Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan Vol. 29 No. 1-2024
ISSN:1813-1786 (Print) 2313-7770 (Online)

Enhancing Text Mining Efficiency by Using Bond Energy Algorithm for Document Clustering

Bushra[1], S. M. Adnan[2], W. Ahmad[3], I. Mahmood[4], G. Mustafa[5], V. Dattana[6]

[1,2,3]*Department of Computer Science, University of Engineering and Technology Taxila, Pakistan*
[4,6]*Department of CSMIS, Oman College of Management and Technology, Oman*
[5]*Department of Computing, Shifa Tameer-e-Millat University, Islamabad, Pakistan*

[2]syed.adnan@uettaxila.edu.pk

Abstract- In the context of the growing volume of textual data and the imperative for automated solutions, this research addresses the challenges of topic analysis and document clustering. Topic analysis and document clustering are the two significant hurdles in the field of text mining. This research aims to extract topics from documents and cluster them based on these topics by implementing a distributed database technique. The study investigates popular topic extraction models such as Latent Dirichlet Allocation (LDA) and Bond Energy Algorithm (BEA), which generate vector representations and make semantic clusters across documents. The research evaluates the clustering performance of documents using various metrics suitable for the task. The goal is to group similar documents in the same cluster to improve document organization and retrieval. Research findings indicate that the integration of topic modeling with vertical partitioning significantly enhances clustering accuracy while concurrently reducing computational and storage overheads.

Keywords- Text Mining, Document Clustering, LDA, Topic modeling

I. INTRODUCTION

In an era of a growing flood of data, the critical task of extracting valuable knowledge requires an integration of domain expertise and advanced computational technology, as illustrated by the practice of data mining, which explores unprocessed data sets to identify hidden patterns and correlations, thus dealing with the prominent challenge of overloading with information in current society. The ever-increasing production of data applies to texts, many applications are switching from human paperwork such as e-books, research papers, journals, scientific articles, and the web to automated solutions. Millions of publications are published yearly in tens of thousands of newspapers in the English language alone [1], and this number is only rising. For example, the Elsevier repository alone provides more than 63,000 works on the COVID-19 topic [2]. All of this is unstructured, which makes it difficult to find relevant information. This underscores the need for tools that can effectively find out, group, and abstract substantial collections of text documents. In the field of text mining, two significant hurdles emerge: topic analysis and document clustering [3]. Clustering is the process of structuring texts into coherent clusters, with intra-clusters sharing greater similarities than inter-clusters. Topic modeling is a type of clustering that is based on probabilities for analyzing text and unveiling semantic patterns within a text. Topic distributions represent texts, and topics are evaluated based on word relevancy 1.

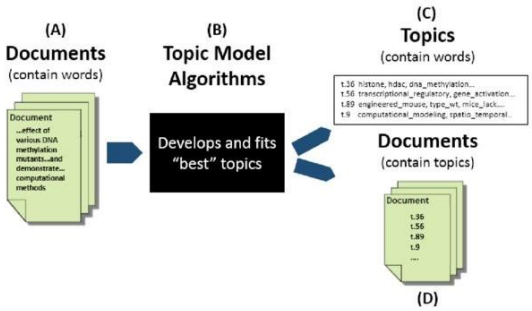

Fig. 1. Basic pipeline of Topic Modeling

Fig. 1. illustrates the fundamental pipeline employed in topic modeling, where clustering aims to identify homogenous groupings of distributions across corresponding articles. These groups contain the semantics of linked or unrelated sections within text corpora. Rather than adhering to the conventional bag-of-words approach [4] because of the high dimensionality and lack of connection between text features, topic modeling represents the semantic interpretation of text across documents based on clusters. Combining topic modeling and document clustering is a complex task because managing a large volume of diverse and unstructured information makes it tricky to search. Our research will proceed with two-step LDA [5] for topic modeling and BEA [6] for vertical fragmentation. The BEA calculates the Attribute Usage Matrix

