Analyzing Clustering Algorithms for Topic Modelling

Manmeet Kaur Baxi
Student ID: 1150990
mbaxi@lakeheadu.ca

Abstract

Document clustering and topic modelling methods on short-texts from various social media platforms help in categorizing and annotating large amounts of user generated content. Many techniques have been developed over the years for document clustering, which involves text mining, topic modelling, latent semantic representations, and neural embedding approaches. However, many of these perform poorly on short-texts (i.e., documents with a length less than 250 characters), and often the results are not comparable across studies.

In this work, I evaluate several techniques for document embeddings and clustering on the AG News dataset. Four different feature representations including Term Frequency-Inverse Document Frequency (TF-IDF), doc2vec, Bag of Words (BoW) and Sentence BERT are combined with four clustering techniques, i.e., k-means, Latent Dirichlet Allocation (LDA), Latent Semantic Indexing (LSI), and BERTopic to benchmark the dataset. Three different evaluation measures have been used to measure the performance of the topic clustering algorithms on news description, and the most appropriate extrinsic measure has been suggested for evaluation.

The results show that corpus based embedding: Bag of Words (BoW) outperforms the others, however, comparable results have been portrayed by transformer based embeddings: Sentence BERT, taking a fraction of time as compared to the top performer, Bag of Words (BoW).

1 Introduction

Document clustering helps organize similar documents together by using machine learning techniques. Various methods have been proposed for topic modelling and document clustering (Maitri P Naik, Harshadkumar B Prajapati, and Vipul K Dabhi 2015a). These techniques typically involve the use of a feature matrix to represent a corpus, with a clustering method applied to this matrix. Recently, neural word embeddings have been applied to social media data as they can generate dense representations with semantic properties and require less manual preprocessing than the traditional methods (Li et al. 2017). Common clustering methods applied in this context uncover the hidden patterns on social media platforms (Irfan et al. 2015).

Topic Modelling is an active research area, which involves uncovering patterns of word usage in the documents (Chinnov et al. 2015). Topic modelling is used to cluster documents by giving a probability distribution over documents, similar to document clustering. Common clustering techniques include Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003), Author-topic model (Rosen-Zvi et al. 2012), and recent dynamic topic models (Jähnichen et al. 2018). The content of social media platforms is best suited for applying clustering techniques due to its' nature of a large amount of unorganized texts (Steinskog, Therkelsen, and Gambäck 2017). Additional features, like demographics, user information have been also used to semantically cluster similar documents. (Crockett et al. 2017)

There are two key challenges with topic modelling of social media platforms. First, failing to reproduce the results, since the data is not publicly available. Also, over time the associated tweets are removed from the platform, which results into complicated data collection and preparation as well as the restrictions by the social media platforms themselves. Second, the absence of a single evaluation metric for measuring the clustering performance across different methods (Stieglitz et al. 2018).

In this work, I have tried to evaluate the performance of four different document embeddings (i.e., Term Frequency-Inverse Document Frequency (TF-IDF), doc2vec, Bag of Words (BoW) and Sentence BERT) with four clustering techniques (i.e., k-means, Latent Dirichlet Allocation (LDA), Latent Semantic Indexing (LSI), and BERTopic) on the AG News dataset over three evaluation measures, namely Adjusted Random Index (ARI), Adjusted Mutual Information (AMI), and Normalized Mutual Information (NMI).

The structure of the paper is as follows: In section 2, I have discussed the related work from other researchers done until now. Section 3 discusses the dataset, preprocessing, feature representations and clustering methods, and the evaluation measures used for the analysis. Section 4 discusses about the results and section 5 highlights the limitations and future work.

2 Literature Review

The literature on document clustering and topic modelling of short-texts is organized into two areas. Firstly, the stud-

ies using topic modelling and clustering approaches to identify topics in short-texts to understand online texts have been discussed. Secondly, the recent advancements, like, neural word embedding models and deep learning based methods to generate dense feature representations have been discussed.

2.1 Document clustering and topic modelling

Various studies have shown that topic modelling has been applied in different fields, to understand the unlabelled data as in the works of (Ding et al. 2020, Brookes and McEnery 2019, Asmussen and Møller 2019). Recently, researchers have focused more on understanding the topics being portrayed in short-texts on various social media platforms, specifically Twitter (Arın, Erpam, and Saygın 2018, Montenegro et al. 2018, Curiskis et al. 2020).

Topic models are statistical models frequently used for uncovering hidden semantic structures in documents. They help in discovering abstract topics from a text. Document clustering refers to methods based on cluster analysis to group similar documents together according to some feature matrix. The short size and noisy nature of social media data, such as Twitter data, facilitates the application of document clustering techniques instead of traditional topic models (Nugroho et al. 2020). Nevertheless, applying topic models to find topics from social media platforms has been an active area of research (Likhitha, Harish, and Kumar 2019). Indeed the term 'topic discovery', refers to either 'topic modelling' or 'document clustering'.

