Document Clustering, Summarization, and Visualization

CSE 573 Semantic Web Mining (Spring 2023): Group 17

Nagavenkata Jagan Chandanada
1224685696
nchandan@asu.edu

Harish Paul Thavisi
1224516787
hthavisi@asu.edu

Deepthi Reddy Obulareddy Gari
1225492203
dobulared@asu.edu

Uday Venkata Mahesh Matta
1223446549
umatta@asu.edu

Blessy Hadassa Konedena
1223871610
bkonedan@asu.edu

Nandakishore Narasimhamurthy
1219727158
nnarasi3@asu.edu

Abstract—**The project proposal aims to investigate a range of techniques for document clustering, summarization, and visualization. The study will involve implementing two machine learning (ML) models to summarize the cluster topic and cluster summaries, respectively. The first step in document clustering will involve creating vectors from text, which will be done using Top2Vec, a topic modeling technique. Next, Bisecting K means will be used to cluster the vectors. To summarize the clusters, abstractive summarization will be implemented. After clustering and summarization, different visualization techniques such as t-SNE, UMAP, and DASH can be used to represent the results. The research will use the 20 NewsGroups dataset, comprising around 20,000 documents distributed across 20 categories in a nearly equal proportion.**

Index Terms—**Natural Language Processing, Abstractive Summarization, BERT, Transformers, Cluster Summarization, Document Clustering, Top2Vec, Bisecting K-means, LDA, Document Visualization, and UMAP.**

I. Project Definition

When users search for a query, the search engine usually returns numerous pages related to the query. It is often challenging for users to locate the relevant information they need. Clustering is a technique that helps group the returned documents into meaningful categories. Document clustering has several use cases. Summaries offer a brief overview of the entire document, which helps readers determine whether a paper/document is worth reading. Researchers typically read titles first, followed by abstracts, to determine if a paper is relevant to their research. As such, the summarization techniques used in this project must provide a clear summary of the document's entire information in a concise manner. Visualizations are useful for representing large datasets graphically, aiding in better understanding. Our project is useful in two ways.

1) **Clustering of Documents to Summarization of the Cluster Topic:** This technique can be beneficial for websites like Medium, Wikipedia, blogs, or any other application that utilizes documents.
2) **Summarization of Documents to Clustering of Summaries:** As summary embedding vectors are smaller, this technique can enhance the performance of live applications such as search and recommender systems.

The steps followed for the project are as follows:

*a) **Data collection**:* This step involved collecting the necessary data from the data source and understanding its structure of it.

*b) **Background study**:* We will study the various existing models that can be employed on our data that is collected.

*c) **Data cleaning/ preprocessing**:* The process of obtaining the initial text data, removing any irrelevant or unwanted elements, and modifying the remaining data to meet the requirements of the algorithm.

*d) **Feature Engineering**:* This step involves generating multiple sentence embeddings and evaluating them based on their effectiveness.

*e) **Document clustering**:* This involves creating embedded vectors using top2Vec and then applying clustering techniques such as Bisecting K-Means to group the vectors.

*f) **Summarization**:* Summarize the clustered documents to provide a brief understanding of the information contained in each document.

*g) **Visualization**:* Use UMap and t-SNE techniques to visualize high-dimensional data.

*h) **Evaluation**:* Evaluate the clustering and summarization techniques using various metrics such as NMI and Rand Index to gauge their real-world usability and effectiveness.

*i) **Report**:* Generate a report detailing all experiments, analysis of various techniques used and compared for performance, and the different testing procedures conducted to evaluate performance.

II. RELATED WORK

Agglomerative hierarchical clustering and K-means are the most commonly used document clustering techniques. While Agglomerative hierarchical clustering is generally considered superior to K-means, it is slower. A well-known study (referenced in [1]) found that Agglomerative hierarchical clustering outperformed K-means, although this was not on textual data. Within the textual domain, Scatter/Gather [2], a document browsing system based on clustering, utilized a hybrid approach involving both K-means and Agglomerative hierarchical clustering.

