Clustering LLM-based Word Embeddings to Determine Topics from Bangla Articles

Rifat Rahman
Department of CSE & IAT
Bangladesh University of Engineering & Technology
rifatrahman05007@gmail.com

Mohammed Eunus Ali
Department of CSE
Bangladesh University of Engineering & Technology
mohammed.eunus.ali@gmail.com

Abstract

Topic modeling methods identify fundamental themes within textual documents, facilitating an understanding of the insights inside them. Traditional topic modeling approaches are based on the generative probabilistic process that assumes the document-topic and topic-word distribution. Hence, those approaches fail to capture semantic similarities among words inside the documents and are less scalable with the vast number of topics and documents. This paper presents a method for capturing topics from Bangla documents by clustering the word vectors induced from LLM models. Corpus statistics are integrated into the clustering & word reordering process within each cluster or topic to extract the top words. Additionally, we deploy dimensionality reduction techniques, such as PCA, prior to clustering. Finally, we perform a comparative study and identify the best-performing combination of clustering and word embedding methods. Our top-performing combination outperforms the traditional probabilistic topic model in capturing topics and top words per topic, and excels notably in terms of computational efficiency and time complexity.

1 Introduction

Topic modeling is a data analysis technique highly used for text mining in Natural Language Processing (NLP) (Sia et al., 2020). Topic models discover the key themes and patterns from a large corpus of textual documents by analyzing and grouping words into clusters or topics based on their co-occurrences inside the documents (Boyd-Graber et al., 2017). Thus, it helps analyze big data, capturing subjects discussed in the text. Topic modeling has several usabilities in NLP, like feature engineering (Rahman, 2020b), sentiment analysis (Rahman et al., 2022), document categorization (Rahman, 2020a), etc. In this study, we present a benchmark investigation focused on the weighted clustering of LLM-based word embeddings to extract topics from Bangla language documents. Additionally, we undertake a comparative analysis, directly comparing our top-performing approach with the widely adopted topic modeling technique, Latent Dirichlet Allocation (LDA).(Blei et al., 2003).

Word embedding refers to the vector representations of words in multi-dimensional space that keep the semantic characteristics of words (Rahman, 2020b) within the corpus. As it holds the semantic attributes of words, similar words stay closely in the multi-dimensional space. Hence, clustering those vectors gives valuable insights related to the main themes of documents. Again, pre-trained LLM models (Wang et al., 2019) consider the attention mechanism that enables the model to effectively hold long-range dependencies among words or phrases in texts. So, we apply clustering on LLM-based word embeddings to identify the topics as clusters of words.

The majority of studies apply probabilistic approaches (Blei et al., 2010) (e.g., LDA, pLSA, biterm topic model, etc.) or linear algebra-based techniques (e.g., LSA). Probabilistic or linear algebra-based topic models do not consider linguistic features inside the corpus. Very few works (De Miranda et al., 2020; Sridhar, 2015; Sia et al., 2020) that conduct clustering on word embeddings do not take standardized word vectors or LLM-based word embeddings into account. Most notably, clustering word embeddings has not yet been explored for topic modeling in the context of the Bangla language.

Our *objective* of this work is to propose a topic model technique by identifying the best-performing combination of clustering algorithm and LLM-based word embeddings that outperforms traditional probabilistic topic models.

To achieve our goal, we introduce a novel topic modeling technique in Bangla by clustering LLM-based word embedding methods. We apply centroid-based weighted clustering as centroid-

Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025), pages 309–321

based clustering helps identify the top words of individual clusters based on the distance from the cluster centers. Weighted clustering is done by incorporating statistical information from the corpus. We utilize LLM-based pre-trained word embeddings as those models have been trained on the vast amount of various contextual information. So, those word embeddings can be regarded as standard embeddings. The dimensionality reduction algorithm, Principle Component Analysis (PCA) (Wold et al., 1987), is applied to word embeddings before clustering. After identifying the clustered words in each cluster, we reorder words according to the frequency statistics of words within the documents to find the top X words. We also do comparison among various combinations of LLM-based word embeddings & clustering methods and find the best-performing combination. The best-performing combination is then compared against LDA. In the meantime, we explore the word embedding technique that performs comparatively well with all clustering processes and the clustering method that gives the best result for all types of vector representations. Our best-performing combination (average NPMI score=0.31) extracts top words having more point-wise mutual information or coherence within a topic than LDA (average NPMI score=0.18). Moreover, our approach requires less run time and computational power than LDA.

The *contribution* of our study is threefold:

- We propose the first word-embedding & clustering based topic modeling for Bangla.
- We incorporate document statistics and linguistic features in our topic models that help extract more informative topic words.
- We conduct a benchmark study and identify the best word embedding technique for clustering and the best clustering method for all kinds of word embeddings.

2 Related Works

Clustering is a much-used approach for analyzing documents and texts. A plethora of studies apply clustering on texts for readability measurement (Cha et al., 2017), argument identification from texts (Reimers et al., 2019), and text classification tasks (Sato et al., 2017). Cha et al. (2017) apply clustering word embeddings for predicting text readability and show how the approach improves overall performance and the text is suitable for the intended reader's comprehension level. They also perform sentence matching based on semantic similarity by clustering word embeddings. Another study (Sato et al., 2017) utilizes the clustering method on paragraph vectors to capture semantic similarities among documents and phrases that outperforms the co-embedding method utilizing bag-of-words representation. Again, Reimers et al. (2019) explore the effectiveness of two contemporary contextualized word embedding techniques, ELMo and BERT, for argument-searching tasks. These techniques are used to classify and cluster arguments specific to various topics. However, clustering word embedding has little been explored regarding topic modeling.