Vector space representations of word documents have been typically used in document clustering methods. In the Bag of Words (BoW) model, each document is represented as a space in the point of words. Each word is a feature or dimension of this space, with element values assigned in one of several ways. These can be one-hot-encodings, where the value is set to 1 if the word exists in the document and 0 otherwise, term frequency, or term-frequency inverse-document-frequency calculations. Given that the total dimension size is the number of unique words, often there is a threshold cut-off to use only those words with high values (Patki and Khot 2017). A range of clustering algorithms may then be applied to the feature matrix, such as k-means, hierarchical clustering, self-organising maps, and so on (Maitri P. Naik, Harshadkumar B. Prajapati, and Vipul K. Dabhi 2015b).

For instance, Montenegro et al. 2018 generated topic clusters on the Twitter dataset of Dumaguete City with 99,942 tweets using LDA and further analyzed the sentiment of each cluster using a supervised machine learning algorithm, Support Vector Machine (SVM). Ayo et al. 2021 developed a probabilistic clustering model for hate speech classification on Twitter. They generated features using Term Frequency-Inverse Document Frequency (TF-IDF) model and enhanced with topics inferred by a Bayes classifier. Then, a rule-based clustering method was used to automatically classify real-time tweets into the correct topic clusters. Further, they applied fuzzy logic for hate speech classification using semantic fuzzy rules and a score computation module.

Rashaideh et al. 2020 proposed a grey wolf optimizer for document clustering. They used the average distance of documents to the cluster centroid (ADDC) as an objective function to repeatedly optimize the distance between the clusters of the documents. The algorithm was tested on six publicly available datasets with 55% of the documents being correctly clustered with a high level of accuracy.

2.2 Recent advancements

A lot of literature on clustering documents used tf-idf representations on tweets at some level. These matrices treat terms as one-hot encoded vectors, where each term is represented by a binary vector with exactly one non-zero element. This means that relationships between words, such as synonyms, are not incorporated and the resulting document matrix representation is sparse and high dimensional. The concept of dense, distributional representations of words, or word embeddings, provide an alternative approach. In these methods, each word is represented by a real valued vector of fixed dimension. Word embeddings are commonly trained using neural network language models, such as word2vec (Mikolov et al. 2013), doc2vec (Le and Mikolov 2014). However, when using word embedding models to create document level representations, the word vectors need to be aggregated in some way. Common approaches in the literature are to simply take the mean of the word vectors for all terms in the document, or to concatenate the vectors to a document vector of fixed size X. Yang, Macdonald, and Ounis 2018.

There have been works to capture the lexical, semantic and syntactic features from the text, like the work by Park et al. 2019 developed an algorithm to leverage semantically and syntactically meaningful features from text for clustering. They first extract features from pre-trained language models and initialize cluster centroids to spread out uniformly. They also utilize cross entropy loss partially, as the self-training scheme can be biased when parameters in the model are inaccurate. Their approach outperformed other available models on four datasets.

The work by Arın, Erpam, and Saygın 2018 provides an interactive application for document clustering using lexical and semantic clustering phase-wise. For the lexical clustering of tweets, they use Longest Common Subsequence as a similarity metric, and to overcome the challenge of the large corpora sizes, they have implemented a suffix tree based index structure in their platform to efficiently cluster tweets. They promise comparable clustering quality in a fraction of the time required by the baseline methods which use Longest Common Subsequence and Suffix Tree. Evidently, their method outperforms the state-of-the-art techniques in terms of time (clustering 60K tweets in 23.8-25.5 seconds).

Convolutional Neural Networks have also been leveraged to incorporate more useful semantic features and learn non-biased deep text representation in an unsupervised manner as in the work of Xu et al. 2017. The word embeddings are explored and fed into convolutional neural networks to learn deep feature representations, meanwhile the output units are used to fit the pre-trained binary codes in the training process. Finally, they apply k-means to cluster the learned representations on three short-text datasets, which outperforms