(AUM), Attribute Affinity Matrix (AFM), and Vertical Fragmentation (VF) for document clustering. BEA outshines traditional methods such as DBSCAN, OPTICS, or K-means in different manners. Firstly, it delivers better clusters by overcoming the major issues in standard clustering. Furthermore, the well-structured boundaries of BEA make it more objective. To achieve this goal, Distributed Database Management Systems (DDBMS) are used. Our contributions include:

- Efficient clustering of different documents using topic modeling.
- Different standard text document datasets are utilized to showcase the efficiency of suggested approaches.
- A novel method over a large collection of raw datasets is proposed.
- To cluster documents, a unique distributed database approach has been applied.

Moving on to the following section, which includes literature from several recently published works. Following that, the methodology was chosen for performing the research. The next section discusses the algorithm's architecture and execution. Finally, the conclusion derived from our findings identifies prospective areas for further research.

II. LITERATURE REVIEW

The amount of unstructured data is growing every day due to the rise of smart devices and similar technologies. A significant body of work has already been done in the areas of document clustering and topic modeling. Grouping and figuring out themes [7] and topic modeling [8] are two essential methods for exploring and categorizing text. Several text mining methodologies have been recently introduced and can be used in the formation of documents, as well as for classification, clustering, summarization, and topic modeling. Approaches to text mining could be either supervised or unsupervised. Text mining has been implemented in the previous era for a wide range of purposes, including the development of pharmaceutical prescription medications, spam filtering, summarizing customer feedback, and monitoring community input [9].

However, it is still a new endeavor in the industrial circle. Kung et al. [10] employed text classification algorithms to find issues that are related to quality in microelectronic devices constructed on the unstructured data in hold records. Dong and Liu [11] proposed a method for identifying and classifying websites based on predetermined website genres. For instance, [12] provides a Reuters dataset where theme patterns are automatically discovered using an LDA-based topic modeling approach [13-14] Researchers address understanding of the sentiments and concerns surrounding online classes. This is especially from the perspective of learners and educators, by leveraging data from social media platforms like Reddit. By using sentiment analysis and topic modeling techniques, research aims to uncover hidden insights and trends in public opinion over time.

Another approach to address the value of creativity in marketing through a comprehensive analysis of the creativity in marketing (CiM) literature seems well-structured and promising. In this research the topic modeling findings present the results of the structural topic modeling analysis, including the ten key topics extracted from the CiM literature [15]. Research article provides a comprehensive survey of topic modeling techniques, covering various aspects including categorization of models, evaluation criteria, applications, available software tools, datasets, and benchmarks. Researcher focus on the development and tuning of neural topic models and the integration of pre-trained language models [16]. We compare and analyze the different approaches for document clustering in Table I.

Table 1 Comprehensive Literature Review

Reference	Year	Approach	Strength	Limitation
[17]	2022	BERTopic + TF-IDF	2nd learns coherent language patterns on a wide range of tasks	Assumes single-topic
[18]	2021	Robust Multi-view Document Clustering using Cosine similarity and the Euclidean distance	Multiple similarity metrics for document clustering and surpass state-of-the-art systems	Need more space and runs slower in high dimensions
[19]	2020	Seeded-ULDA + NMF + K-means Seeded-ULDA produces better results on overlapping and non-overlapping datasets as compared to	Seeded-ULDA produces better results on overlapping and non-overlapping datasets as compared to LDA	Word-embedding not covered in overlapping data
[20]	2020	K-means, Cosine and soft cosine similarity	Achieving query performance with better speed and accuracy Use cosine similarity-based k-means. Cost, delay concerns, and accuracy	Use cosine similarity-based k-means. Cost, delay concerns, and data duplication are also not addressed
[21]	2020	Bayesian probabilistic approach based on Word Embeddings	Capturing semantic and syntactic word embed-dings using multivariate Gaussian distributions	Datasets has few categories and ignore topic correlations
[22]	2016	LDA + Term Frequency – Inverse Cluster Frequency	Fast Training with high accuracy as compared to state-of-the-art models	Limited dataset

III. PROPOSED METHODOLOGY

Our goal was to uncover document categories while keeping semantic and syntactic relationships. To achieve this, LDA for topic modeling [23] and an improved BEA to solve the challenges of document clustering are used. The primary purpose is to arrange documents into clusters, which include documents that share a similar topic. Figure 2 discusses the two-phase proposed methodology.

