# Document Clustering and Visualisation

Aditya Deepak Bhat
ASU ID: 1222133796
ASU Email: abhat31@asu.edu
Arizona State University

Aniruddha Bhowmik
ASU ID: 1223096615
ASU Email: abhowmi5@asu.edu
Arizona State University

Fenil Bharat Madlani
ASU ID: 1222149747
ASU Email: fmadlani@asu.edu
Arizona State University

Ishan Srivastava
ASU ID: 1219537111
ASU Email: isrivas2@asu.edu
Arizona State University

Pawan Wagh
ASU ID: 1219432396
ASU Email: pwagh2@asu.edu
Arizona State University

Shivam Malviya
ASU ID: 1222318565
ASU Email: smalviy2@asu.edu
Arizona State University

***Abstract*—This project aims to explore and implement various clustering, and visualization techniques on textual documents. State-of-the-art algorithms to cluster documents will be applied on news data sets and results will be visualized using Uniform Manifold Approximation and Projection (UMAP). Sentence Embeddings will be generated for the text using the Universal Sentence Encoder. For clustering these documents, techniques like K-Means, HDBSCAN and LDA (Latent Dirichlet Allocation) will be used on the generated embedding vectors. The proposed solution clusters similar documents based on the embedding generated and provides a graphical visualization for these articles. At last, Sentiment Analysis is also done using VADER and the results are visualized.**

***Index Terms*—Latent Dirichlet Allocation (LDA), Uniform Manifold Approximation and Projection (UMAP), HDBSCAN, Document Clustering, VADER, Universal Sentence Encoder, Document Visualization.**

## I. Introduction

Clustering is dealt with by human beings in every aspect of life, from the neural activity in the brain onto how it recognises patterns to actually clustering physical data for ease of reproduction and computation. It is no wonder that clustering has been the subject of active research in several fields such as statistics, pattern recognition, and machine learning. In data mining, clustering deals with very large data sets with different data attributes. Because of this the clustering algorithms have many impositions onto their performance. A variety of algorithms have recently emerged and have been successfully applied to real life data mining problems.

With the recent advances in deep learning, the ability of algorithms to analyse text has improved considerably. Creative use of advanced artificial intelligence techniques can be an effective tool for analysing the sentiments of an individual from textual data.

## II. Problem Statement

This project can be summed up to find solutions to the following problems:

- Explore various clustering, and visualization techniques for textual documents.
- Perform clustering on these documents and create visualizations for the generated clusters.
- Perform sentiment analysis within each category to detect positive and negative sentiments.
- Visualize these documents to see if there is any relation between the categories and the sentiments.

## III. Related Works

The two most common document clustering techniques that have been used in history are Agglomerative hierarchical clustering and K-means. Agglomerative hierarchical clustering is often portrayed as “better” than K-means, although slower. A widely known study, discussed in [1], indicated that agglomerative hierarchical clustering is superior to K-means, although these results were not on textual data. Within the textual domain, Scatter/Gather [2], a document browsing system based on clustering, used a hybrid approach involving both K-means and agglomerative hierarchical clustering.

Due to its simplicity and high performance the bag of words model has always been preferred in literature and is one of the mose widely used feature model for all kinds of text classification tasks. The model represents the text to be classified as a bag or collection of individual words with no link of one word with the other, i.e. it completely disregards grammar and order of words within the text. This model is also very popular in sentiment analysis and has been used by various researchers. This is a very simplifying assumption but it has been shown to provide rather good performance. There are three ways of using prior polarity of words as features. The simpler un-supervised approach is to use publicly available online lexicons/dictionaries which map a word to its prior polarity.

Some of the earliest work in this field classified text only as positive or negative, assuming that all the data provided is subjective (for example in [3] and [4])

## IV. System Architecture & Algorithms

There are five major modules of the system that have been developed. Modules include Data Cleaning and Preprocessing, Data Embedding, Clustering, Sentiment Analysis and Visualization. Overview of the system is shown in Fig. 1.

Functionality of each listed module and algorithms which are used in each stage are as follows.

### A. Data Cleaning and Preprocessing

Before converting news article data into embedding vectors, data has to be cleaned up and preprocessed. This module performs cleanup operations on the news data set. On Each of the textual documents, data is read and noise characters from data are removed. Text is converted into lower case letters. Normalization is performed on this text. Any stop words which are present in the data are removed and lemmatization is performed on data.

### B. Data Embedding

Operations such as text classification and clustering require the data into vector format. When text embeddings are generated, the vector should capture context of the sentence. Universal Sentence encoder is used to convert the text into higher dimensional vectors. Google's Universal Sentence Encoder [5] is used to convert text into embedding vectors. Fixed dimensional vector is generated by a universal sentence encoder for a given input document.