Several studies attempt to include word embeddings in probabilistic topic modeling. Liu et al. (2015) introduce topical word embeddings (TWE) for creating multi-prototype word embeddings where word vectors are different based on topics. They use LDA to determine word topics and apply collapsed Gibbs sampling to assign topics to each word token. Another research work (Nguyen et al., 2015) develops a hybrid version of the topic modeling method, expanding two different probabilistic topic models by integrating word embeddings trained on a little data to figure out the word-topic distribution. These models achieve notable improvements in topic coherence, document clustering, and document classification. Das et al. (2015) introduce an alternative parameterization of "topics" in the LDA framework where topics are represented as categorical distributions over concealed word types, combined with multivariate Gaussian distributions in the embedding space. Some authors (Zhao et al., 2017) present the WEI-FTM that yields focused topics with representative words, enhancing perplexity and topic quality. It efficiently employs a Gibbs sampling algorithm for inference, accommodating both regular and short texts without loss of generality. Dieng et al. (2020) implement the Embedded Topic Model (ETM), merging probabilistic generative topic models with word vectors that outperform LDA for identifying topics from short-sized texts or documents. The model also shows robustness with larger vocabularies.

Very few works have investigated the efficacy of directly clustering word embeddings for topic analysis. Xie and Xing (2013) propose a Multi-

Grain Clustering Topic Model (MGCTM) that clusters similar documents and introduces topics for those individual clusters. Every word is assigned a variable that represents the origin of a global (combination of documents) or local (separate documents) topic. In CluWords, Viegas et al. (2019) utilize nearest words from pre-trained embeddings to create meta-words for document representation and use the Tf-Idf score as weight for weighted clustering. They introduce a novel word representation technique using the syntactic & semantic information obtained from word embedding. Sridhar (2015) present an unsupervised topic model for short texts that employ soft clustering and Gaussian mixture models (GMM) with distributed word representations to overcome sparse word co-occurrence patterns. Another research work (De Miranda et al., 2020) utilizes word2vec to get vector representations of words and implement a mapping mechanism that maps a word vector to a specific topic, aiming to identify topics within texts. However, none of these works utilize standard word embeddings or transformer-based word vectors that are more robust. Sia et al. (2020) apply different clustering processes on various word embedding methods and identify the best combination. But they only consider Tf as corpus statistics.

There are few works (Hasan et al., 2019; Helal and Mouhoub, 2018) in the context of the Bangla language that focuses on just the probabilistic generative approach rather than linguistic attributes. The majority of studies utilize the clustering method for the purpose of developing spell checker (Mandal and Hossain, 2017), identifying syntactically similar words (Ismail and Rahman, 2014), grouping documents based on genre (Ahmad et al., 2018), clustering sentences (Husna et al., 2018), speech recognition (Rahman et al., 2010), etc. In all these studies, clustering in Bangla languages has been performed based on n-gram language models rather than advanced word embedding techniques. Ritu et al. (2018) compare the performances of non-contextualized word embeddings and clustering word vectors, but determining topic words by clustering word vectors is yet to be explored in the context of the Bangla language.

3 Methodology

In this section, we describe our proposed approach and implementation details. Figure 1 depicts the overview of our proposed methodology.

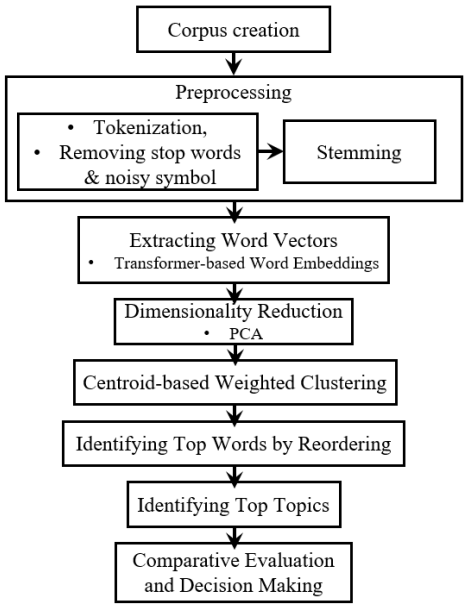

Figure 1: Workflow diagram of our method

3.1 Corpus Creation

We create our corpus of documents or articles by collecting Bangla news articles from two popular online news portals (e.g., prothom alo[1] and bangla tribune[2]). We collect those news articles using a web crawler that crawls the news text from the HTML pages of respective web pages. For this purpose, we utilize the urllib[3] package and BeautifulSoup[4] library of the Python programming language.

Our corpus contains 197,238 news articles or documents (97,073 documents from "Prothom Alo" and 100,165 from "Bangla Tribune") dated from 2020 to 2023. Table #1 depicts the category-based distribution of our collected data. Our corpus comprises articles on health, national, international, crime, sports, entertainment, etc. We develop the corpus so that the number of articles in different categories is approximately balanced. Our corpus includes 4,299,788 sentences and 47,856,640 token words from various fields of context. All other statistical overviews of the corpus have been shown in table #2.

3.2 Preprocessing

News articles often contain noisy token sequences and foreign alphabets or symbols. In the preprocessing phase, we tokenize our documents into

[1] https://www.prothomalo.com
[2] https://www.banglatribune.com
[3] https://docs.python.org/3/library/urllib.html
[4] https://pypi.org/project/beautifulsoup4/

Table 1: Category-based distribution of our corpus

Category	Number of Articles	Category	Number of Articles
National	26262	Economics	11767
International	19661	Health	24624
Crime	16785	Religion	15996
Sport	18263	Opinion	13115
Education	16491	Agriculture	15520
Entertainment	18754	**Total**	**197,238**

Table 2: Statistical overview of our corpus

Parameters	Total Amount
Article/Document	197,238
Sentences	4,299,788
Token Words	47,856,640
Unique Token Words	715,670
Average Sentences per Article	21.8
Average Words per Article	242.63
Average Words per Sentence	11.13

tokens, remove stopwords[5], hashtags, IP addresses, URL links, punctuation, and digits. In order to enhance the quality of our analysis, we choose to exclude tokens (i.e., words) that belong to less than five documents and are found inside lengthy sentences exceeding a length of 50 words (Sia et al., 2020). We also eliminate foreign words or symbols, email artifacts, noisy token sequences, etc. Then, we apply stemming to identify the root words of all sub-words and determine the vector representation of root words by averaging the vector representations of their corresponding sub-words. Sometimes, stemming results in a meaningless root word, but it does not affect much as all the sub-words convert to the same root words. After stemming, we find 715,670 unique words (i.e., vocabulary size) in our corpus.