The bag of words model is a widely used feature model for text classification tasks due to its simplicity and high performance. The model represents text as a collection of individual words, with no consideration for grammar or word order. The bag of words model is also commonly used in sentiment analysis and has been employed by many researchers. Although this model represents a significant simplification, it has been shown to provide relatively good performance. There are three ways to use prior polarity of words as features, with the simplest unsupervised approach being to use publicly available online lexicons/dictionaries that map a word to its prior polarity.

Some of the earliest work in this field focused solely on classifying text as positive or negative, assuming that all provided data was subjective (e.g., [3] and [4]).

III. DATASET

The 20 NewsGroup dataset is a widely used and publicly available dataset for text classification and clustering tasks. Originally collected by Ken Lang, it contains about 20,000 documents that are partitioned across 20 newsgroups, each corresponding to a different topic. The dataset is known for its diverse topics, including politics, religion, sports, and technology, among others.

One of the notable characteristics of the dataset is the variation in the similarity between the newsgroups. Some of the newsgroups are closely related, such as comp.sys.ibm.pc.hardware and comp.sys.mac.hardware, while others are highly unrelated, such as rec.autos and sci.space. This feature of the dataset makes it a good challenge for testing advanced clustering techniques, especially those that can handle high-dimensional and sparse data.

The dataset has been widely used in research and benchmarking of text classification and clustering algorithms. It has also been used as a benchmark for evaluating topic models and document embedding techniques. Its availability and standardization make it a popular choice for researchers and practitioners who want to compare the performance of their algorithms against existing methods.

Overall, the 20 NewsGroup dataset is a valuable resource for the natural language processing community, providing a diverse and challenging collection of documents for various text analysis tasks.

IV. ALGORITHM PLANNING

A. DATA PREPARATION/PRE-PROCESSING:

We all know that one of the forms of unstructured data is text data, it is necessary to split (phrase, sentence, or paragraph) the data into a set of valid tokens.[4] A method known as stemming is used to reduce the word to its stem by precisely removing the suffix and prefix words present in the text. Even though stemming is useful, we will also use another method: lemmatization. It is an algorithmic process of determining the lemma of a word based on its meaning. Also, we will identify all the stop-words which are nothing but articles, prepositions, pronouns, and conjunctions that form the basic words and do not have much meaning in the context and remove them.[18] Moreover, a predetermined set of punctuation marks will be eliminated. Also, we will eliminate any extra spaces and numbers that might not be necessary. We will use a process called vectorization, which simply involves transforming the raw text input data into a vector of real values that machine models can readily support. This may be done in a number of ways.

B. DOCUMENT CLUSTERING:

There are several methods for performing document clustering. In this project, we've opted for a two-step method in which we first extract the embedding/vector from each text before using clustering algorithms to group the documents into k clusters based on these embeddings. Below is a detailed explanation of each action:

STAGE-1: Extracting Document Embedding Vectors: There are a number of methods used here, ranging from BOW, TF-IDF, which determines the significance of specific words, to BERT, Word2Vec, Doc2Vec, and Top2Vec, which builds an embedding vector that encodes the data from complete texts. We would use this strategy since embedding vector approaches take into consideration the intricate semantic links between the words.

STAGE-2: Clustering Algorithm: Once the embedding vectors for each Document have been extracted, we can choose between flat clustering algorithms like K-Means, which co-optimize all the K-Clusters simultaneously and thus speed up training times, and hierarchical clustering algorithms like HDBScan, which build clusters sequentially and therefore slow down training times but produce better clusters. We combine these two methods to create a hybrid strategy that clusters data sequentially, two at a time, is quite quick, and can have non-spherical cluster distributions to enable more complicated cluster forms.

C. CLUSTER SUMMARIZATION:

Assigning a Summary, Title, or Tag to a Cluster based on the Node that represents the Cluster's Mean is known as Cluster Summarization. The highest performance is offered by abstractive summarizing, which is one of various approaches for summary that are accessible, including maximum vector distance, extractive summarization, and abstractive summarization. So, depending on the themes, we will use a pre-trained Summarization Model to provide a summary for each detected cluster.