### C. Clustering

Clustering is an operation which groups similar documents together. Latent Dirichlet Allocation (LDA), K-Means and HDBScan algorithms are used on embedded data vectors for clustering. Metrics including Homogeneity and completeness are collected for each of the algorithms which are used in later stages of evaluation. Each of the algorithms is described below.

- Latent Dirichlet Allocation (LDA)
  LDA [6] is an unsupervised learning technique. Any hidden relations in data can be identified by LDA. Each of the generated groups are known as topics. Dirichlet Distributions are used to identify these groups in data obtained after the embedding stage. Words in text can be affiliated with a document as well as with a topic. These are two different distributions in LDA. This algorithm requires a number of topics as input. It uses frequency of words to determine the topic for the document.
- K-Means
  K-means [7] is partition based approach used to find clusters in the data. K-means begins with selecting random points in space as centroids and then performing recalculations on each iteration until iteration count is reached or there is no change in centroids position. Number of clusters which are needed is given as input to the K-means algorithm. data point is assigned to the nearest cluster in K-Means.
- HDBScan
  HDBScan is also known as Hierarchical Density-based Spatial Clustering of Applications with Noise [8]. This algorithm works well even if data is unclean. Densities are estimated by this algorithm. High Density regions are picked and points are combined for the region.K-th nearest point distance is used as a core measure for picking high density areas. Threshold is picked so that clusters can be identified within the density landscape created by the algorithm. Local clusters can modify thresholds for the region.

### D. Visualization

This module creates visualization for the high dimensional data produced by clustering algorithms. t-distributed stochastic neighbor(t-SNE) [9] and Uniform manifold approximation and projection(UMAP) [10] are used to visualize the data set in this module. Both algorithms are described below.

- t-SNE
  t-SNE is known as t-distributed stochastic neighbor [9] and widely applied to perform visualization of higher dimensional data. Dimensionality reduction is applied by t-SNE algorithm and lower dimension can preserve the local structure of the data. This algorithm is based on a non-linear reduction technique. t-SNE is a compute-heavy algorithm. It finds similar data points to identify clusters and uses. Parameterization can affect the visual clusters generated by t-SNE [10].
- Uniform Manifold Approximation and Projection (UMap)
  Umap can be used on non-linear data for dimensionality reduction. Preprocessing is not required for this algorithm. It is one of the alternatives to t-SNE for creating visualizations of the data. Manifold structure is identified by Umap and neighbor graph is created. For a given manifold, a lower dimensional representation is found by UMap by minimizing the cost function. UMao is more performant in terms of speed as compared to t-SNE algorithm [10]

### E. Sentiment Analysis

Vader (Valence Aware Dictionary and sentiment Reasoner) [11] is used for sentiment analysis of data generated at the embedding state. It is a rule based analysis tool and lexical database is used in Vader. Sentiment analysis is performed on the entire data set as well as within a cluster. Vader can provide information about whether sentiment is positive or negative as well as their degree. Both data are visualized using the UMap algorithm in this module.

## V. Dataset

The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date. [12]