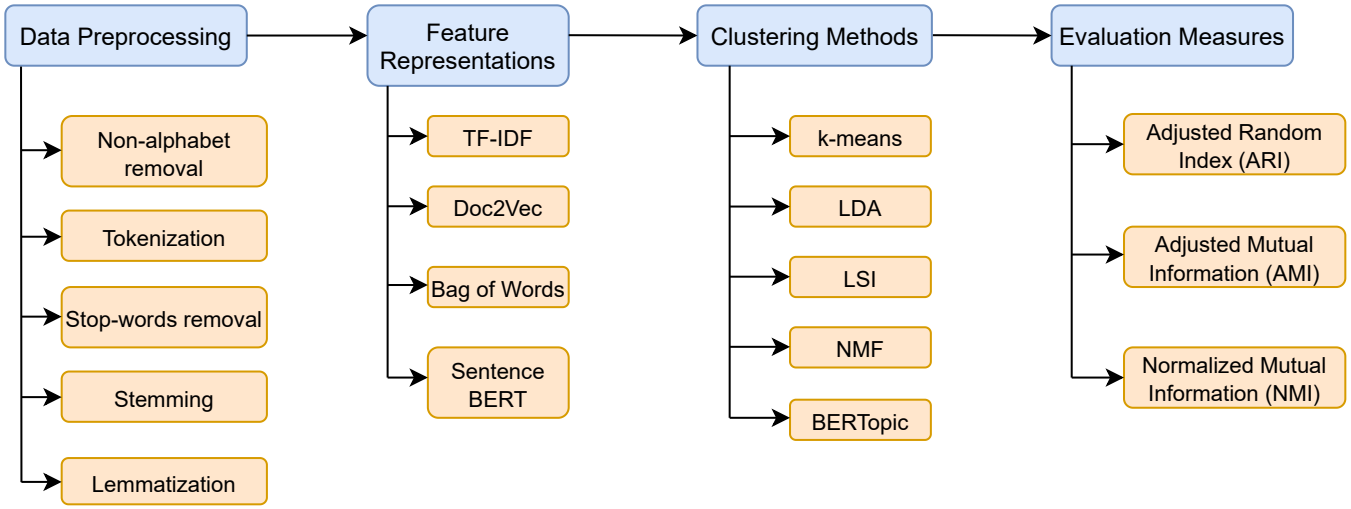


Figure 1: Analysis pipeline

several popular clustering methods.

Recently, Hamm et al. 2021 tried using topological considerations about the co-occurrence of networks of terms to discover topics on a text corpora published by the European Commission over the last decade.

In addition to the document clustering and topic modelling approaches discussed so far, a new series of deep learning based clustering methods have been developed (Min et al. 2018, Su et al. 2021). Many of these techniques use deep neural networks to learn feature representations trained at the same time as clustering. Examples include several deep autoencoder networks with a clustering layer, where the loss function is a combination of reconstruction loss and clustering loss. Clustering methods based on generative models such as Variational Autoencoders and Generative Adversarial Networks look promising from a document clustering perspective since they can also generate representative samples from the clusters.

Many approaches for document clustering and topic modelling have been proposed for online social media data.

3 Methods

In this section, the dataset, data preprocessing steps, embedding representations and clustering algorithms, and the evaluation measures used are discussed. Figure 1 provides an overview of all the steps in the analysis.

3.1 Dataset

The AG's corpus of news articles was obtained from the web[1]. It contains 496,835 categorized news articles from more than 2000 news sources. The four largest classes (i.e., 'Business', 'Sci/Tech', 'Sports', 'World') were chosen from this corpus to construct the dataset, using only title and description fields. The number of training samples for each class is 30,000 and testing 1900, totaling to 120,000 training and 7600 testing samples. The average length of news description is 193 characters.

3.2 Data Preprocessing

The analysis reported was done using *python 3.7.12*. For preprocessing the text, firstly, all the non-alphabets (numbers, punctuation, new-line characters and extra spaces) were removed from the text using the regular expression module (*re 2.2.1*). Then, the text was tokenized using *nltk 3.2.5*, followed by the removal of stopwords. Also, very small words (i.e., words with length less than 3 characters) were removed from the text. This was followed by stemming the text using *PorterStemmer* and lemmatizing it using the *WordNetLemmatizer* from *nltk*.

3.3 Computational Resources

The analysis was done using Google's online Colaboratory (or, Colab) service[2]. The computational resources provided by the Colab environment were as listed below:

- **CPU:** 2X Intel Xeon CPU @ 2.20GHz
- **Memory (RAM):** 13 GB
- **GPU:** Tesla T4 @ 15 GB

3.4 Embedding Representations

In this study, the performance of four embedding representation methods is evaluated combined with four commonly used clustering algorithms. I have also included the BERTopic model in a separate category since it is a recently published model used for topic modelling (Grootendorst 2020). TF-IDF, doc2vec, and Bag of Words (BoW) feature representations were generated using *gensim 4.1.2*.[3]

[1] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[2] https://colab.research.Google.com/

[3] https://radimrehurek.com/gensim/

For TF-IDF, the *tf-idf* matrix was limited to top 1000 terms per document since no performance improvement was noticed even after adding more terms.

A *doc2vec* model is a neural network trained to create a dense vector with fixed dimension for each document in a corpus (Le and Mikolov 2014). The *doc2vec* model was trained with 100 dimensions, a minimum word count of 40, and an initial learning rate of 0.025, dropping linearly at a rate of 0.001 using distributed memory.