V. Research and Literature Survey

A. DOCUMENT CLUSTERING:

The semantic information between words in a document is captured by embedding-based approaches, which generally outperform BOW-based models. We are going to research and compare a few SOTA models to see which is better for our dataset in order to select the best model:

1. BERT- transformer encoder: A Transformer-based ML approach for pretraining NLP is called BERT. Similar to the original transformer Model, it includes self-attention heads and a changeable number of encoder layers. In this scenario, BERT creates a hidden vector representation of an English sentence that can be utilized for clustering, classification, and creative tasks. In order to create the vector representation of each document, we will use this.

2. Word2Vec: A Neural Network Algorithm was used to learn word associations from a sizable text corpus [5]. This can help us comprehend how different or similar two words are by identifying synonymous words or suggesting extra words. In order to obtain the document embedding, we use this to create the vector representations of each word in the document.

3. Doc2Vec: This is a more inclusive variation of the Word2Vec that was mentioned. This model uses the hierarchical SoftMax or negative sampling technique to learn paragraph and document embeddings using distributed memory and distributed bag of words models [6]. Instead of the prior method of naively averaging the embedding vectors for each word in the document, using technique can give a better representation of the complete content.

4. Top2Vec: This method of topic modeling is employed to identify topics (latent semantic structures) within a sizable body of documents [7]. The limitations of the preceding approaches, including the need for a certain number of topics to be familiar with, custom stop-word lists, lemmatization and stemming, LDA, and probabilistic latent semantic analysis, are all eliminated by this model. Moreover, they rely on the BOW representation, which disregards word order and semantics.This method makes use of distributed representations of words and texts, which encapsulate the semantics of both. In comparison to probabilistic generative models, it produces topics that are substantially more informative and representative since it makes use of the joint document and word semantic embedding to find topic vectors.

5. Bisecting K-Means Clustering: We divide the data points into two clusters, and unlike standard KMeans, we only compute the distance between each data point and each cluster centroid once for each step until convergence [8]. Then, we select one cluster and further divide it in the steps that follow. Unlike to K-Means, which assumes that clusters are spherical in shape, this can recognize a cluster of any size and shape. According to Abirami K's discussion of Web Data in the IJETER Journal, this is particularly crucial [9].

6. Neural topic modelling: Using Neural Topic Modeling to create soft cluster assignments for each document is a totally different approach to document clustering. Also, we'll investigate a number of these approaches to see if they perform better than our chosen strategy. Use LDA (or) Neural Topic Modeling directly, which eliminates the need to create intricate Embedding vectors. We vectorize the material relatively quickly, using a technique called count vectorization. Following that, the model groups them into Topics using a mathematical formulation, such as LDA, or by using neural networks, as in Neural Topic Modeling. For each of the identified Topics, it generates a probabilistic representation of a document.

B. DOCUMENT SUMMARIZATION:

While Abstractive Summarization is the more sophisticated of the two high-level ways to dealing with Document Summarization, we'd like to briefly examine whether the other approaches are a better fit given the training infrastructure and the timescales.

1. Maximum Vector Difference: An alternative method of summarization compares the components of one cluster mean, c, to the components of another cluster mean, e, which represents the mean of everything else not utilized to construct c. By doing this, we may generate a distinctive summarization for c. Call this e for now. D = e-c is the result of comparing the elements of the two vectors and computing the difference between the two values. We can find the c that deviates from the rest of the dataset the most by looking at the component of d with the highest value. The collection of items in set C can be summed up using this [14].

2. Extractive Summarization: Identification of whether the article phrases belong in the summary is modeled as a classification task here [10]. It generates an intermediary representation, the primary function of which is to highlight or omit the text's most crucial details so that they can be distilled using the representations [11].

3. Abractive Summarization: This method uses natural language generation to create summaries with the goal of creating an abstract representation of the original content [13]. In other words, it creates new text while preserving the main sense of the old document. The new content may contain expressions, sentences, or words that did not present in the original text. The objective is to produce summaries that are cohesive, readable, and redundant. This is a more difficult modeling assignment than extractive summarization, but it frequently produces better summaries that are closer to those written by humans.