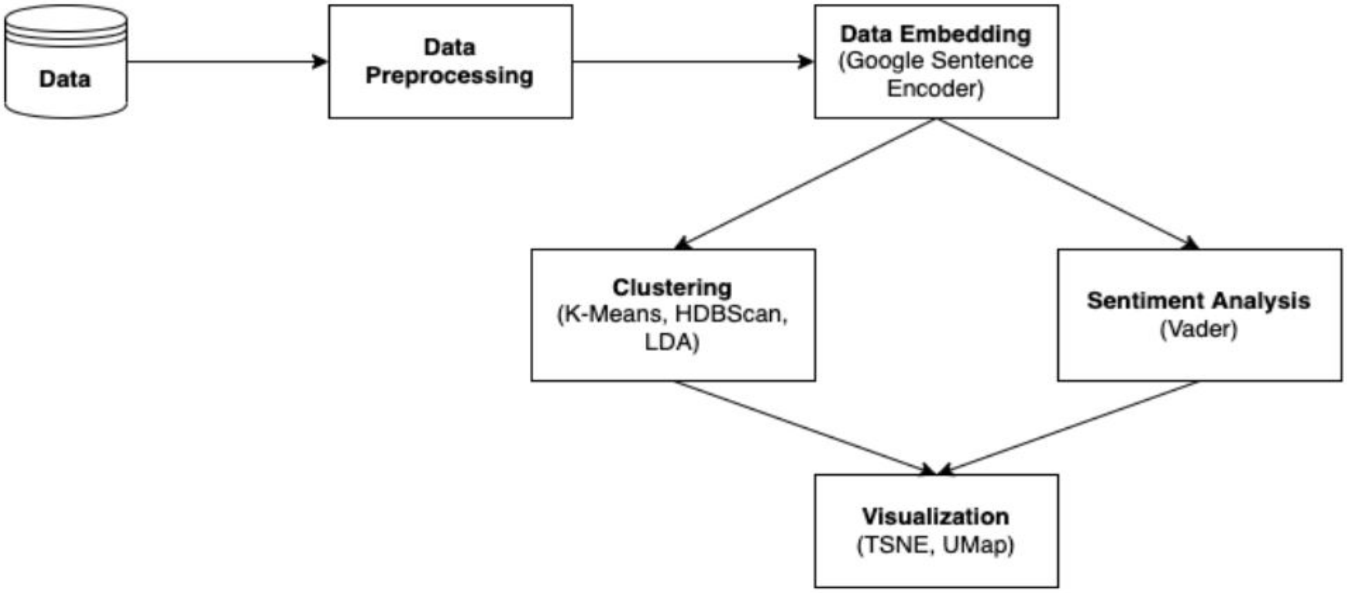
Data
Data Preprocessing
Data Embedding (Google Sentence Encoder)
Clustering (K-Means, HDBScan, LDA)
Sentiment Analysis (Vader)
Visualization (TSNE, UMap)

Fig. 1. System Architecture

### A. Preprocessing

The textual news data is preprocessed and following operations were done on data.

- Noise Removal
- Converting to Lowercase
- Normalization
- Stop word removal
- Lemmatization

Google's Universal Sentence Encoder is used to convert text into embedding vector. For given input, it gives fixed 512 dimensional embedding vector.

## VI. Evaluations (Metrics, Experiments, Findings)

We have used the following metrics for evaluation of Clustering:

- Homogeneity: Measures how much the samples in a cluster are similar. This score is useful to check whether the clustering algorithm meets an important requirement: a cluster should contain only samples belonging to a single class.
- Completeness: Measures how many similar samples are put together by the clustering algorithm. An entirely complete clustering is one where each cluster has information that directs a place toward a similar class cluster. Completeness portrays the closeness of the clustering algorithm to this perfection.
- V-measure: Harmonic mean between homogeneity and completeness.Finally to obtain a measure of the goodness of our clustering algorithm we can consider the harmonic average between homogeneity and completeness and obtain the V-measure or Normalised Mutual Information.
- Adjusted Rand-Index: Measures the similarity between two classifications of the same objects by the proportions of agreements between the two partitions.
- Silhouette Coefficient: Used to calculate the goodness of a clustering technique. Range : -1 to 1.The Silhouette Coefficient is defined for each sample and is composed of two scores: The mean distance between a sample and all other points in the same class. The mean distance between a sample and all other points in the next nearest cluster.

### A. Findings

Using the above mentioned metrics we were able to infer that LDA was best clustering algorithm for this dataset. The results of our findings are given in Table I.

## VII. Visualizaion

This section describes the visualization results achieved during experimentation on the dataset.

### A. Document Visualization

As mentioned in Sec. IV, we have used UMAP and t-SNE to visualize the dataset. To achieve this, we apply these algorithms over the 512 dimensional sentence embedding vectors to reduce its dimensions to either 2 or 3 (for 2-D and 3-D plots respectively). Fig. 2 and Fig. 3 show the 2-D and 3-D visualizations of the dataset respectively, generated from applying UMAP over the sentence embedding vectors. Fig. 4 shows the 2-D visualization from t-SNE.

As we can see from Fig. 2, the sentence embedding vectors are able to capture a sense of the topic, as similar categories of news articles are closer to each other. For example, all news articles related to *talk.politics*, *talk.religion* and *soc.relegion.christian* are close to each other on the top

TABLE I
CLUSTERING EVALUATIONS

| Algorithm | Homogeneity | Completeness | V-measure | Adjusted Rand-Index | Silhouette Coefficient |
|---|---|---|---|---|---|
| KMeans | 0.462 | 0.472 | 0.467 | 0.310 | 0.012 |
| Gaussian Mixture Model | 0.392 | 0.403 | 0.397 | 0.235 | 0.021 |
| Agglomerative Clustering | 0.379 | 0.396 | 0.387 | 0.206 | 0.005 |
| Birch | 0.387 | 0.409 | 0.398 | 0.218 | 0.006 |
| HDBScan | 0.563 | 0.268 | 0.363 | 0.002 | -0.190 |
| LDA | 0.583 | 0.584 | 0.584 | 0.491 | 0.010 |