For the Bag of Words(BoW) representation, first, the news description is processed to create a *(token,key)* mapping using Gensim's corpora dictionary[4], where only the first 20,000 frequent tokens are kept. Then, the tokens from each news description are converted into a BoW corpus.

Sentence BERT, presented in (Reimers and Gurevych 2019), aims to adapt the BERT architecture by using siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. I use the model '*all-MiniLM-L6-v2*' from the existing set of sentence-transformer models to map news descriptions into a 384 dimensional dense vector space.

3.5 Clustering Algorithms

For the clustering algorithms, four techniques used in the literature (Y. Chen et al. 2019, Maitri P Naik, Harshadkumar B Prajapati, and Vipul K Dabhi 2015a) were selected, which have given comparable results on the dataset used for analysis. Each of the clustering algorithms were trained on an optimal number of epochs found after training the models 20 times varying across different random states, as discussed in Section 4.1. Firstly, I applied a k-means clustering algorithm using the Euclidean metric over optimal number of epochs for each of the embedding representations (Section 4.1). The algorithm was run multiple times over the data with varying random seeds.

I used Latent Dirichlet Allocation (LDA) multi-core provided by Gensim[5], which uses all CPU cores to parallelize and speed up model training. The LDA model was trained with 7 workers, 1,000 chunk size, evaluated every 10 seconds for each of the embedding representations and trained for their respective number of optimal epochs (discussed in Section 4.1) with varying random states.

Latent Semantic Indexing (LSI) model[6] implements fast truncated SVD (Singular Value Decomposition), and was also trained with a chunk size of 1,000 for each of the embedding representations.

Non-negative matrix factorization (NMF) model[7] uses the online NMF proposed by Zhao and Tan 2016 for large corpora. The NMF model was trained on a chunk size of 1,000, evaluated every 10 seconds, with a gradient descent step size (i.e., kappa) 1, normalizing the corpus by filtering out documents with smaller probabilities (i.e., minimum probability is set to 0), across varying random states and optimal number of epochs for each of the embedding representations (Section 4.1).

BERTopic is a topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions (Grootendorst 2020). BERTopic was trained in a supervised fashion with the Sentence BERT embeddings.

3.6 Evaluation Measures

There are two types of measures for evaluating document clustering algorithms, intrinsic and extrinsic. Instrinsic measures do not require a ground truth label and measure the variation within and between the clusters. These methods evaluate a clustering technique based on the compactness and cohesion of a cluster. As these methods are dependent on the embedding representations used while clustering, so, they do not give comparable results for methods which use different feature sets, and hence, are not used as an evaluation measure for the analysis. Extrinsic measures, however, can be compared across methods but require a ground truth label. Accuracy, Precision, Recall and F1 are the common extrinsic measures (Maitri P Naik, Harshadkumar B Prajapati, and Vipul K Dabhi 2015a), but these are dependent on the ordering of the clustering labels to ground truth labels which is a problem for large corpora. Mutual information and Random index are appropriate measures for this analysis as they are independent of the absolute label values.

Mutual Information (MI) is the measure of mutual dependence between two random variables. It quantifies the reduction in uncertainity about one discrete random variable given the information about another. Higher MI indicates lower levels of uncertainity and better label predictions. For two discrete random variables *X* and *Y* with a joint probability distribution *p(X,Y)*, the mutual information, *MI(X,Y)* is given by:

$$MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{1}$$

Normalized Mutual Information (NMI) is a normalization of the the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation). This measure is useful to compare results across different methods, however, it does not account for chance. NMI is given as:

$$NMI(X,Y) = \frac{MI(X,Y)}{\sqrt{H(X)H(Y)}}, \tag{2}$$

where, *H(X)* and *H(Y)* are the marginal entropies of the discrete random variables *X* and *Y*. The marginal entropy for a variable *X* is given by:

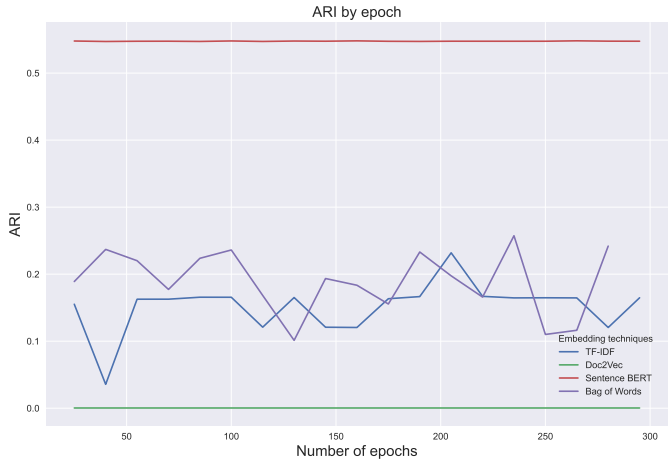
$$H(X) = -\sum_{i=1}^{n} p(x_i) \log(p(x_i)) \tag{3}$$