Dataset and its Preparation Phase

We used a large dataset with over 1,000 scientific documents. Each document contains a summary, called an abstract, from various research papers. These papers cover basic topics of computer science, mathematics, and science subjects, some are from specific quantitative biology, and quantitative finance topics. What's unique about this dataset is that it doesn't come with predefined labels or categories. With the help of Natural language processing (NLP), can find patterns and groups in these documents, helping us understand how these different subjects relate to each other. These are the raw documents that may contain punctuation marks and other special characters that must be deleted. The text preprocessing phase involves the following steps: tokenization, lemmatization, stop-word removal, word embeddings, stemming, and vectorization.

1) *Tokenization:* This step is used to undertake linguistic analysis of the text. Tokenization is dividing a string of text into slighter chunks called "tokens" by removing punctuation marks and other non-essential characters. The purpose of tokenization is to split the text into a stream of individual tokens so that it can be analyzed more easily.
2) *Stop-word Removal:* Every natural language has its gathering of stop-words. These are terms that have no lexical meaning yet appear often in texts. Articles, pronouns, and prepositions are frequent terms in papers that add no sense to the phrases [24]. This is accomplished by adding terms to the stop list that have a document frequency value greater than a predefined threshold [25]. This is known as automated stop-word creation.
3) *Lemmatization:* It refers to the act of reducing various forms of a word into a single base form, also known as the lemma. This process brings together different versions of a word, such as "builds", "building", and "built" and simplifies them to a common base form like "build".

Phase 1: Latent Dirichlet Allocation (LDA)

One of NLP's most prominent and widely adopted probabilistic topic modeling techniques is Latent Dirichlet Allocation (LDA) [26]. LDA is used to uncover and categorize topics within our dataset while also discerning their semantic significance. The application of LDA allows us to fix the number of topics present and to abstract the top "n" frequently occurring words within each topic. These word-frequency associations will serve as the foundation for the second phase, where they will be utilized as user queries. Figure 3 provides a visual representation of the processing steps involved in the application of LDA.

1) Choice of the Ideal Number of Topics: In LDA, the 'k' value represents the number of linear discriminants (features) extracted from the dataset. The optimal 'k' value depends on the specific problem at hand and can be determined through a combination of domain knowledge, trial and error, and performance evaluation. Algorithm 1 defined our approach to select the best number of k.

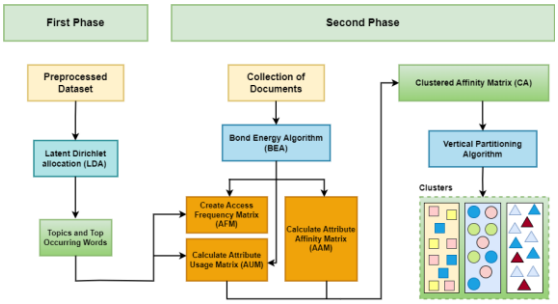

Fig. 2. Refined Approach: A Two-Phase Methodology for Enhanced Predictive Modeling

Algorithm 1 Selecting the Ideal number of topics in LDA

Input: Number of classes in the dataset, C
Output: Number of linear discriminants, k
// Consider the number of classes
if C > k **then**
k ← C − 1 {Ensure k is less than the number of classes}
end if
// Evaluate classification accuracy
for k from 1 to C − 1 **do**
Perform cross-validation or hold-out validation to evaluate LDA performance with k discriminants
end for
k ← argmax (k1, k2, . . . , kC−1) {Choose k with highest accuracy}
// Leverage domain knowledge
// (No specific mathematical notation)
// Consider the trade-off between complexity and performance
// (No specific mathematical notation)
Return Number of linear discriminants, k

Phase 2: Bond Energy Algorithm (BEA)

The bond energy algorithm (BEA) [6], originally invented by McCormick et al. in 1972, and applied in various contexts [27], is designed to group attributes within a relation based on their degree of

closeness [28]. This algorithm plays a significant role in distributed databases by determining how to group and physically place data on a disk.