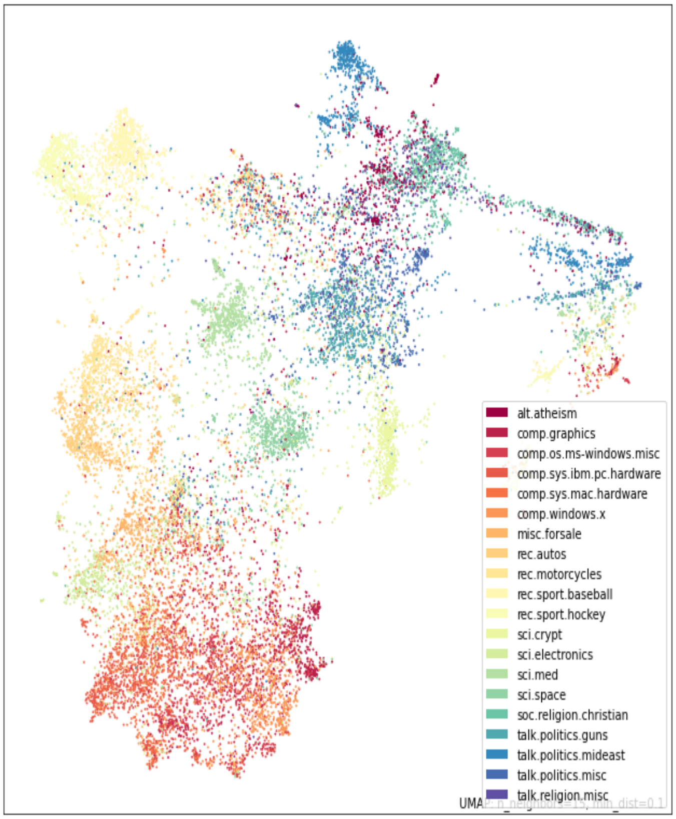

Fig. 2. Visualization of the dataset using UMAP.

of the visualization. Similarly categories related to computers are close to each other at the bottom, those related to sports are close to each other at the top left, etc.

### B. Clustering

To visualize the clustering results, we first obtain the cluster label predictions for each document by training the clustering algorithm using the 512-dimensional sentence embedding vectors. We then use the 2-D or 3-D UMAP projections of these vectors to visualize the clusters. Fig. shows a sample result of clustering using the K-Means clustering algorithm with $k$ = *20* (as there are 20 categories in the dataset).

### C. Sentiment Analysis

To visualize the sentiment of each news article in the dataset, we first obtain the sentiment prediction (i.e. positive

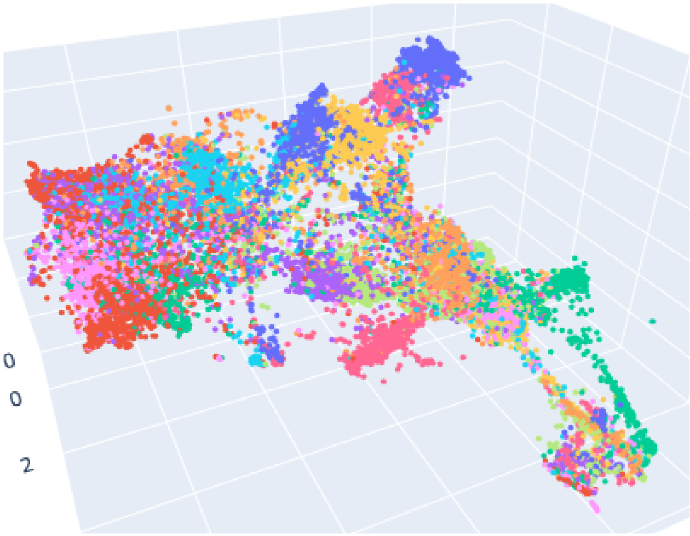

Fig. 3. Visualization of the dataset using UMAP.

