepared and provided as input to the system. System picks one word at a time and removes it from the entire input text data.
2. In this phase, the similar function of the data replacement is used as defined in the first pre-processing stage. In this phase in place of special character list, the list of words that is known as noisy words are included in list. Additionally, using the similar function the words from the entire input text dataset are removed.

Feature selection: The input text dataset is reduced after pre-processing of the data. Therefore, in further process the dataset is analyzed in order to find the important or highly weighted terms. Thus, the probability of each word is computed which is used in dataset. In this context, the word frequency is one of the most essential feature computation techniques that provide the probability of individual word in the given domain. The word frequency f can be computed by the ratio of total time a word appeared in document W_c or total count of words in document and the total words available in document W_t. Thus, word frequency is given by [7]:

$$f = \frac{W_c}{W_t}$$

Thus in this phase, the words are evaluated in order to find the importance of the word.

Feature optimization: In this phase, the optimization of features is performed during optimization; each individual social media text block is evaluated and only the high priority words remain, and the less priority words are removed from each individual posts. In addition to that it also considers all the features remain in the same length. Therefore, the largest length of features F_l and smallest length F_s are used to define the length of regular feature. The regular length of feature is computed as:

$$R = \frac{F_l + F_s}{2}$$

where R is length of each feature. The feature list which does not have enough data, null values are appended at their end, and similarly the feature list which has large amount of data are reduced on the basis of word frequency sorting.

Input number of communities: As we know that any unsupervised learning technique or clustering algorithm needs an input number of clusters to identify. Basically, the entire process leads to design a clustering approach, thus number of clusters are required to process data. In the similar manner, the number of communities is a user input which is provided by the user to identify the data communities. According to the user's input, algorithm computes the different possible communities in data.

Compute community centers: That phase initially accepts the two inputs: First, the number of communities needs to be detected and second the list of regular features. Initially, the similar number of instances of data is selected from the feature list as

number of communities is needed to detect randomly. These selected instances of features are termed here as the community centers. Now each individual community center is compared with the all the list of data instances. In order to compare the community center and instance of data, two distance functions are used.

1. **Word-to-word comparison**: In order to compare and find the difference among two instance features, the Levenshtein distance [8] is used. That results the amount of characters similar to each other.
2. **Combined difference**: After computing the difference between two data instances, the obtained dissimilarity is converted into the percentage value. Because Levenshtein distance values can be higher than the 100 points, thus to regulate the values the percentage conversion is performed.

Based on the obtained distance among the points (selected community center and the data instance), the community is prepared which is having maximum similarity. Therefore, the similarity is given by:

$$\text{Sim} = 100 - \text{difference}$$

After computing the first initial phase of community, the optimization of the community centers is required, by which the dense and more effective communities are identified. In order to obtain this, combined similarity score or the internal community distance is computed by the following formula:

$$\text{ID} = \frac{1}{N} \sum_{i=1}^{N} \text{Sim}_i$$

If the ID values of the community centers are in a range, then the optimization process is stopped, otherwise the previous community centers are again computed. During computation of the new community centers, the intersection of nearest points is used to replace the previous community center, and again the process of similarity measurement is performed. This process is carried out till all the IDs remain in a range or the numbers of default evaluation cycles are reached.

Populate community: That is the final phase of modeling where the computed communities are visualized using the graphical manner. That visualization helps to provide understanding about the instances of data which belongs to the same community. In addition to that it also provides the information which data is partially similar between more than on communities. In addition to that the performance of the system is also computed in this phase for recognizing the efficiency and accurate cluster formation ability.

This section provides the detailed understanding about the proposed community detection model. In next section, the algorithm of the system is provided.

Table 1 Proposed algorithm

Input: Text dataset D, Number of communities C, Optimization rang R Output: number of communities groups G
Process: 1. $R = readDataset(D)$ 2. $P = PreProcessData(R)$ 3. $T_N = TokenizeData(P)$ 4. $for(i = 1; i \leq N; i++)$ a. $[f, T] = \frac{W_c}{W_t}$ 5. $end\ for$ 6. $Om = feature.Optimize(f, T, P)$+ 7. I_c = Select C instance from Om+ 8. $for(j = 1; j \leq C; j++)$ a. $X = I_j$ b. $for(k = 1; k \leq m; k++)$ i. $D = \text{LevenshteinDistance}(X, O_k)$ ii. $if(D < 0.25)$ 1. $G_j(I_j, O_k)$ iii. $end\ if$ c. End for 9. End for 10. $for(l = 1; l \leq C; l++)$ a. $ID_l = \frac{1}{N}\sum_{i=1}^{N} Sim_i$ b. $if(ID_l \leftarrow R)$ i. Return $G_j(I_j, O_k)$ c. Else i. Go to step7 d. End if 11. End for

2.3 *Proposed Algorithm*

This section provides the summarized steps of the proposed methodology in terms of algorithm steps. The step process of the working is defined using Table 1.