[4] https://radimrehurek.com/gensim/corpora/dictionary.html

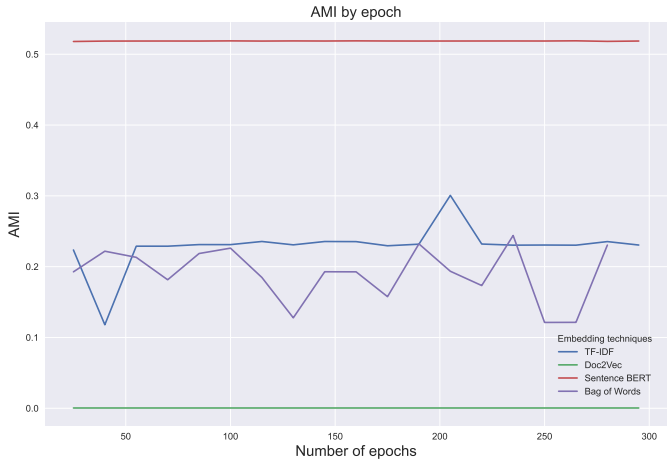
[5] https://radimrehurek.com/gensim/models/ldamulticore.html

[6] https://radimrehurek.com/gensim/models/lsimodel.html

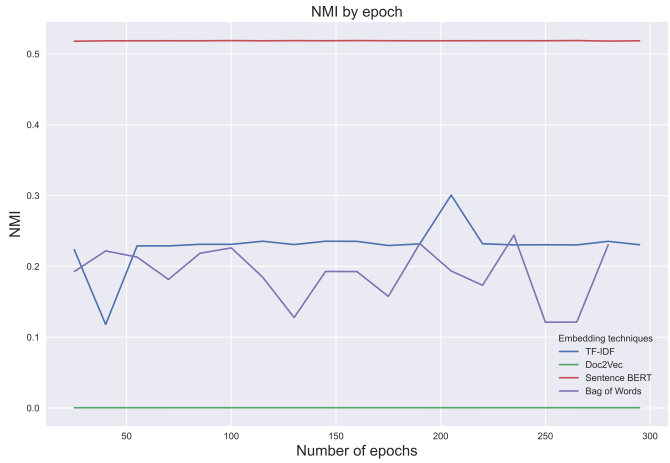
[7] https://radimrehurek.com/gensim/models/nmf.html



(a) ARI by epoch



(b) AMI by epoch



(c) NMI by epoch

Figure 2: Plot of three evaluation measures, i.e., (a) Adjusted Random Index (ARI), (b) Adjusted Mutual Information (AMI), and (c) Normalized Mutual Information (NMI) (vertical axes) by epoch (horizontal axes) for 20 runs for TF-IDF, doc2vec, Bag of Words (BoW), and, Sentence BERT representations on AG News dataset using k-means clustering.

The Random Index (RI) computes a similarity measure between two clustering algorithms by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusters. The value of RI also varies between 0 and 1, where 0 indicates a random and 1 indicates identical labels. Given a set of elements $S = \{o_1, o_2, ...o_n\}$ and two partitions of S to compare, $X = \{X_1, ..., X_r\}$ and $Y = \{Y_1, ..., Y_s\}$, the Random Index represents the frequency of times the partitions X and Y are in agreement over the total number of observation pairs. Mathematically the RI, is given by:

$$RI(X, Y) = \frac{a + b}{a + b + c + d}, \tag{4}$$

where, a represents the numbers of pairs in S that are in the same subset in X and the same subset in Y, and b represents the number of pairs of elements in S that are in different subsets of X and different subsets of Y. Values a and b together give the number of times the partitions are in agreement. The value c represents the number of pairs of elements in S that are in the same subset of X and different subsets of Y, and d gives the number of pairs of elements in S that are in different subsets of X and the same subset of Y.

For extrinsic clustering evaluation measures to be useful for comparison across methods and studies, such measures need a fixed bound and a constant baseline value. Both NMI, and RI are scaled to have values between 0 and 1, so, they satisfy the first condition. However, it has been shown that both measures increase monotonically with the number of labels, even with an arbitrary cluster assignment (Vinh, Epps, and Bailey 2010). This is because both the mutual information and Random index do not have a constant baseline, implying that these measures are not comparable across clustering methods with different numbers of clusters. To ac-

count for this, adjusted versions of the MI and RI have been proposed. The Adjusted Random Index (ARI), adjusts the RI by its expected value, and is given by:

$$ARI(X,Y) = \frac{RI(X,Y) - E\{RI(X,Y)\}}{max\{RI(X,Y)\} - E\{RI(X,Y)\}}, \quad (5)$$

where, $E\{RI(X,Y)\}$ is the expected value of $RI(X,Y)$. The ARI is thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical (up to a permutation).