3.3 Word Embedding Methods

In this phase, we extract the vector representations of the vocabulary words from the documents. We choose pre-trained transformer-based Large Language Models (LLM) for extracting those word vectors. The reason behind selecting LLM-based models is their self-attention mechanism. As a result, these models can capture long-term dependencies in textual data. Again, LLM-based models have been pre-trained utilizing large corpora of text containing various contextual information. So, these models can be utilized for any context. At the same time, due to the pre-training with extensive data, LLM-based models can generate standardized representations of word vectors, which makes those models more scalable. In this study, we choose bloom model[6] with 3 billion parameters as the multilingual model and select several BanglaBERT variants (e.g., BanglaBERT (BBert_bha) by (Bhattacharjee et al., 2022), BanglaBERT (BBert_kow) by (Kowsher et al., 2022), and BanglaBERT (BBert_sag) by (Sarker, 2020)) as Bangla LLMs. Another reason behind choosing pre-trained models is that we need not spend time or resource for training models and getting word vectors.

3.4 Dimnesionality Reduction

After converting vocabulary words into word vectors, we get word vectors with a dimension size of 768. This dimension size is identical for all LLM-based models. Due to the sparsity and redundancy of high-dimensional space, clustering algorithms may perform poorly. So, we apply a dimensionality reduction process, Principle Component Analysis (PCA) (Wold et al., 1987), to reduce the dimension size of word vectors. To determine the appropriate dimension size, we examine multiple values ranging from 100 to 700 with an interval of 100.

3.5 Centroid-based Weighted Clustering

We choose to apply centroid-based clustering techniques. Those techniques offer a logical method for obtaining topic words in each cluster by measuring the distance from the cluster center. Again, previous studies suggest that non-centroid-based hierarchical clustering methods lead to inferior performance and necessitate the adjustment of a significant number of hyperparameters (Sia et al., 2020). We apply several clustering techniques

[5] https://github.com/stopwords-iso/stopwords-bn

[6] https://huggingface.co/bigscience/bloom-3b

like KMeans (KM), Spherical k-means (SKM), k-medoids (KMd), and Gaussian Mixture Models (GMM).

We also consider weighted clustering, as the vector representation of words only holds the semantic feature of words but can not understand the corpus statistics (Rahman, 2020b). But word co-occurrence statistics inside a document or corpus are important for identifying topic words. For weighted clustering, we provide weights to our vocabulary words by measuring- Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF).

$$TF(t,d) = \frac{count\ of\ word,\ t\ in\ document,\ d}{total\ number\ of\ words\ in\ document,\ d}$$

$$IDF(t,D) = log(\frac{Number\ of\ documents\ in\ the\ Corpus,\ D}{Number\ of\ documents\ in\ D\ containing\ t+1} + 1)$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) * IDF(t,D)$$

3.6 Reordering Clustered Words and Identifying Top X Words

After the clustering phase, we do the reordering of the words in each cluster to identify the top X words. Reordering refers to the organization of the clustered words based on some parameters. These parameters can be any statistical or probabilistic values extracted from the corpus. Clustering based on the data points' similarities & weights, along with the reordering, enhances the appropriateness of the highly relevant topic words inside a topic. Thus, a cluster or topic can be described effectively.

In section 3.5, we get our initial top words of individual clusters by measuring the shortest distance or high similarity between data points and respective cluster centers. We also incorporate the weights of the words during clustering. This procedure does not guarantee to find out the underlying original themes of a cluster or topic (Sia et al., 2020). So, we apply reordering of clustered words based on their average TF & $TF\text{-}IDF$ scores inside particular documents (Sia et al., 2020). Thus, we obtain the top X words from each topic or cluster that can effectively describe the respective topic.

3.7 Getting Top Topics of Documents

Top topics are representative of a particular document, which help understand the context of that document. After getting all the clusters/topics and their corresponding top X words, we determine the top topics of documents or the whole corpus. For identifying the most informative clusters from a particular document, we measure the sum of *euclidean distances* between each cluster center and all word vectors of the document. Finally, we normalize the obtained values for each cluster center using the softmax function to get a probability distribution for all clusters or topics. Clusters with low probabilities obtained from normalization are considered the most informative topics.

4 Experimental Setup

In this section, we discuss the implementation details of our approach.

4.1 Performance Metrics

For measuring the performances of topic modeling methods, we use NPMI (Normalized Point-Wise Mutual Information) (Bouma, 2009) that ranges from [-1, 1] where '1' indicates perfect association, '0' denotes statistical independence, and '-1' represents complete negative association. We only choose NPMI as the performance metric because similar previous studies (Sia et al., 2020) measure only NPMI for measuring performances.