BEA is applied to create clusters of attributes by grouping

them based on their similarities or relationships. After clustering the attributes, the algorithm computes the cost of assigning each cluster to different sites or locations. Similar to BEA, our proposed methodology aims to organize the documents into clusters, with each cluster containing documents related to the same topic. The procedural outline of our proposed methodology comprises two phases, as depicted in Figure 4. To implement BEA, we start with an Attribute Usage (AU) matrix as input. This matrix comprises attributes

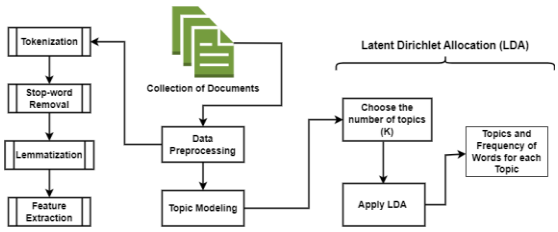

Fig. 3. Detail architecture and flow of LDA [7]

as columns, queries as rows, and query access frequencies as values. Unlike previous data fragmentation algorithms that used an Attribute Affinity (AA) matrix derived from the input AU matrix, BEA directly employs the AU matrix itself. See the detailed steps in algorithm 2.

- *Create AU Matrix:* We create a search function that will examine each document one by one for a certain query and determine if the query is used or not in that document; if it is, it will insert 1 in the AU matrix; otherwise, it will give 0. AU = [∀di ∈ Dqk] Action for each document di in query qk.
- *Create Access Frequency Matrix:* Frequency of top keywords obtained from our first LDA phase, which occurred in documents. Each cell in the matrix contains a numeric value that represents the number of times a certain query keyword was accessed in a single document. The matrix has a single row for each query and one column for each document. This matrix is used to summarize the number of times various documents have been accessed via queries.
- *Create AA Matrix:* The AA matrix is a n x n matrix with papers in every row and column. The AU matrix and access frequency matrix are input into the AA matrix.
- *Create CA Matrix:* Add two randomly chosen columns from the AA matrix to the CA matrix. [7], [9-11], [29]. Then put the remaining n-i columns in position i+1, where i is the number of columns already placed in the CA matrix. The positioning of these columns should be done so that the bond energy change is maximized. This step repeats until no more columns are left. At the end, the positioning of the rows should be adjusted to the relative location of the columns once the column order has been decided. The CA matrix's symmetry is therefore restored.

Although the BEA efficiently predicts the ordering of attributes, personal expertise is required to form clusters. To achieve accurate clusters, vertical partitioning is introduced that divides large CA matrices upon the bond energy algorithm.

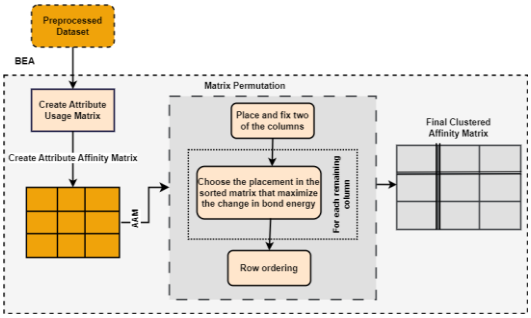

Fig. 4. Insight into the Proposed BEA algorithm

Algorithm 2 Attribute Clustering using Bond Energy Algorithm

Input: AU matrix AU
Output: CA matrix CA
Initialize AU matrix AU with zeros
for each query q_i in the dataset **do**
  for each attribute A_j utilized by q_i **do**
    $AU[q_i][A_j] \leftarrow 1$
  end for
end for
Create an Access Frequency Matrix based on query execution counts from sites
Initialize the AA matrix AA with zeros
for each pair of attributes, A_i and A_j **do**
  Compute affinity $Aff(A_i, A_j)$ using AU, Access Frequency Matrix
  Store $Aff(A_i, A_j)$ in AA[i][j]
end for
Use Bond Energy Algorithm (BEA) to create a CA matrix CA from AA
Reorder rows and columns of AA to maximize global affinity measure AM
Return matrix CA