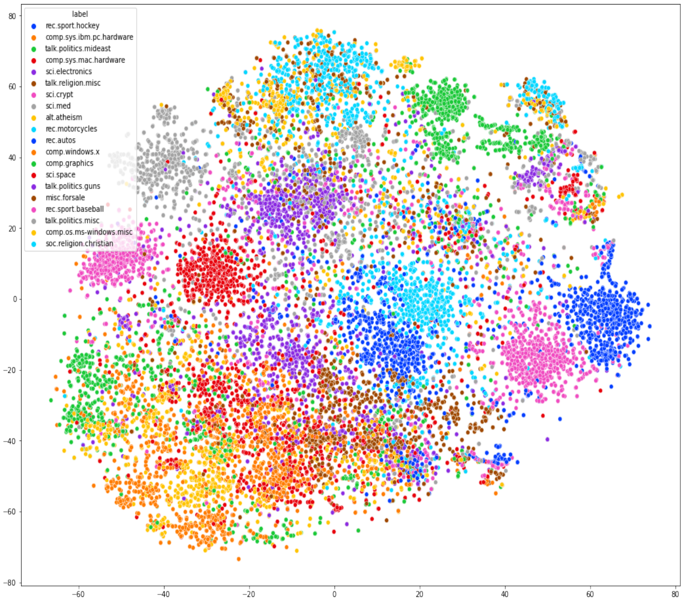

Fig. 4. Visualization of the dataset using t-SNE.