NPMI is a statistical process that measures the direction & strength of association between two words or terms. It is the normalized version of PMI. It measured the extent to which the observed co-occurrence of two terms deviates from what would be expected if the terms were statistically independent. NPMI facilitates a more balanced comparison of associations across different word pairs by providing a bounded range. It is beneficial in topic modeling, where understanding semantic relationships is crucial in extracting meaningful insights from textual data. The formula of NPMI has been shown in the equation 1.

$$NPMI(w_i, w_j) = \frac{log(\frac{P(w_i,w_j)}{(P(w_i)*P(w_j))} + \epsilon)}{-log(P(w_i,w_j) + \epsilon)} \quad (1)$$

$P(w_i, w_j)$ refers to the probability of co-occurrence of words w_i and w_j within the topic and $P(w_i)$ & $P(w_j)$ indicates the probability of the occurring of w_i and w_j within the topic respectively. ϵ is a small smoothing factor.

In our study, we evaluate the NPMI scores of all possible word combinations inside each cluster and average the values to get the average NPMI score of each cluster. This is also called the topic coherence (Blair et al., 2020). Equation 2 depicts the formula of *topic coherence* (Coh) of a topic,

t where N represents the number of topic words inside the topic, t.

$$Coh(t) = \frac{2}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} NPMI(w_i, w_j) \quad (2)$$

Again, the "average NPMI score for a document" is evaluated by averaging the topic coherence scores of all clusters inside the document.

4.2 Baseline Model and Parameters

We select LDA as our baseline model as it is the most commonly used probabilistic topic modeling technique (Blei et al., 2003). It considers that documents are mixture of latent topics, and each word within a document is assigned to one of these topics. Thus it identifies latent topics, their related word distributions, and the composition of these topics in documents.

To represent a single cluster or topic, we will identify the top 10 topic words. For determining the number of topics, we tune the value of the number of topics, k, from 2 to 20 based on a high average intra-topic similarity score (i.e., topic coherence in equation # 2).

5 Results

In this section, we analyze our experimental results.

5.1 Computational Cost Measurement

We explain the time complexity of our proposed clustering algorithms as well as word embedding methods compared with the probabilistic topic model, LDA.

5.1.1 Time Complexity Analysis for Clustering Algorithms

Table #3 represents the time complexity measurements of different clustering methods. In the table, t is the maximum number of iterations for the worst case, n represents the vocabulary size or the number of data points, k is the number of clusters, and d is the dimension size of the data.

Table 3: Time complexity of clustering algorithms

Algorithms	Initial	Iteration	Overall
KM	$O(kdn)$	$O(tnkd)$	$O(tnkd)$
GMM	-	$O(tnkd^3)$	$O(tnkd^3)$
KMd	$O(kn)$	$O(tkn^2)$	$O(tkn^2)$
SKM	$O(kdn)$	$O(tnkd)$	$O(tnkd)$

t differs based on the clustering algorithms and word embedding methods, as different algorithms require different numbers of iterations for convergence. However, our study considers the t constant factor, which represents the maximum number of iterations for the worst case. Weighted versions of clustering incur an initial cost for weight initialization and introduce a constant factor for remeasuring the cluster centers. The procedure of reordering introduces an additional time complexity of $O(nlog(n_k))$, where n_k represents the average number of words within a cluster.

On the other hand, it is worth noting that the complexity of LDA using collapsed Gibbs sampling is $O(tkN)$, with N representing the total number of tokens inside documents. Consequently, when the value of N far exceeds the value of n, clustering approaches have the potential to offer more favorable trade-offs between performance and complexity.

5.1.2 Word Embeddings Cost

Since we use pre-trained transformer-based word embedding methods, we do not need to train those models from scratch. We need to generate the word vectors for all vocabulary words. So, the tokenized vocabulary words are passed through the transformer layers. The procedure requires linear time complexity. Again, by defining a batch size, this process can be done simultaneously for all batches.

5.2 Best Performed Word Embedding & Clustering Combination

Table #4 presents the average NPMI scores of all the combinations of our proposed transformer-based word embedding techniques and clustering algorithms for our corpus. In this table, the weighted clustering and reordering have been performed based on Tf-Idf scores as those scores improve the performance significantly by statistics ($p = 0.029$; $\alpha = 0.05$ by t-test) rather than Tf scores (Figure 2).

We observe that BBert_kow outperforms other word embedding methods in all conditions as BBert_kow considers 40GB of Bangla textual data during training, which is approximately 1.5 times the corpus size (29.5 GB) used by BBert_bha. Again, the multi-lingual model (Bloom) underperforms due to the low distribution rate of Bangla (0.5%) in their training corpus.

$KM_{w,r}$ shows the best result with all word embeddings among all the clustering variants, as

Table 4: Average NPMI Scores of different combinations of clustering algorithms and word embedding techniques for our corpus

Models		BBert_bha	BBert_kow	BBert_sag	Bloom
Non-weighted Clustering and No reordering	KM	0.07	0.08	0.07	-0.19
	GMM	0.18	0.21	0.19	-0.13
	KMd	0.05	0.05	0.06	-0.22
	SKM	0.06	0.08	0.06	-0.2
Weighted Clustering	KM_w	0.09	0.11	0.09	-0.17
	GMM_w	0.21	0.22	0.22	-0.09
	KMd_w	0.05	0.07	0.06	-0.21
	SKM_w	0.08	0.09	0.08	-0.19
Reordering	KM_r	0.26	0.29	0.28	0.05
	GMM_r	0.25	0.27	0.26	0.01
	KMd_r	0.24	0.26	0.24	-0.02
	SKM_r	0.26	0.27	0.26	0.01
Weighted Clustering and Reordering	$KM_{w,r}$	**0.29**	**0.31**	**0.29**	**0.09**
	$GMM_{w,r}$	0.26	**0.28**	0.28	0.07
	$KMd_{w,r}$	0.25	**0.27**	0.25	0.05
	$SKM_{w,r}$	0.26	**0.29**	0.29	0.07