Vertical Partitioning Algorithm

Vertical partitioning divides a large matrix (CA) into smaller sub-matrices, reducing calculations and usage of memory. The size and number of sub-matrix are task-specific. Typically, it is recursive splitting for finer granularity from top to bottom. See Figure 5 for a better illustration of binary partitioning.

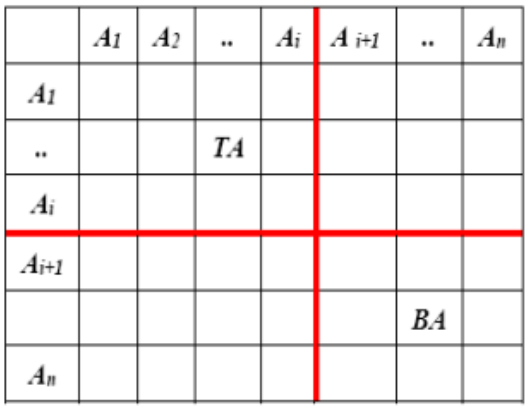

Fig. 5. Binary Partitioning

To determine the partition of the attributes from the matrix CA, we use the exhaustive technique as follows:

$$Z = CTQ \times CBQ - COQ^2 (1) \quad (1)$$

$$CTQ = \sum_{q_i \in TQ} \sum_{\forall s_j} ref_j(q_i) aac_j(q_i) \quad (2)$$

$$CBQ = \sum_{q_i \in BQ} \sum_{\forall s_j} ref_j(q_i) aac_j(q_i) \quad (3)$$

$$COQ = \sum_{q_i \in OQ} \sum_{\forall s_j} ref_j(q_i) aac_j(q_i) \quad (4)$$

Where

$$AQ(q_i) = \{A_j | use(q_i, A_i) = 1\}$$

$$TQ = \{q_i | AQ(q_i) \subseteq TA\}$$

$$BQ = \{q_i | AQ(q_i) \subseteq BA\}$$

$$OQ = Q \setminus \left(TQ \bigcup BQ\right)$$

The complexity of the partitioning algorithm is n^2.

IV. Evaluation Metrics

Accurate evaluation measures are essential to gauge how well the clustering algorithm has grouped documents into meaningful clusters.

A. Silhouette Score

It quantitatively benchmarks clustering quality by considering both intra-cluster cohesion and inter-cluster separation. it raging from -1 to 1, assesses how well objects match their clusters and how poorly they match neighboring clusters. The calculation involves the formula:

$$Silhouette\ Score = \frac{(b-a)}{\max(a,b)} \quad (5)$$

where α is the mean intra-cluster distance and b is the mean nearest-cluster distance.

B. Davies-Bouldin Index

It evaluates clustering quality, focusing on the average similarity between each cluster and its closest counterpart. It assesses the separation and distinctiveness among clusters, taking into account both 'within-class' and 'between-class' similarities. Lower values indicate superior cluster separation. The calculation formula is:

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max\left(\frac{R_{ij} + R_{ji}}{d_i}\right) \quad (6)$$

where n is the number of clusters, R_{ij} is the average distance between points in cluster i, R_{ji} is the average distance between points in cluster j, and d_i is the distance between the centroid of cluster i and its farthest point.