Adjusted Mutual Information (AMI) is used and is given by:

$$AMI(X,Y) = \frac{MI(X,Y) - E\{MI(X,Y)\}}{max\{H(X),H(Y)\} - E\{MI(X,Y)\}}, \quad (6)$$

where, $E\{MI(X,Y)\}$ is the expected value of mutual information for the two discrete variables (Vinh, Epps, and Bailey 2010). AMI and ARI are the best measures to ensure a comparable evaluation. However, it is needed to check how these compare to each other. By developing theory regarding generalised information theoretic measures, Romano et al. 2016 concluded that the AMI is the preferable measure when the labels are unbalanced and there are small clusters, while the ARI should be used when the labels have large and similarly sized volumes.

In this analysis, I report AMI, NMI and ARI measures. Many previous studies have reported NMI measure, so for comparison purposes, I have included it in the evaluation. For the given data and methods of this study, it is likely that the ARI is more appropriate than the AMI as the distribution of documents across labels is balanced. AMI is still included, since it is interesting to see how much the results may differ from the NMI.

Due to the short and noisy nature of the data used in this study, I have examined the effect of different random seeds on performance. I ran each method 20 times with different random seeds, calculated the mean of the NMI, AMI and ARI, and plotted the distributions of these measures.

4 Results

In this section, first the results on optimal number of epochs for all the embedding representations, i.e., TF-IDF, doc2vec, Bag of Words (BoW) and Sentence BERT are described. Then, the performance is evaluated on the different clustering algorithms, i.e., k-means, Latent Dirichlet Allocation (LDA), Latent Semantic Indexing (LSI), Non-negative Matrix Factorisation (NMF), and BERTopic.

4.1 Optimal training epochs for embedding representations

Number of epochs is the key hyper-parameter for training a neural network. Large number of epochs can lead to overfitting the data, however, a lower number of epochs can lead to under-fitting and bad performance. First, the performance change of the mean TF-IDF, doc2vec, Bag of Words (BoW) and Sentence BERT models were explored with the number of epochs. I used k-means clustering as it gave the best results for the embedding representations. For each epoch value between 25 and 300, with increments of 15, I trained the models 20 times using different random seeds and evaluated against the ground truth labels of the dataset. Table 1 summarizes the optimal number of epochs for each of the embeddings.

The doc2vec and Sentence BERT embeddings show no significant improvement when increasing the number of epochs for all the three evaluation metrics, ARI, AMI and NMI; with SentenceBERT having the highest and doc2vec having the lowest values. Sentence BERT generally delivered better performance with higher number of epochs, a maximum value of 265; and doc2vec gave better results at an early stage of 40 epochs. However, TF-IDF and Bag of Words (BoW) portray a completely different behaviour with their performances. For TF-IDF, the performance first saw a sudden drop and then started increasing around 60 epochs. The peak performance for TF-IDF was noticed at 205 epochs. Bag of Words (BoW) had a more negative graph with the performance constantly seeing gains and drops at a regular interval of 100 epochs, however, the best performance for Bag of Words (BoW) is closer towards the higher end of the range, i.e. 235 epochs.

In general, Sentence BERT required the maximum number of training epochs, followed by Bag of Words (BoW), TF-IDF and doc2vec.

Document Embedding	**Optimal number of epochs**
TF-IDF	205
doc2vec	40
Bag of Words (BoW)	235
Sentence BERT	265

Table 1: Optimal training epochs for document embeddings

4.2 Performance evaluation with clustering algorithms

In this section, the mean evaluation measures for the four feature representations with the clustering methods are discussed. Table 2 provides the mean for each of the three evaluation measures and the CPU time taken to train and test the models. I have set the optimal number of epochs as per the findings in the previous section (4.1), i.e. 205 epochs for TF-IDF, 40 epochs for doc2vec, 235 epochs for Bag of Words (BoW), and 265 epochs for Sentence BERT.

The doc2vec model, Bag of Words (BoW) and Sentence BERT embeddings used for this analysis did not support testing clustering algorithms apart from the ones mentioned in the Table 2, and hence, potentially limit the explicit comparison.