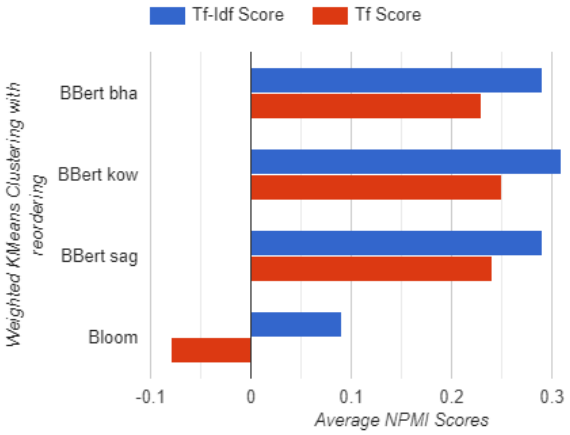

Figure 2: Difference between Tf-Idf and Tf scoring during weighted KMeans clustering and reordering across different word embedding techniques

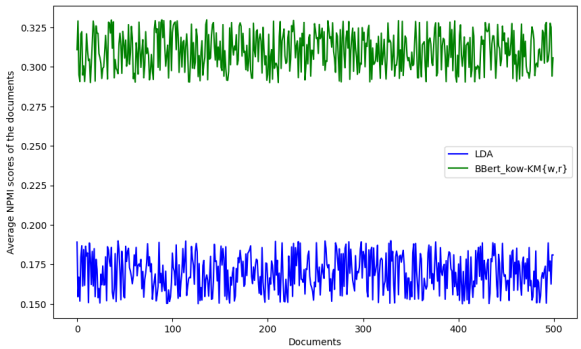

Figure 3: Curves of average NPMI scores for 500 random documents using BBert_kow-KM$_{w,r}$ and LDA methods

KMeans perfectly determines cluster centers by averaging clustered word vectors. Weighted clustering and reordering improve the result that is statistically significant ($p = 0.0002$; $\alpha = 0.01$ for $KM_{w,r}$ vs KM). This implies that corpus statistics is an informative attribute for extracting topics. Thus, we determine BBert_kow-KM$_{w,r}$ as the best-performed combination with the average NPMI score of 0.31.

5.3 Comparison between BBert_kow-KM$_{w,r}$ and LDA

On the whole corpus, BBert_kow-KM$_{w,r}$ (average NPMI score= 0.31) also performs significantly better than the traditional topic model (LDA) (average NPMI score= 0.18). We measure the individual NPMI scores of both BBert_kow-KM$_{w,r}$ and LDA across all documents and find a significant difference with a zero p-value in the t-test between those two topic modeling methods. Figure 3 shows the curves of average NPMI scores for randomly chosen five hundred sample documents using BBert_kow-KM$_{w,r}$ and LDA methods. In the figure, we observe that the curve for LDA ranges from 0.15 to 0.19, and the curve for BBert_kow-KM$_{w,r}$ ranges from 0.29 to 0.33.

LDA is a generative probabilistic procedure where each topic is a distribution over all vocabulary words. In the topic-word distribution predicted from LDA, the probability of a particular word can

be significant for multiple topics. As a result, a common topic word can belong to multiple topics in the LDA method. This scenario will increase the average inter-topic similarity and decrease the average NPMI score of a document.

On the other hand, BBert_kow-KM$_{w,r}$ considers hard clustering where a topic word can be included into a cluster or topic. So, all the topic words of a cluster differ from those of the other cluster. For this reason, the average inter-topic similarity is low, and the average NPMI score is high for the BBert_kow-KM$_{w,r}$ method. Again, we consider the word embedding of transformer-based LLMs in our proposed method. Transformer-based word embeddings are contextualized word embeddings where a word with different meanings is considered in different contexts. As word embedding holds both the semantic and syntactic characteristics of words, the topic coherence of a cluster increases. As a result, the average NPMI score of the corpus becomes high. Furthermore, corpus statistics are also incorporated during weighted clustering and reordering. Hence, we identify that linguistic attributes (i.e., semantic characteristics of documents) and corpus statistics help extract top-topic words.

The execution time of BBert_kow-KM$_{w,r}$ is far less than that of LDA for the whole corpus. For evaluating the entire corpus with 197,343 documents, BBert_kow-KM$_{w,r}$ requires 54 seconds, whereas LDA takes 5 minutes 21 seconds on GPU. This empirical study supports the time complexity measurements in Section 5.1 for both our proposed and baseline methods.

5.4 Weighting

Weighted clustering significantly improves the performance of clustering word embeddings. From table #4, we observe the improvement due to the weighting in all cases of clustering and word embedding without reordering. Similarly, the improvement is also visible for all the instances of clustering and word embedding while considering reordering.

Figure 4 presents that the average NPMI score degrades with increased vocabulary size for the entire corpus if weighting is not considered during clustering. With the increase in vocabulary size, words' semantic characteristics are becoming so sparse. This scenario decreases the NPMI score of two words, making identifying words with high topic coherence challenging. Corpus statistics mitigates this issue as weighted clustering. So,

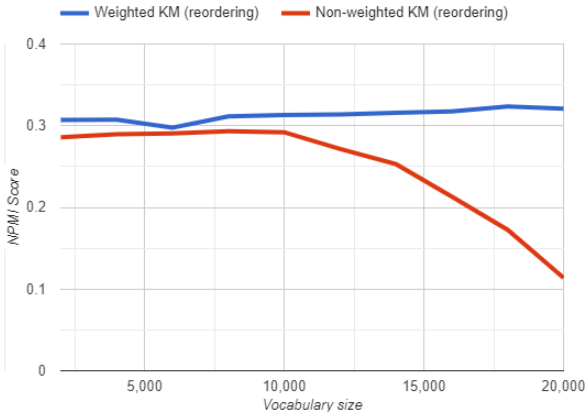

Figure 4: Average NPMI of both weighted and non-weighted clustering with the increase of the vocabulary size for our corpus

weighted clustering based on Tf-Idf is effective for topic analysis from extensive data. The statistical test results for measuring the difference in the performances between weighted clustering and non-weighted clustering are described in appendix A.1.