C. Purity

Purity is a measure of how well data points are assigned to the majority class within each cluster, offering a clear indication of clustering accuracy. A higher purity score signifies better cluster quality. The purity calculation is:

$$Purity = \frac{1}{N} \sum_i \max\left(count(class_j, cluster_i)\right) \quad (7)$$

Here, 'N' is the total number of points, and count(class j), cluster i) is the number of points of class 'j' in cluster 'i'.

D. Entropy

Entropy measures the diversity of categories within a cluster, offering insights into the overall diversity of categories represented in each group. Lower entropy values indicate more homogeneity. The formula to calculate entropy:

$$Entropy = -\sum_i p_i log_2 (p_i) \quad (8)$$

where p_i is the proportion of data points in cluster i.

E. F-Measure

The F-Measure combines precision and recall values to assess how well clustering aligns with a reference partitioning of the data. It quantifies the ability of clusters to match predefined categories, making it valuable for external validation.

$$F - Measure = \frac{2.Precision \cdot Recall}{Precision + Recall} \quad (9)$$

Where precision is calculated as the ratio of true positive predictions to total positive predictions. Recall is the proportion of positive predictions to the total number of true positive cases.

F. Experimental Setup

The research was conducted using a personal computer equipped with an Intel(R) Core (TM) i5-4310U CPU running at 2.00GHz and 16 GB of Memory. Python was employed for creating and evaluating the proposed algorithms.

V. Results and Discussion

All experiments were conducted encompassing the following tasks:

I. Employing LDA on the preprocessed dataset to extract topics and identify the most frequently occurring words in each topic.
II. Employing BEA on the dataset and partitioning the matrix into clusters.

G. LDA Experiment

LDA needs predefined topics as input. To figure out the best topics, we took a trial-and-error approach. Several experiments are carried out to train numerous LDA models with different topic

numbers, and the results are compared and visualized. Figure 6 shows two curves representing changes in coherence and perplexity scores for models with topic numbers ranging from 2 to 20. Initially, two topics were chosen and the coherence scores improved as the number of topics increased. The scores reached a high point at nine topics, but after that, they gradually went down to larger numbers. Because LDA models have this built-in randomness due to the initial random topic assignments, I had to run the training multiple times to be sure about the pattern of this curve.

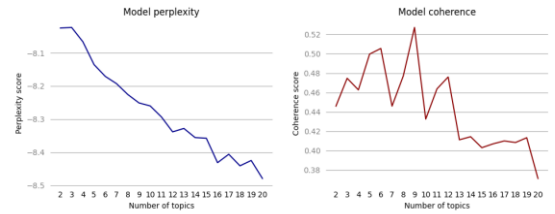

Fig. 6. Topic model coherence and perplexity

H. BEA on Documents and Matrix Partitioning into Clusters

This algorithm created vertical fragments or clusters, further organizing the dataset. To assess the quality and accuracy of the clusters, two distinct evaluation methods—the Silhouette Score and the Davies-Bouldin Index. The resulting values were 0.79 and 2.63, respectively, indicating favorable results with our proposed approach.

In addition, manual labeling is conducted on the dataset to validate the accuracy of the clusters. Various metrics, including the F-measure, Purity, and Entropy, were used to measure the precision of the clusters. In Table 2, we compare the existing state-of-the-art clustering algorithms such as k-mean, k-mean++, CADBE (Clustering Arabic Documents algorithm based on Bond Energy Algorithm), and CoclusMod with the proposed method.

Table 2 Comparison of Results by Clustering Techniques

Algorithm	Purity	Entropy	F-Measure
K-means	0.68	1.10	0.65
K-means++	0.67	1.11	0.64
CADBE	0.72	1.04	0.70
CoclusMod	0.67	1.07	0.63
Proposed Method	0.69	1.01	0.71

We compare all of the abovementioned metrics to each algorithm. Our proposed approach outperforms existing approaches based on purity, entropy, and F-measure. The CADBE achieved higher purity than the proposed technique because our approach makes clusters based on topics, and according to LDA, one document may have multiple topics. However, our proposed methodology performed well for an unstructured dataset, as indicated by the Davies-Bouldin index of 2.63 and silhouette score of 0.79. Additionally, it also performed better for a structured dataset, as observed in Table 2. Figure 7 shows graphic representations of performance outcomes for specific datasets.