3 Results Analysis

This section provides the evaluation of the proposed text mining technique using community detection approach. Therefore, different performance parameters are computed and their captured results are reported in this section.

3.1 *Precision*

Precision of the any algorithm demonstrates the part of data actually relevant to the current instance of community. That can be evaluated by the following formula:

$$\text{precision} = \frac{\text{relevant pattern} \cap \text{total pattern}}{\text{total pattern}}$$

The precision of the proposed technique of text mining is defined in Fig. 2. In this diagram, the X-axis indicates the amount of data instances used for experimentation and the Y-axis shows the obtained precision values between 0 and 1. According to the demonstrated results, the performance of the system improves with the amount of data instances for community identification. Therefore, the proposed model is acceptable for the utilizing with the different text mining application.

Fig. 2 Precision

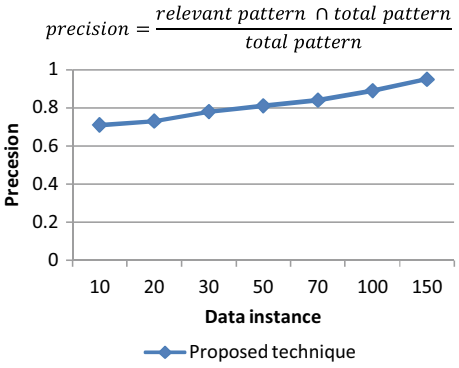
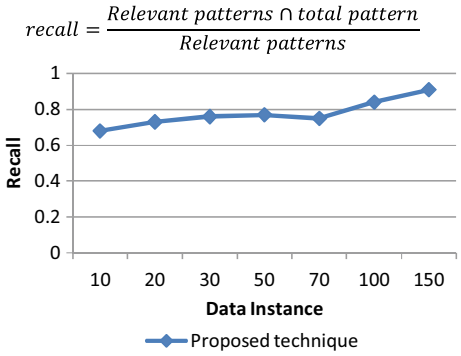


Fig. 3 Recall



3.2 Recall

Recall is the amount of data that are recognized during the clustering process is relevant to be the same cluster. That can be estimated using the following formula:

$$\text{recall} = \frac{\text{Relevant patterns} \cap \text{total pattern}}{\text{Relevant patterns}}$$

The computed recall of the community detection-based text clustering or community detection system is described in Fig. 3. The X-axis of the diagram explains the amount of data instances consumed during the experimentation. Additionally, the Y-axis demonstrates the obtained fraction of recall values between 0 and 1. According to this parameter, the performance is acceptable due to continuous improvement of recall values for increasing amount of data instances.

3.3 F-measures

That measure combines precision and recall in terms of harmonic mean of precision and recall rate of the obtained results that can also be termed F-measure or balanced F-score:

$$\text{F-measure} = 2 \cdot \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

The F-measures of the proposed text mining technique are given in Fig. 4. The X-axis of this figure contains the amount of experimental data supplied, and the Y-axis shows the obtained harmonic mean of precision and recall values. The result demonstrates that both the parameters are enhanced with the increasing amount of instances. Therefore, the proposed data model is acceptable for real-world text mining applications.

Fig. 4 F-measures

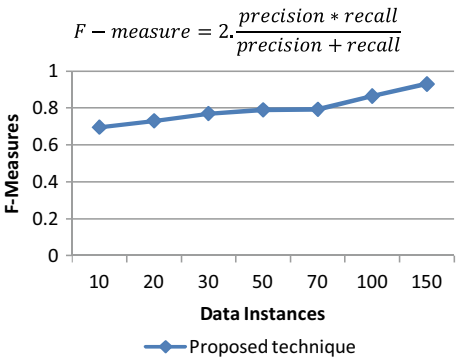
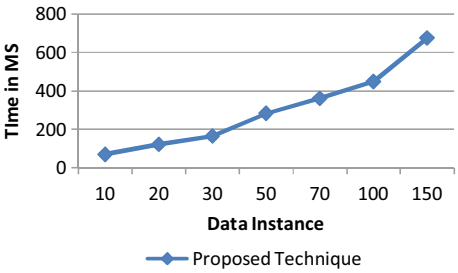


Fig. 5 Time requirements



3.4 Time Requirements

The algorithms need a time to process the input data using the applied algorithms. The amount of time required for this purpose is termed here as time requirements or time complexity of the algorithm. The time requirement of the proposed technique is defined using Fig. 5. In this figure, X-axis indicates that the amount of dataset instances is produced for experiments and the Y-axis simulates the obtained time requirement. The time measurement is performed here in milliseconds. According to the obtained performance, the proposed model time consumption increases with the amount of data due to the number of optimization phases, in order to find optimal results. Thus, the model is acceptable for working with accurate data analysis.