Clearly, Bag of Words (BoW) embedding with LDA clustering outperformed the others on all three evaluation measures, followed by TF-IDF embedding with LDA clustering, and Sentence BERT with k-means clustering algorithm. It is noticeable that out of the top three performers, TF-IDF embedding with LDA clustering takes the maximum amount of

Document Embeddings	Clustering Algorithms	ARI	AMI	NMI	CPU Time (min:sec)
TF-IDF	k-means	0.152854	0.227686	0.228108	00:15
	LDA	0.527628	0.509607	0.509818	38:17
	LSI	0.001938	0.039898	0.040358	00:26
	NMF	0.390923	0.402022	0.402300	18:26
doc2vec	k-means	0.000064	0.000070	0.000498	00:13
Bag of Words (BoW)	LDA	**0.530627**	**0.516904**	**0.517112**	32:56
	LSI	0.187384	0.208299	0.208655	00:10
	NMF	0.170421	0.172203	0.172589	29:06
Sentence BERT	k-means	0.519152	0.499055	0.499270	00:42
	BERTopic	0.064128	0.137070	0.137593	08:28

Table 2: Performance evaluation of the feature representations and clustering algorithms on AG News dataset with Adjusted Random Index (ARI), Adjusted Mutual Information (AMI), Normalized Mutual Information (NMI) measures, and CPU Time

time to train and test the topic clustering followed by Bag of Words (BoW) embedding with LDA clustering, and Sentence BERT with k-means clustering taking the least amount of time.

Some methods had a relatively large drop in score between the NMI and AMI measures, indicating that the chance adjustment of the AMI is important. The TF-IDF representation is the most effected by this. For instance, the tf-idf matrix with LDA clustering gave a high NMI of 0.5098, well ahead of the doc2vec and Sentence BERT methods, but an AMI of 0.5096. Comparatively, doc2vec and the Bag of Words (BoW) methods had smaller drops. As discussed earlier, the AMI and ARI are more appropriate evaluation measures than NMI due to their adjustment for chance. On this data set, the ARI is more appropriate as the topics are equally distributed among the news description. The Bag of Words (BoW) representation with LDA clustering, therefore, far outperformed the other methods.

5 Conclusion and Future Work

In this study, I have analyzed the performance of different document embeddings with clustering algorithms on AG News dataset. The results demonstrate that document embeddings and clustering techniques can be effectively used for document clustering. Bag of Words (BoW) and LDA outperformed the traditional TF-IDF based approaches. Also, Sentence BERT performs reasonably well with the k-means clustering algorithm and can be used instead of the top performing Bag of Words (BoW) and LDA combination, because it takes the least amount of time among the top-3.

I plan to extend this work, by inculcating more dimensionality reduction techniques, like, Principal Component Analysis (PCA), Factor Analysis (FA), and t-distributed Stochastic Neighbour Embedding (t-SNE) before passing the corpora to clustering algorithms. Given the adequate computational resources, I would also try to extend the work by using distributed doc2vec embedding for more flexibility with clustering algorithms, and supervised document embeddings, like, Universal Sentence Encoder (USE). Additionally, deep learning based clustering algorithms as discussed by (Su et al. 2021) and parallel clustering algorithms surveyed by (Dafir, Lamari, and Slaoui 2021) may be applied to deliver improved feature representations or document clusterings. Word and document embeddings may also be used as pre-trained initial layers in deep clustering and topic modelling techniques.

6 Acknowledgements

Sincere thanks to Dr. Rahman and Dr. Mago for the course materials, guidance throughout the semester, and for giving an extension to submit the work. This work was inspired by one of the review comments I received from the submission made to ICWSM'21 and tries to do an ablation analysis on "The reason behind choosing a particular document embedding and topic clustering technique for short-texts". Also, the work by Curiskis et al. 2020 helped in understanding the detailed reasons and implementing it on my preferred dataset.

7 Appendix

The dataset, code, and images used in this analysis are available on the Github Repository: link.

References

Arın, İnanç, Mert Kemal Erpam, and Yücel Saygın (2018). "I-TWEC: Interactive clustering tool for Twitter". In: *Expert Systems with Applications* 96, pp. 1–13.

Asmussen, Claus Boye and Charles Møller (2019). "Smart literature review: a practical topic modelling approach to exploratory literature review". In: *Journal of Big Data* 6.1, pp. 1–18.

Ayo, Femi Emmanuel et al. (2021). "A probabilistic clustering model for hate speech classification in twitter". In: *Expert Systems with Applications* 173, p. 114762.

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

Brookes, Gavin and Tony McEnery (2019). "The utility of topic modelling for discourse studies: A critical evaluation". In: *Discourse Studies* 21.1, pp. 3–21.

Chen, Yong et al. (2019). "Experimental explorations on short text topic mining between LDA and NMF based Schemes". In: *Knowledge-Based Systems* 163, pp. 1–13.

Chinnov, Andrey et al. (2015). "An overview of topic discovery in Twitter communication through social media analytics". In.

Crockett, KA et al. (2017). "Cluster analysis of twitter data: A review of algorithms". In: *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*. Vol. 2. Science and Technology Publications (SCITEPRESS)/Springer Books, pp. 239–249.