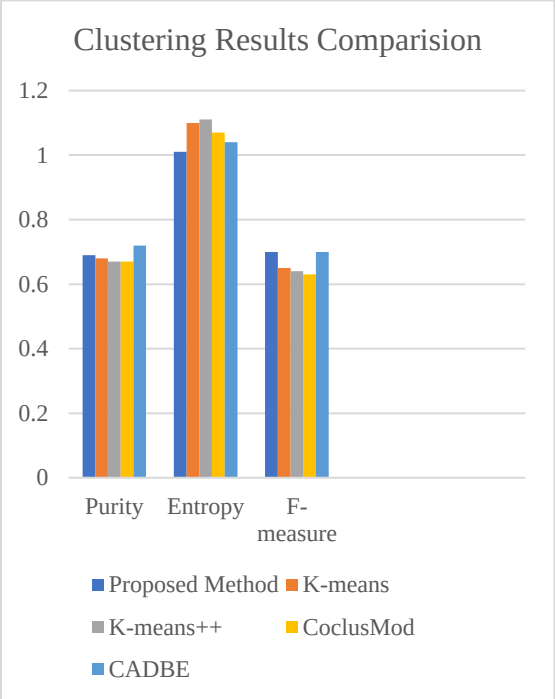

Fig. 7. The performance statistics of all traditional algorithms in comparison of proposed algorithm

Conclusions AND Future Work

Developed a text clustering technique based on the Bond Energy Algorithm. The Clustering of documents as a corpus using Bond Energy aims to comprehend cluster descriptions by finding and exhibiting naturally different clusters within complex data. Additionally, merging algorithms have been introduced to merge clusters based on links and interrelationships among them. The proposed model offers several advantages over past clustering algorithms. our algorithm distinguishes itself from conventional clustering methods by not requiring an initial number of clusters. To evaluate our algorithm, a series of experiments is conducted based on various monitored metrics, demonstrating that our algorithm significantly outperforms other algorithms like k-means, k-means++, spherical k-means, EM with Gaussian mixture model (diagonal covariance matrix per mixture component), and SLHA. The study's findings suggest that combining topic modeling and vertical partitioning can enhance clustering accuracy while simultaneously reducing computational and storage expenses. The proposed

algorithm shows promising results regarding purity and entropy rates.

For future analyses, investigating new partitioning and clustering methods, such as graph-based approaches or hybrid techniques, holds promise for enhancing the effectiveness of document organization and processing. These techniques could also be extended to other domains, like image or video clustering, to optimize data processing and analysis.

REFERENCES

[1] R. Johnson, A. Watkinson, and M. Mabe, The STM report. An overview of scientific and scholarly publishing, 5th ed., October 2018.

[2] "Novel coronavirus resource directory," 2020, [cited 2020 Oct 1].[Online]. Available: https://www.elsevier.com/novel-coronavirus-covid-19

[3] M. Allahyari et al., "A brief survey of text mining: Classification, clustering and extraction techniques," arXiv preprint arXiv:1707.02919, 2017.

[4] H. Ren et al., "A weighted word embedding model for text classification," in Database Systems for Advanced Applications: 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22–25, 2019, Proceedings, Part I, vol. 24. Springer, 2019.

[5] B. Griciute˙ et al., "Topic modeling of Swedish newspaper articles about coronavirus: a case study using Latent Dirichlet Allocation method," arXiv preprint arXiv:2301.03029, 2023.

[6] AlMahmoud, R. H., Hammo, B., & Faris, H. (2020). A modified bond energy algorithm with fuzzy merging and its application to Arabic text document clustering. Expert Systems with Applications, 159, 113598.

[7] D. Manning Christopher, R. Prabhakar, and S. Hinrich, Introduction to Information Retrieval. Cambridge University Press, 2008.