C. VISUALIZATION:

To have a better understanding and demonstrate our results, we will attempt to display both the data and the results. Here are a few visuals that we intend to investigate further:

1. SIMILARITY GRAPH: By algorithm and approach visualization, Gaining understanding of sample similarity clusters requires using a sample similarity graph [16].To find samples with extremely comparable feature vectors/embeddings, we can apply a similarity metric that suits our data.

2. DASH: It is implemented using flask and plotly. A framework is offered by Plotly express and DASH that makes it simple to produce high-quality visualizations [15].

3. T-SNE: Unsupervised, non-linear, and computationally intensive, t-Distributed Stochastic Neighbor Embedding (t-SNE) is mostly used for data exploration and visualizing high-dimensional data. For pairs of instances in both the high-dimensional space and the low-dimensional space, the t-SNE algorithm determines a similarity measure. Then, using a cost function, it attempts to maximize these two-similarity metrics. Segmentation research, learning, and evaluation could all be done using t-SNE. t-SNE frequently reveals distinct separation in the data.

4. UMAP: UMAP is a dimension reduction algorithm that draws inspiration from topological data analysis and various learning strategies. It offers a very broad framework for thinking about dimension reduction and manifold learning, but it can also lead to, tangible realizations. When dealing with huge datasets with intricate and numerous dimensions, this method performs faster than t-SNE.A new cost function and the lack of normalization of high- and low dimensional probability are two enhancements that UMAP has over t-SNE.

5. CLUSTER GRAPHS: By Results and Performance visualization, this graph was created by disjointly joining together complete graphs [17]. If and only if a graph doesn't have a three-vertex induced path, it qualifies as a cluster graph.

VI. EVALUATION PLAN:

A. CLUSTERING:

Typical objective functions in clustering formalize the goal of attaining high similarity of the documents within the same cluster, and low similarity of the documents from different clusters. This is an internal/offline criterion for quality. But external criteria are more robust in terms of representing the model performance. So, we choose 4 external criteria to evaluate clustering quality

1. Normalized Mutual Information (NMI): Like how we calculate Maximum Likelihood estimates of the probability, we calculate the information based on the odds of a document being a cluster. We can also say that the relative frequency is the estimation of each likelihood. We still rely on the Rand Index because the NMI/MI still does not penalize high cardinalities or clusters.

2. F-measure: Although separating similar documents might sometimes be worse than grouping together pairs of different documents, we utilize the F-measure to balance the accuracy-like metric that was previously described.

3. Purity: The class that is most frequently found for each cluster is assigned to each cluster, and the accuracy of this assignment is determined by counting the number of documents that were correctly assigned and dividing that number by N. A perfect clustering has purity of 1, while bad clusters have purity that is close to 0. When there are many clusters, it is simple to attain high purity; NMI helps to balance this trade-off.

4. Rand Index: This is comparable to a classification accuracy metric that takes the confusion matrix into account and gives both false positives and false negatives equal weight. Rand Index calculates the proportion of right decisions.

B. SUMMARIZATION:

In the past, human-annotated summaries have been used to compare and contrast summarizing methods. This makes evaluating this model exceedingly challenging. As an alternative, we compare the performance of a Summarization Model to that of the SOTA Transformer Models. To evaluate the effectiveness of the Summarization Model, we compute a correlation coefficient between the summaries produced by our implementation and the SOTA.

VII. PROJECT TIMELINE:

Task	Deadline
Data collection	February 1st - February 7th
Background study	February 1st - February 15th
Data Cleaning	February 15th - March 1st
Feature Engineering	March 1st - March 15th
Document Clustering	March 15th - March 20th
Data Visualization	March 20th - March 24th
Report	March 1st - March 28th

VIII. WORK DIVISION:

MEMBER	WORK ALLOCATED
HARISH PAUL THAVISI	Background study and Data Collection, Feature Engineering, Clustering the documents, and Visualization of the data.
NAGAVENKATA JAGAN	Background study and Data Collection, Data Cleaning and Preprocessing, Visualization of Preprocessing, Visualization of the data, and report.the data, and report.
DEEPTHI REDDY	Background study and Data Collection, Clustering of the documents, Visualization of the data, and report.
BLESSY KONEDANA	Background study and Data Collection, Data Cleaning and Preprocessing, Feature Engineering, and report.
NANDAKISHORE	Background study and Data Collection, Data Cleaning and Preprocessing, Clustering the documents, and report.
UDAY MAHESH	Background study and Data Collection, Data Cleaning and Preprocessing, Feature Engineering, and report.