3.5 Memory Usage

The processes need an amount of memory to be executing their task successfully. This amount of main memory is termed as memory usages or the space complexity of the algorithms.

The required amount of main memory is demonstrated using Fig. 6. The amount of main memory is described in Y-axis of the diagram, and the X-axis contains the

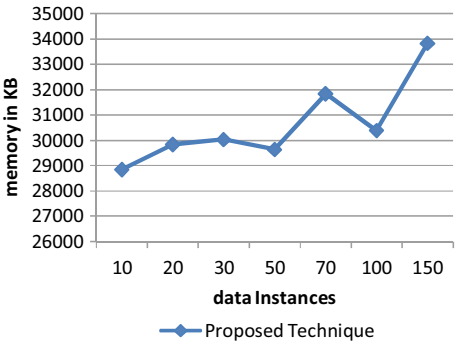


Fig. 6 Memory usage

amount of data size provided input for processing. According to the obtained memory utilization graph, which is measured in KB (kilobytes); it increases as the amount of to be processed. But the nature of memory usages is not obtained in regular manner that produces uneven ups and downs.

4 Conclusion and Future Work

This section provides the summary of the entire work performed for designing and developing the proposed text clustering technique. The summary of the work is demonstrated here as the conclusion, and the limitations and possible future extension are described as the future work for the system.

4.1 Conclusion

The proposed technique is community detection technique for mining the text data. This technique used the text data and conducted analysis in three main phases: first the pre-processing or preparation of data and their format, second the feature selection and optimization, and finally implementation of cluster formation technique. The model is a continuous process of data analysis and optimization which can take a significant amount of time for completing the process of accurate data analysis. Thus, two additional limits or stopping criteria are developed. First criteria are to reach the number of iterations reached and second are if the objective function is accomplished during the clustering processes. Both the processes help to perform the accurate and dense clusters which additionally also provides the information about the shared data instances.

The implementation of the proposed technique is provided in JAVA technology. After the implementation, the experimental results are computed and on the basis of these experiments, the outcomes of the system are reported in Table 2.

Table 2 Performance summary

S. no.	Parameters	
1	Precision	Optimizes with the amount of data but becomes consistent after a significant amount of data
2	Recall	Optimizes the accuracy of recognition with amount of data
3	F-measures	Increases with the amount of data means provides more promising results
4	Time requirement	Increase with the amount of data but also depends upon the number of evaluation cycles
5	Memory usage	Uneven in nature and increases with the size of input data

According to the obtained performance, the proposed technique is found acceptable for accurate data analysis.

4.2 Future Work

The proposed work is a promising approach for analyzing data that can be used in various complicated places where the pre-labeled data is not available; therefore, the following extensions are possible with the proposed technique.

1. The technique can be extended for social media topic tracking and trending topic detection
2. Providing good efforts for stream data mining techniques
3. Suitable to utilize the digital document retrieval system implementation

References

1. He, W., Zha, S., Li, L.: Social media competitive analysis and text mining: a case study in the pizza industry. Int. J. Inf. Manage. **33**, 464–472 (2013)
2. Patel, R., Sharma, G.: A survey on text mining techniques. Int. J. Eng. Comput. Sci. **3**(5), 5621–5625. ISSN:2319-7242 (2014)
3. Mostafa, M.M.: More than words: social networks' text mining for consumer brand sentiments. Expert Syst. Appl. **40**, 4241–4251 (2013)
4. Aggarwal, C.C., Zhao, Y., Yu, P.S.: On the use of side information for mining text data. IEEE Trans. Knowl. Data Eng. **26**(6) (2014)
5. Kou, G., Peng, Y., Wang, G.: Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Inf. Sci. **275**, 1–12 (2014)
6. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval **2**(1–2), 1–135 (2008)

7. Nassirtoussi, A.K., Aghabozorgi, S., Wah, T.Y., Ngo, D.C.L.: Text mining for market prediction: a systematic review. Expert Syst. Appl. **41**, 7653–7670 (2014)
8. Michael Gilleland, Merriam Park Software: Levenshtein distance. In: Three Flavors. https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein%20Distance.htm