5.5 Reordering

Reordering cluster words plays an important role in finding top words per topic. From table #4, we observe that reordering enhances the efficacy of the KMeans clustering more significantly than that of GMM. The statistical test results for measuring the difference in the performances between reordered and non-reordered clustering are described in appendix A.2.

5.6 Performance of PCA

To identify the significance of dimensionality reduction, we consider the weighted KMeans clustering with reordering in different types of word vectors. We examine different dimension sizes that range from 100 to 700 with an interval of 100 and measure the average NPMI score of the whole corpus.

From Figure 5, we find the improvements of the average NPMI scores as the dimension size increases. This is due to the fact that high dimensions can capture more information. However, by observing the elbow points, we can determine that the NPMI values get saturated when the dimension size is 300. So, we consider the dimension size of 300 by applying PCA. Our finding of the dimension size also supports the previous study (Rahman, 2020b) for measuring robust and consistent dimension size of word vectors. In the future, we plan to

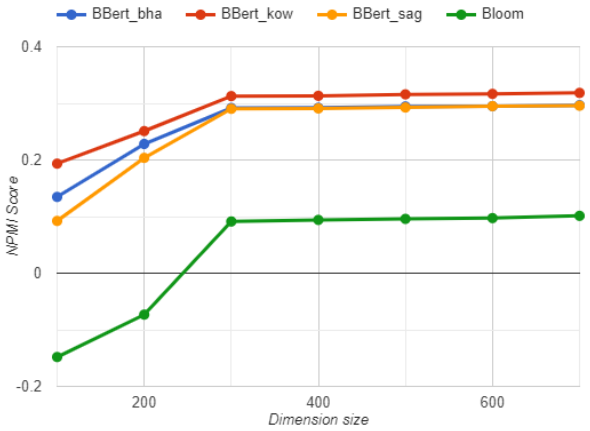

Figure 5: Average NPMI scores of different embedding methods (weighted clustering & reordering) for KMeans with different dimension size for our corpus

apply other neural network-based dimensionality reduction approaches to get better results.

5.7 Qualitative Findings

For the qualitative analysis, we extract the top ten topic words of each topic or cluster from both BBert_kow-$KM_{w,r}$ and LDA. These top topic words describe what the respective topic or cluster actually expresses. We choose top ten topics from both the baseline (i.e., LDA) and our proposed method (i.e., BBert_kow-$KM_{w,r}$). Rather than the real theme exposed by the topics, we determine which genres- the categories that are selected during creating corpus (table #1)- are being covered by those topics. Our findings say that the top ten topics extracted by the BBert_kow-$KM_{w,r}$ technique are different and similar to the ten categories of the corpus. In contrast, LDA retrieve topics from the corpus that represent eight unique genres or categories. The full descriptions of the qualitative results are delineated in appendix B.

6 Limitations

This study has several limitations. First, we use only four language models for word embedding, three of which were Bangla language models. There are also many Bangla LLMs or multi-lingual LLMs with enriched Bangla datasets. These models can be explored in the future. Again, we use pre-trained models to determine word vectors. Integrating dense layers with the LLMs and fine-tuning or pre-training the model for generating word vectors can be a future direction. Second, we only apply centroid-based clustering techniques in our approach. We plan to explore more advanced clustering methods in the future. Another limitation in clustering is the outliers that mislead the position of the cluster centers. A better method to reduce the outliers can positively impact the clustering performance. Third, we only experiment with TF and TF-IDF as corpus statistics. Other informative corpus statistics can be studied in the future. Finally, the neural-network-based dimensionality reduction method can efficiently increase the performance of our proposed topic modeling.

7 Conclusion

Probabilistic generative topic models evaluate word co-occurrences inside documents and ignore linguistic attributes. We apply clustering on word vectors to extract informative topics and compare word embedding techniques and clustering algorithms. By analyzing the whole corpus, the study offers that BanglaBERT (BBert_kow), along with Tf-Idf-based weighted clustering & reordering ($KM_{w,r}$) (average NPMI=0.31 and run-time=54 seconds), outperforms traditional topic models (average NPMI=0.18 and run-time=5 minutes 21 seconds) with respect to top words extraction and time complexity. However, one limitation in clustering is the outliers that mislead the position of cluster centers. We plan to address this issue and perform fine-tuning or pre-training transformer-based word embedding models in the future.

References

Adnan Ahmad, Md Ruhul Amin, and Farida Chowdhury. 2018. Bengali document clustering using word movers distance. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Stuart J Blair, Yaxin Bi, and Maurice D Mulvenna. 2020. Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, 50:138–156.

David Blei, Lawrence Carin, and David Dunson. 2010. Probabilistic topic models. *IEEE signal processing magazine*, 27(6):55–65.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40.

Jordan Boyd-Graber, Yuening Hu, David Mimno, and 1 others. 2017. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296.

Miriam Cha, Youngjune Gwon, and HT Kung. 2017. Language modeling by clustering with word embeddings for text readability assessment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2003–2006.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804.

Guilherme Raiol De Miranda, Rodrigo Pasti, and Leandro Nunes de Castro. 2020. Detecting topics in documents by clustering word vectors. In *Distributed Computing and Artificial Intelligence, 16th International Conference*, pages 235–243. Springer.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Md Hasan, Md Motaher Hossain, Adnan Ahmed, and Mohammad Shahidur Rahman. 2019. Topic modelling: A comparison of the performance of latent dirichlet allocation and lda2vec model on bangla newspaper. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5. IEEE.