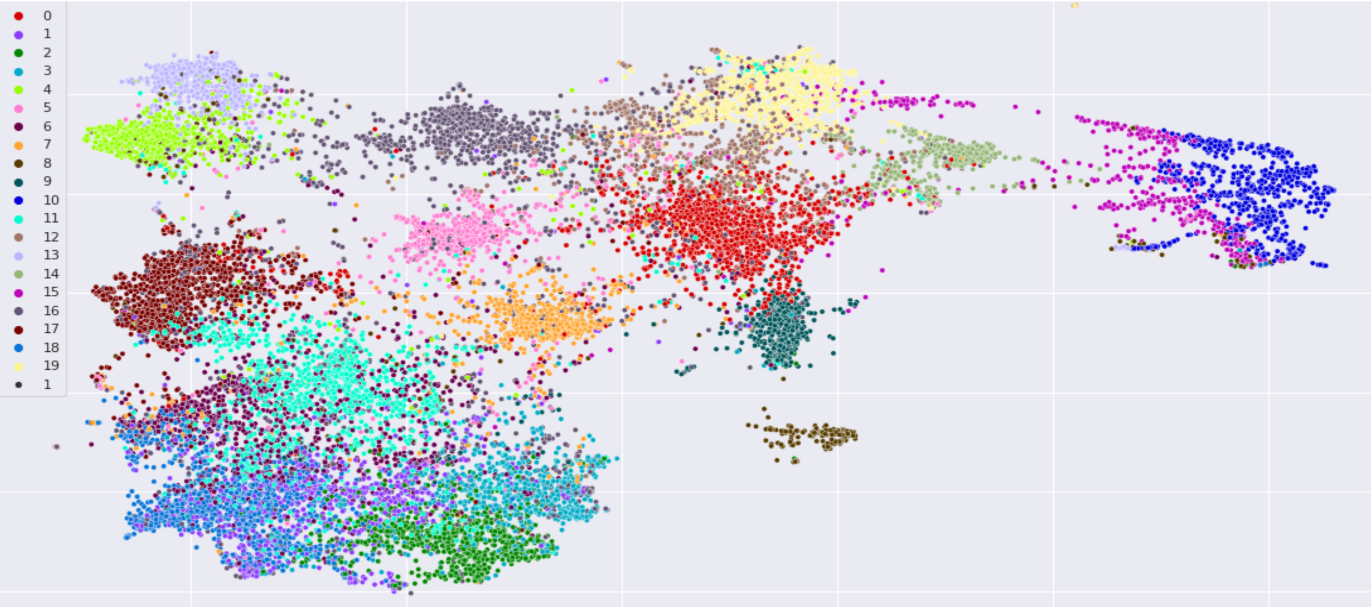

Fig. 5. Visualization of K-Means Clustering on the dataset using UMAP

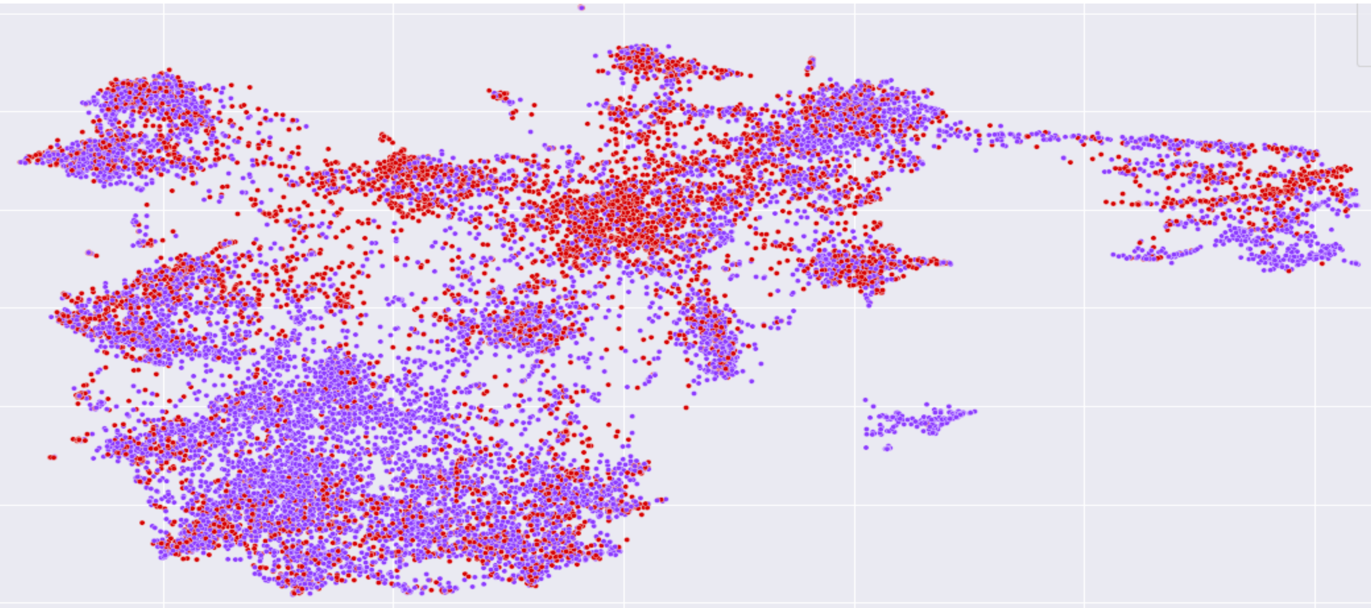

Fig. 6. Visualization of Sentiment Analysis on the dataset using UMAP. Color Map: Red - Negative, Purple - Positive.

or negative) for each document using VADER on the 512-dimensional sentence embedding vectors. We then use the 2-D or 3-D UMAP projections of these vectors to visualize the sentiment of these articles. Fig. shows the visualization of the sentiment in the dataset. From the visualization, we can observe that there are few areas which are having higher negative sentiment and vice versa.

When we compare the regions which have higher concentration of positive or negative sentiment, we can see that these correspond to regions related to *politics* (negative) and *forsale* (positive). Fig. 7 and Fig. 9 show the 2-D and 3-D visualizations respectively of the sentiment in the *misc.forsale* category. We can observe that there is a higher concentration of positive sentiment in the overall category. However, there is also a small region of news articles having negative sentiment.

Similarly, Fig. 8 and Fig. 10 show the 2-D and 3-D visualizations respectively of the sentiment in the *talk.politics.guns* category. We can observe that there is a higher concentration of negative sentiment in the overall category. However, there is also a small region of news articles having positive sentiment.

From the above two cases, we can see that the generated sentence embeddings are not only able to capture the sense of

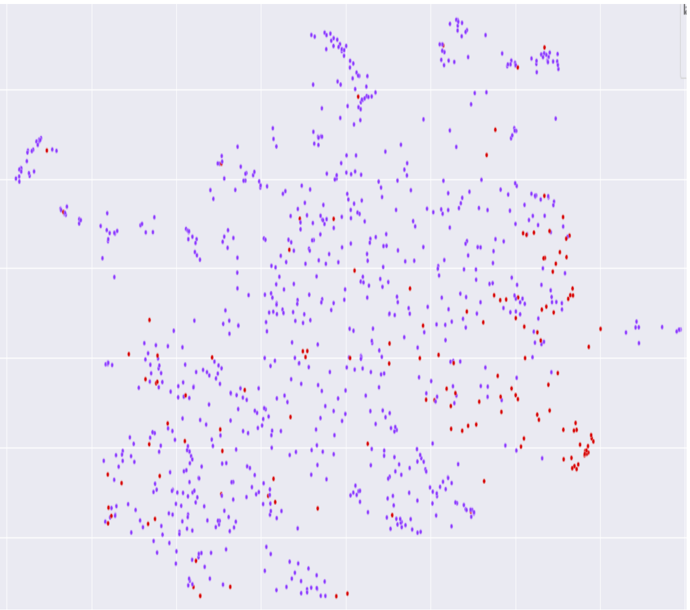

Fig. 7. 2-D Visualization of sentiment in the *misc.forsale* category of the dataset using UMAP. Color Map: Red - Negative, Blue - Positive.