References

[1] Home Page for 20 Newsgroups Data Set. http://qwone.com/ jason/20Newsgroups/. Accessed 12 Oct. 2022.

[2] Violante, Andre. An Introduction to T-SNE with Python Example. Medium, 30 Aug. 2018, https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1.

[3] Oskolkov, Nikolay. How Exactly UMAP Works. Medium, 10 Mar. 2021, https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668.

[4] Agrawal, Raghav. Must Known Techniques for Text Preprocessing in NLP. Analytics Vidhya, 14 June 2021, https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp.

[5] Word2vec. Wikipedia, 11 Oct. 2022. Wikipedia, https://en.wikipedia.org/w/index.php?title=Word2vecoldid=1115451429.

[6] Gensim: Topic Modelling for Humans. https://radimrehurek.com/gensim/models/doc2vec.html. Accessed 12 Oct. 2022.

[7] Angelov, Dimo. Top2Vec: Distributed Representations of Topics. arXiv, 19 Aug. 2020. arXiv.org, https://doi.org/10.48550/arXiv.2008.09470.

[8] Mishra, Prakhar. Bisecting K-Means Algorithm Clustering in Machine Learning. Medium, 28 June 2021, https://towardsdatascience.com/bisecting-k-means-algorithm-clustering-in-machine-learning-1bd32be71c1c.

[9] Savaresi, Sergio Matteo and Daniel Boley. On the performance of bisecting K-means and PDDP. SDM (2001).

[10] Licht, Gary. Extractive Summarization Using BERT. Medium, 31 Oct. 2020, https://towardsdatascience.com/extractive-summarization-using-bert-966e912f4142.

[11] Roy, Abhijit. Understanding Automatic Text Summarization-1: Extractive Methods. Medium, 7 Aug. 2020, https://towardsdatascience.com/understanding-automatic-text-summarization-1-extractive-methods-8eb512b21ecc.

[12] Kouris, Panagiotis et al. Abstractive Text Summarization: Enhancing Sequence-to-Sequence Models Using Word Sense Disambiguation and Semantic Content Generalization. Computational Linguistics 47 (2021): 813-859.

[13] Nlp.stanford.edu. 2022. Topical Clustering, Summarization, and Visualization. [online] Available at: https://nlp.stanford.edu/courses/cs224n/2003/fp/millersj/cs224nfp.pdf [Accessed 13 October 2022].

[14] Kouris, Panagiotis et al. Abstractive Text Summarization: Enhancing Sequence-to-Sequence Models Using Word Sense Disambiguation and Semantic Content Generalization. Computational Linguistics 47 (2021): 813-859.

[15] Hwang, J. P. NLP Visualisations for Clear, Immediate Insights into Text Data and Outputs. Plotly, 30 Mar. 2020, https://medium.com/plotly/nlp-visualisations-for-clear-immediate-insights-into-text-data-and-outputs-9ebfab168d5b.

[16] Nunes, Diogo A. P. Visualising Similarity Clusters with Interactive Graphs. Medium, 15 Dec. 2021, https://towardsdatascience.com/visualising-similarity-clusters-with-interactive-graphs-20a4b2a18534.

[17] Cluster Graph. Wikipedia, 26 June 2022. Wikipedia, https://en.wikipedia.org/w/index.php?title=Cluster_grapholdid=1095082294.

[18] Sanketh, Ruthu S. Text Preprocessing with NLTK. Medium, 27 May 2021, https://towardsdatascience.com/text-preprocessing-with-nltk-9de5de891658.