Curiskis, Stephan A et al. (2020). "An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit". In: *Information Processing & Management* 57.2, p. 102034.

Dafir, Zineb, Yasmine Lamari, and Said Chah Slaoui (2021). "A survey on parallel clustering algorithms for big data". In: *Artificial Intelligence Review* 54.4, pp. 2411–2443.

Ding, Kai et al. (2020). "Employing structural topic modelling to explore perceived service quality attributes in Airbnb accommodation". In: *International Journal of Hospitality Management* 91, p. 102676.

Grootendorst, Maarten (2020). *BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics.* Version v0.9.4. DOI: 10.5281/zenodo.4381785. URL: https://doi.org/10.5281/zenodo.4381785.

Hamm, Andreas et al. (2021). "TeCoMiner: Topic Discovery Through Term Community Detection". In: *arXiv preprint arXiv:2103.12882*.

Irfan, Rizwana et al. (2015). "A survey on text mining in social networks". In: *The Knowledge Engineering Review* 30.2, pp. 157–170.

Jähnichen, Patrick et al. (2018). "Scalable generalized dynamic topic models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1427–1435.

Le, Quoc and Tomas Mikolov (2014). "Distributed representations of sentences and documents". In: *International conference on machine learning*. PMLR, pp. 1188–1196.

Li, Quanzhi et al. (2017). "Data sets: Word embeddings learned from tweets and general data". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 11. 1.

Likhitha, S, BS Harish, and HM Keerthi Kumar (2019). "A detailed survey on topic modeling for document and short text data". In: *International Journal of Computer Applications* 178.39, pp. 1–9.

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Min, Erxue et al. (2018). "A survey of clustering with deep learning: From the perspective of network architecture". In: *IEEE Access* 6, pp. 39501–39514.

Montenegro, Chuchi et al. (2018). "Using latent dirichlet allocation for topic modeling and document clustering of dumaguete city twitter dataset". In: *Proceedings of the 2018 International Conference on Computing and Data Engineering*, pp. 1–5.

Naik, Maitri P, Harshadkumar B Prajapati, and Vipul K Dabhi (2015a). "A survey on semantic document clustering". In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, pp. 1–10.

– (2015b). "A survey on semantic document clustering". In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–10. DOI: 10.1109/ICECCT.2015.7226036.

Nugroho, Robertus et al. (2020). "A survey of recent methods on deriving topics from Twitter: algorithm to evaluation". In: *Knowledge and Information Systems* 62.7, pp. 2485–2519.

Park, Jinuk et al. (2019). "ADC: Advanced document clustering using contextualized representations". In: *Expert Systems with Applications* 137, pp. 157–166.

Patki, U and DP Khot (2017). "A Literature Review on Text Document Clustering Algorithms used in Text Mining". In: *Journal of Engineering Computers and Applied Sciences* 6.10, pp. 16–20.

Rashaideh, Hasan et al. (2020). "A grey wolf optimizer for text document clustering". In: *Journal of Intelligent Systems* 29.1, pp. 814–830.

Reimers, Nils and Iryna Gurevych (2019). "Sentence-bert: Sentence embeddings using siamese bert-networks". In: *arXiv preprint arXiv:1908.10084*.

Romano, Simone et al. (2016). "Adjusting for chance clustering comparison measures". In: *The Journal of Machine Learning Research* 17.1, pp. 4635–4666.

Rosen-Zvi, Michal et al. (2012). "The author-topic model for authors and documents". In: *arXiv preprint arXiv:1207.4169*.

Steinskog, Asbjørn, Jonas Therkelsen, and Björn Gambäck (2017). "Twitter topic modeling by tweet aggregation". In: *Proceedings of the 21st nordic conference on computational linguistics*, pp. 77–86.

Stieglitz, Stefan et al. (2018). "Social media analytics–Challenges in topic discovery, data collection, and data preparation". In: *International journal of information management* 39, pp. 156–168.

Su, Xing et al. (2021). "A Comprehensive Survey on Community Detection with Deep Learning". In: *arXiv preprint arXiv:2105.12584*.

Vinh, Nguyen Xuan, Julien Epps, and James Bailey (2010). "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance". In: *The Journal of Machine Learning Research* 11, pp. 2837–2854.

Xu, Jiaming et al. (2017). "Self-taught convolutional neural networks for short text clustering". In: *Neural Networks* 88, pp. 22–31.

Yang, Xiao, Craig Macdonald, and Iadh Ounis (2018). "Using word embeddings in twitter election classification". In: *Information Retrieval Journal* 21.2, pp. 183–207.

Zhao, Renbo and Vincent YF Tan (2016). "Online nonnegative matrix factorization with outliers". In: *IEEE Transactions on Signal Processing* 65.3, pp. 555–570.