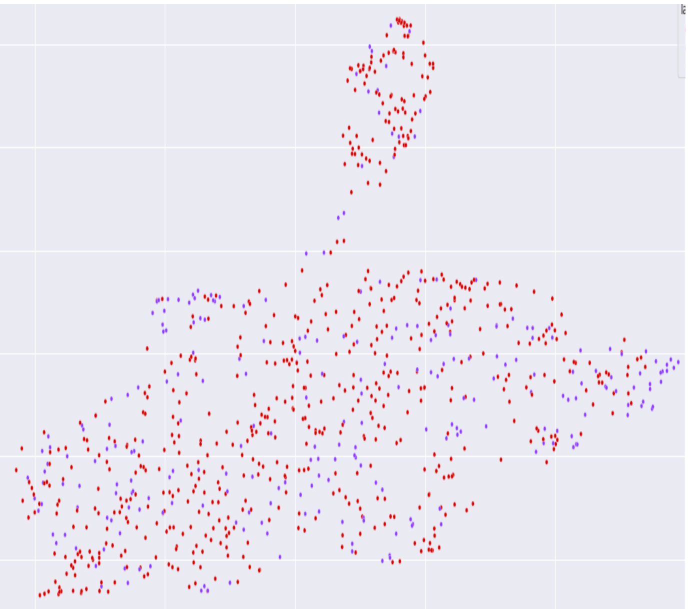

Fig. 8. 2-D Visualization of sentiment in the *talk.politics.guns* category of the dataset using UMAP. Color Map: Red - Negative, Blue - Positive.

similar categories in the dataset, but also able to differentiate between the positive and negative sentiment in each of the categories.

## VIII. Division of Work and Team Members' Contributions

The division of work is summarized in Table II.

The contribution of each team members are outlined below:

- **Background Study and Data Collection**: This part was done by cooperatively by all the team members.

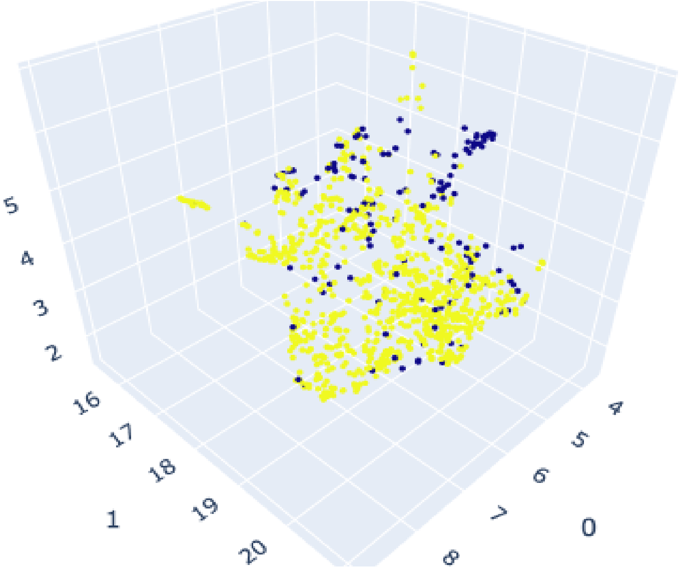

Fig. 9. 3-D Visualization of sentiment in the *misc.forsale* category of the dataset using UMAP. Color Map: Blue - Negative, Yellow - Positive.

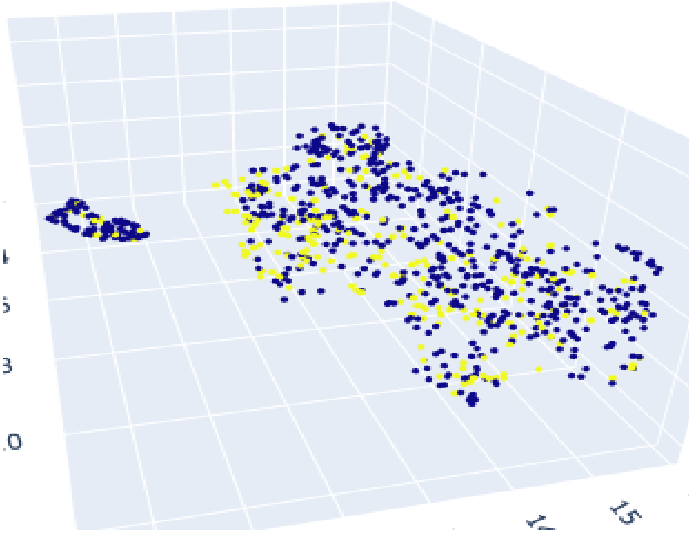

Fig. 10. 3-D Visualization of sentiment in the *talk.politics.guns* category of the dataset using UMAP. Color Map: Blue - Negative, Yellow - Positive.