MA Helal and Malek Mouhoub. 2018. Topic modelling in bangla language: An lda approach to optimize topics and news classification. *Computer and Information Science*, 11(4):77–83.

Asmaul Husna, Maliha Mostofa, Ayesha Khatun, Jahidul Islam, and Md Mahin. 2018. A framework for word clustering of bangla sentences using higher order n-gram language model. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pages 1–6. IEEE.

Sabir Ismail and M Shahidur Rahman. 2014. Bangla word clustering based on n-gram language model. In *2014 international conference on electrical engineering and information & communication technology*, pages 1–5. IEEE.

Md Kowsher, Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Prianka Mandal and BM Mainul Hossain. 2017. Clustering-based bangla spell checker. In *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 1–6. IEEE.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Md Mijanur Rahman, Md Farukuzzaman Khan, and Mohammad Ali Moni. 2010. Speech recognition front-end for segmenting and clustering continuous bangla speech. *Daffodil International University Journal of Science and Technology*, 5(1):67–72.

Rifat Rahman. 2020a. A benchmark study on machine learning methods using several feature extraction techniques for news genre detection from bangla news articles & titles. In *Proceedings of the 7th International Conference on Networking, Systems and Security*, pages 25–35.

Rifat Rahman. 2020b. Robust and consistent estimation of word embedding for bangla language by fine-tuning word2vec model. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE.

Rifat Rahman, Sheikh Abir Hasan, and Fardous Ahmed Rubel. 2022. Identifying sentiment and recognizing emotion from social media data in bangla language. In *2022 12th International Conference on Electrical and Computer Engineering (ICECE)*, pages 36–39. IEEE.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. *arXiv preprint arXiv:1906.09821*.

Zakia Sultana Ritu, Nafisa Nowshin, Md Mahadi Hasan Nahid, and Sabir Ismail. 2018. Performance analysis of different word embedding models on bangla language. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5. IEEE.

Sagor Sarker. 2020. Banglabert: Bengali mask language model for bengali language understanding.

Motoki Sato, Austin J Brockmeier, Georgios Kontonatsios, Tingting Mu, John Y Goulermas, Jun'ichi Tsujii, and Sophia Ananiadou. 2017. Distributed document and phrase co-embeddings for descriptive

clustering. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 991–1001.

Suzanna Sia, Ayush Dalmia, and Sabrina J Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! *arXiv preprint arXiv:2004.14914*.

Vivek Kumar Rangarajan Sridhar. 2015. Unsupervised topic modeling for short texts using distributed representations of words. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 192–200.

Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. 2019. Clubwords: exploiting semantic word clustering representation for enhanced topic modeling. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 753–761.

Chenguang Wang, Mu Li, and Alexander J Smola. 2019. Language models with transformers. *arXiv preprint arXiv:1904.09408*.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Pengtao Xie and Eric P Xing. 2013. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874*.

He Zhao, Lan Du, and Wray Buntine. 2017. A word embeddings informed focused topic model. In *Asian conference on machine learning*, pages 423–438. PMLR.

A Statistical Test Results for Weighted Clustering and Reordering Cluster Words

A.1 Weighting

Table #5 represents the statistical significant differences between weighted clustering and non-weighted clustering considering all documents. The differences are measured based on the average NPMI scores of documents. We observe significant differences in all the cases when considering reordering. This implies that both the semantic characteristics of words and the corpus statistics play essential roles in determining topic words.

A.2 Reordering

Table #6 depicts the significance of reordering the topic words after weighted clustering. We perform the t-test to measure the difference between reordering and non-reordering. The test is done over all the documents and word embedding methods based on the average NPMI score of documents. In all cases except the Gaussian Mixture Model, reordering causes significant improvement in identifying topic words. This implies that reordering has little impact on weighted GMM as it assumes the mixture of several Gaussian distributions for different clusters where reordering is performed during expectation maximization. Since reordering is performed by default in the GMM, there is no impact of another reordering for extracting topic words.

B Top 10 topics from Corpus & Qualitative discussion

Table #7 and #8 present the topic words for individual topics for BBert_kow-KM$_{w,r}$ and LDA respectively. We find a number of similarities between those two tables while considering the topic words of a particular topic.

From table #7, we identify that the ten topics are mostly similar to the corpus categories. The topic coherence score of the "health" topic is highest. As we collect our corpus for 2020 to 2022 from the archive of the online news portal, the frequency of health-related documents is very high due to the effect of the COVID-19 pandemic. Again, most news articles or documents are relevant to COVID-19. Our findings also reveal that all the topic words under the "Health" topic are related to COVID-19 (table #7). For the same reason, the topic coherence score of this topic is highest. The count of the "National" category news is maximum in our corpus. So, we get the top ten topic words under the "National" topic with a high topic coherence score. The minimum topic coherence score is observed for the "Agriculture" topic. The probable reason behind the low topic coherence score of this topic is the fewer co-occurrences of the topic words inside the agriculture-category documents. The table also displays the topic ranking from top to bottom following the procedure described in the section 3.7. We discover that the "National" topic is the most crucial topic of the corpus. The "National" category is a large domain. We can incorporate new categories (except the "International" category) as the subdomain of the "National" category because all these incidents happen in a nation. So, news of all categories except the "International" category can be considered the "National" category. As a result, the sum of the distances of all the word vectors with respect to the centroid of the "National"

Table 5: t-test result for average NPMI values between non-weighted and weighted clustering considering all individual documents and all word embedding methods

Comparison (without Reordering)	P-value $\alpha = 0.01$	Comparison (with Reordering)	P-value $\alpha = 0.01$
KM Vs KM_w	0.001	KM_r Vs $KM_{w,r}$	0.0001
GMM Vs GMM_w	0.003	GMM_r Vs $GMM_{w,r}$	0.004
KMd Vs KMd_w	0.001	KMd_r Vs $KMd_{w,r}$	0.001
SKM Vs SKM_w	0.002	SKM_r Vs $SKM_{w,r}$	0.001