[8] D. Blei, "Probabilistic topic models," Communications of the ACM, vol. 55, no. 4, pp. 77–84, 2012.

[9] S. Debortoli et al., "Text mining for information systems researchers: An annotated topic modeling tutorial," Communications of the Association for Information Systems (CAIS), vol. 39, no. 1, p. 7, 2016.

[10] Y.-H. Liu et al., "Using text mining to handle unstructured data i semi-conductor manufacturing—yan-hsiu liu," in 2015 Joint e-Manufacturing and Design Collaboration Symposium (eMDC) & 2015 International Symposium on Semiconductor Manufacturing (ISSM). IEEE, 2015.

[11] B. Dong and H. Liu, "Enterprise website topic modeling and web re-source search," in Sixth International Conference on Intelligent Systems Design and Applications. IEEE, 2006.

[12] L. AlSumait, D. Barbar´a, and C. Domeniconi, "On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking," in 2008 Eighth IEEE International Conference on data mining. IEEE, 2008.

[13] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, no. Jan, pp. 993–1022, 2003.

[14] Li, Shanghao, et al. "Sentiment analysis and topic modeling regarding online classes on the Reddit Platform: educators versus learners." Applied Sciences 13.4 (2023): 2250.

[15] Das, Kallol, et al. "Creativity in marketing: Examining the intellectual structure using scientometric analysis and topic modeling." Journal of Business Research 154 (2023): 113384.

[16] Abdelrazek, Aly, et al. "Topic modeling algorithms and applications: A survey." Information Systems 112 (2023): 102131.

[17] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," arXiv preprint arXiv:2203.05794, 2022.

[18] B. Diallo et al., "Multi-view document clustering based on geometrical similarity measurement," International Journal of Machine Learning and Cybernetics, pp. 1–13, 2022.

[19] M. Mustafa et al., "Urdu documents clustering with unsupervised and semi-supervised probabilistic topic modeling," Information, vol. 11, no. 11, p. 518, 2020.

[20] S. Tarun, R. S. Batth, and S. Kaur, "A novel fragmentation scheme for textual data using similarity-based threshold segmentation method in distributed network environment," International Journal of Computer Networks and Applications, vol. 7, no. 6, p. 231, 2020.

[21] G. Costa and R. Ortale, "Document clustering meets topic modeling with word embeddings," in Proceedings of the 2020 SIAM International Conference on Data Mining. SIAM, 2020.

[22] L. H. Suadaa and A. Purwarianti, "Combination of Latent Dirichlet Allocation (LDA) and term frequency-inverse cluster frequency (tfxicf) in Indonesian text clustering with labeling," in 2016 4th International Conference on Information and Communication Technology (ICoICT). IEEE, 2016.

[23] A. Pon, C. Deisy, and P. Sharmila, "A case study on topic modeling approach with Latent Dirichlet Allocation (LDA) model."

[24] J. Boyd-Graber, Y. Hu, and D. Mimno, "Applications of topic models," 2017, vol. 11, no. 2-3, pp. 143–296.

[25] B. Hirchoua, B. Ouhbi, and B. Frikh, "Topic modeling for short texts: A novel modeling method," in AI and IoT for Sustainable Development in Emerging Countries: Challenges and Opportunities. Springer, 2022, pp. 573–595.

[26] H. Jelodar et al., "Latent Dirichlet Allocation (LDA) and topic modeling: models, applications, a survey," Multimedia Tools and Applications, vol. 78, pp. 15 169–15 211, 2019.

[27] S. Climer, W. Zhang, and T. Joachims, "Rearrangement clustering: Pitfalls, remedies, and applications," Journal of Machine Learning Research, vol. 7, no. 6, 2006.

[28] M. Mi et al., "An improved differential evolution algorithm for tsp problem," in 2010 International Conference on Intelligent Computation Technology and Automation. IEEE, 2010.

[29] T. Hofmann, "Probabilistic latent semantic indexing," in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999.