TABLE II
Contributions of Team Members

| Tasks | Team Members |
|---|---|
| Background Study and Data Collection | All Team Members |
| Data Preprocessing | Aniruddha, Fenil, Ishan |
| Data Embedding | Aditya, Ishan, Shivam |
| Clustering | Aditya, Fenil, Pawan, Shivam |
| Sentiment Analysis | Aditya, Ishan, Pawan |
| Visualization | Aditya, Aniruddha, Shivam |
| Testing, and Evaluation | All Team Members |

Aditya, Aniruddha, and Fenil explored work related to text embeddings and clustering algorithms. Ishan, Pawan, and Shivam explored work related to data preprocessing and algorithms for text visualization.

- **Data Preprocessing**: It was handled by Aniruddha, Fenil, Ishan. The preprocessing of the text data of news was done via the operations of noise removal, lowercase conversion, normalization, stop word removal, and lemmatization.
- **Data Embedding**: We used Google's Universal Sentence Encoder to convert the preprocessed textual data into an 512 dimensional embedding vector. This module was implemented by Aditya, Ishan, and Shivam.
- **Visualization**: The module to visualize the textual data using UMAP and t-SNE using the embedding vectors was implemented by Aditya, Aniruddha, and Shivam using seaborn. The 3-D visualization using UMAP and t-SNE was implemented in plotly by Aditya.
- **Clustering**: We explored and implemented a number of clustering algorithms using scikit-learn to identify groups of related news articles based on the embedding vectors. These implementations was split between Aditya (K-Means, LDA), Fenil (Agglomerative Clustering, Gaussian Mixture Model), Shivam (HDBScan), and Pawan (Birch).
- **Sentiment Analysis**: Sentiment Analysis is used to analyze the emotional timbre of the articles. We have used Valence Aware Dictionary and sEntiment Reasoner(VADER) from the vaderSentiment library to perform sentiment analysis. This module was implemented by Aditya, Pawan, and Ishan.
- **Testing, and Evaluation**: The testing and evaluation has been managed jointly by all the team members - Aditya (K-Means, VADER), Fenil (Agglomerative Clustering, Gaussian Mixture Model), Shivam (HDBScan), Pawan (Birch), Aniruddha (Preprocessing, Sentence Embedding), Ishan (LDA).

## IX. Conclusion

In conclusion, we did a subjective analysis during this project through visualization of the clusters which helped us to thoroughly analyze the dataset. We were able to implement algorithms such as LDA, HDBScan, and K-Means to perform clustering and visualised the clustered categories using UMAP and t-SNE. Comparision of all mentioned algorithms was done using the evaluations metrics set by us. We could infer that LDA was giving optimal results on the given dataset. Additionally as suggested by Professor, we used the Sentiment Analysis using VADER which helped us to visualize the various emotional quotient of the data. From the visualization results, we observed that the generated sentence embeddings are not only able to capture the sense of similar categories in the dataset, but also able to differentiate between the positive and negative sentiment in each of the categories.

## References

[1] Richard C. Dubes and Anil K. Jain, Algorithms for Clustering Data, Prentice Hall, 1988.

[2] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey, Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, SIGIR ‘92, Pages 318 – 329, 1992.

[3] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

[4] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002.

[5] Cer, Daniel, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant et al. ”Universal sentence encoder.” arXiv preprint arXiv:1803.11175 (2018).

[6] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. ”Latent dirichlet allocation.” Journal of machine Learning research 3, no. Jan (2003): 993-1022

[7] Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. ”The global k-means clustering algorithm.” Pattern recognition 36.2 (2003): 451-461

[8] Campello, Ricardo & Moulavi, Davoud & Sander, Joerg. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. 7819. 160-172.

[9] Maaten, Laurens van der and Geoffrey E. Hinton. “Visualizing Data using t-SNE.” Journal of Machine Learning Research 9 (2008): 2579-2605.

[10] McInnes, Leland, John Healy, and James Melville. ”Umap: Uniform manifold approximation and projection for dimension reduction.” arXiv preprint arXiv:1802.03426 (2018).

[11] Hutto, Clayton, and Eric Gilbert. ”Vader: A parsimonious rule-based model for sentiment analysis of social media text.” Proceedings of the international AAAI conference on web and social media. Vol. 8. No. 1. 2014.

[12] Scikit-learn.org. 2022. 5.6.2. The 20 newsgroups text dataset — scikit-learn 0.19.2 documentation. [online] Available at: https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html [Accessed 3 May 2022].