Table 6: t-test result for average NPMI values between non-reordered and reordered clustering considering all individual documents and all word embedding methods

Comparison (without weighting)	P-value $\alpha = 0.01$	Comparison (with weighting)	P-value $\alpha = 0.01$
KM Vs KM_r	0.0001	KM_w Vs $KM_{w,r}$	0.0003
GMM Vs GMM_r	0.027	GMM_w Vs $GMM_{w,r}$	0.043
KMd Vs KMd_r	0.0005	KMd_w Vs $KMd_{w,r}$	0.0007
SKM Vs SKM_r	0.0003	SKM_w Vs $SKM_{w,r}$	0.0004

Table 7: Topic coherence score of each topic or cluster obtained from BBert_kow-KM$_{w,r}$

Top 10 topic words for each topic	Topic Coherence score	Topic Name
Politics; Election; Corruption; Hasina; Padma-Bridge; Digital-Bangladesh; Governance; Freedom-Fighters; Awami-League; Dhaka	0.373	National
COVID-19; Vaccine; Lockdown; Delta-Variant; Quarantine; Hospital; Health-Workers; ICU; Mask; Telemedicine	0.385	Health
Rohingya; Myanmar; China; India; UN; Diplomacy; Climate-Change; Trade; SAARC; Refugees	0.334	International
Cinema; OTT Platforms; Dhallywood; Music; Celebrity; Drama; Festival; Television; YouTube; Fashion	0.297	Entertainment
Cricket; BPL; Football; Olympics; Tamim; Shakib; Sports Ministry; Dhaka-League; BCB; World-Cup	0.379	Sport
Rape; Murder; Cybercrime; Trafficking; Corruption; Drug; Robbery; Violence; Police; Arrest	0.318	Crime
Online-Classes; University; HSC; Primary-Education; Scholarship; E-Learning; Reopening; Education-Policy; Ministry-Education; SSC	0.272	Education
GDP; Inflation; Remittance; Export; RMG; Unemployment; Economic-Growth; Budget; SME; Banking	0.257	Economics
Rice; Farmer; Crop; Subsidy; Fisheries; Livestock; Irrigation; Agricultural-Policy; Food-Security; Agrarian-Reform	0.239	Agriculture
Islam; Eid; Mosque; Hindu; Puja; Religious-Freedom; Fatwa; Harmony; Zakat; Madrasa	0.269	Religion

topic or cluster is the lowest, and the "National" topic becomes the most informative topic of our corpus.

Our proposed topic modeling applies KMeans clustering, which is a hard clustering technique. So, the topic words under a topic cannot be in another

Table 8: Topic coherence score of each topic or cluster obtained from LDA

Top 10 topic words for each topic	Topic Coherence score	Topic Name
Advancement; Private; Limited; Institution; Online-classes; MS; Education; Professor; e-Learning; University	0.173	Education
Promise; Human; Sheikh; Bangabadhu; Country; Prime-Minister; Nation; Hasina; Bangladesh; Politics	0.217	National
Secretary; Awami-League; President; Zilla; Sheikh; Chairman; Parliament; Election; Leader; BNP	0.253	National
Corona; Health; Death; Health-Complex; Identification; Sample; Healthy; Hospital; Infection; Treatment	0.238	Health
Police; Arrest; Rescue; Hospital; Union; Corruption; Police-station; OC; Dead-body; Injured	0.158	Crime
Bangladesh; Institution; Dhaka; Bank; Percent; Financial; Limited; Government; Loan; Development	0.147	Economics
Money; Price; Sale; Food; Kilogram; Market; Transaction; Rice; Budget; Onion	0.154	Economics
Cricket; Bangladesh; Test-match; Field; Match; Sport; Captain; Coach; Club; Football	0.177	Sport
August; Complaint; Case; Money; Court; Investigation; Arrest; Lawyer; Mission; Rape	0.162	Crime
India; US; President; China; World; Refugees; International; Organization; Saudi-Arab; UN	0.157	International

topic, and the topics or clusters cannot overlap. This is the main reason for the diversity characteristic of the topics, and we can extract diverse themes from a document or corpus. Another interesting finding (table #7) is that the ten topics determined from our proposed topic modeling are identical to the categories of our corpus except the "Opinion" category. As the news under the "opinion" category are the reflection of various national issues, BBert_kow-KM$_{w,r}$ does not consider this category as a separate cluster.

In table #8, we observe topic names, their topic coherence scores, and the top ten topic words of each topic obtained from the state-of-the-art topic modeling method, LDA. The top two topics with high coherence scores are "Health" and "National", similar to the finding from table #7. LDA fails to extract diverse topics from the corpus, and we can observe some common words (i.e., money, hospital, etc.) that belong to multiple topics. As LDA predicts the topic-word distribution, a common word can have a high probability value across various topics. It reduces diversity in topics. There are eight unique topics in the table #8 where we extract ten topics from the corpus. The "Entertainment" and "Agriculture" topics are missing when we apply LDA. Some studies (Das et al., 2015) argue that top informative topics from LDA can be identified by their coherence scores. A topic with the highest coherence score is considered the top informative topic. So, the "Health" topic is the most informative topic obtained from LDA. It also supports the result from BBert_kow-KM$_{w,r}$. As the COVID-19 pandemic broke out in early 2020 and continued till 2022, the "Health" topic was highly concerned then.

So, we can decide that our proposed topic modeling method, BBert_kow-KM$_{w,r}$ qualitatively outperforms LDA in understanding the insights of a document or corpus. Again, it can unsupervisedly group the vocabulary words of the corpus into some clusters that are similar to the categories of the corpus.