Combining Hierarchical Clustering with Sentance-BERT for fast Information Retrieval

PavanYadav Shivaji Rao, Sanjana Sudarshan Shroff, Vaishnavi Gonela

Abstract:

The PubMed database has an overall collection of 34 million citations and has an update of around 300 citations everyday as given by the PubMed website. Thus, it is immensely important that users can fetch the most relevant documents from a big data corpus. Another important factor is to reduce the time complexity when retrieving similar documents from a large corpus pertinent to the user's area of interest. This paper proposes a method of initially applying an unsupervised hierarchical clustering method like Agglomerative clustering on the Medical Subheadings to derive the cluster that the query belongs to. Once the cluster is fetched, Sentence BERT using Siamese network architecture is used to derive the semantic similarity between the user's query and the remaining sentences in the corresponding cluster to fetch the top-K sentences with maximum cosine similarity. This method reduces the retrieval time by a factor of cluster size divided by corpus size for performing semantic similarity and reduces clustering time as it uses an average vector of title, abstract and MESH terms. Combining the two models gives the semantic meaning and is comparatively faster as the sample size is reduced after clustering. This approach does not compromise on the efficiency of the algorithm for information retrieval and produces accurate results within a stipulated time frame. SBERT is used as it provides good performance for sentence embeddings and it is anticipated to capture unseen semantic meaning of sentences which is proven to be better than other embedding methods.

Keywords: Document Clustering, Agglomerative clustering, Term Frequency-Inverse Document Frequency, Sentence-BERT

1.Introduction:

Like all other search engines, PubMed's objective is to provide citations that are thought to be pertinent to a user query. In an effort to prioritize the most pertinent results at the top of the ranking list, modern search engine developers have put a lot of effort on optimizing retrieval result rankings. Due to the inherent complexity of ranking search results, no ranking system is ideal. This complexity has several different conceivable query types, which contribute to one component of it. Over 17 million citations to biomedical documents are indexed by MEDLINE, which has emerged as the primary database for biomedical text mining. Web-based programmes were created to offer search options across MEDLINE as well as other related collections, such as PubMed. It is a continuously expanding database, therefore many users, including biomedical researchers, must cope with difficult and time-consuming tasks like finding, searching for, reading, and assessing biological publications to exploit these resources. As a result, many techniques are employed in the retrieval of the biomedical papers to give the user the best possible result. However, users are still getting a lot of irrelevant results [11].Text mining, sometimes referred to as text data mining, is the process of obtaining relevant knowledge or information about a certain topic from a collection of papers. The technique of extracting useful information from textual content is known as text mining [5]. It is helpful when the data is in an unstructured format that can't be analyzed using traditional means. It is seen as a means to enable computers to use unstructured data and is also known as "text analytics"[6].Some of the operations used in the Text Mining are [6]:

- Extraction of Features. It is used to examine the relationships between text data. This helps in classifying essential concepts and outlining their connections is helpful.
- Term document frequency: Preparing the term document matrix (TDM), in which words are represented by rows and documents are represented by columns, comes after pre-processing is finished. Important metrics in the text mining process include TF-IDF.
- Categorization (Unsupervised classification): Documents are grouped on the basis of some similarity measure.

In document clustering, each cluster is given a topic, and topic representation models are built on the documents in each cluster. We therefore want to understand why that particular topic was assigned to that cluster rather than the other clusters. In order to determine this, we'll use the TF-IDF measure, which defines a word's significance to a document and helps in understanding its relation to a specific topic. The TF-IDF, which combines the statistics of term frequency and inverse document frequency, is an acronym for this phrase [18] in which the term frequency corresponds with the term frequency of the paper. A measure of the amount of information a word adds to a document is the inverse document frequency. This is determined by dividing a corpus's overall document number by the algorithm used to estimate how many papers are in the corpus. The TF-IDF Transformer, which generates the term frequency with inverse document frequency, is utilized in this work. Therefore, in order to ascertain the term frequency of a specific text, tf-idf transformers require a count vectorizer.

Text is transformed by Countvectorizer into a word count matrix known as a Term Document Matrix, where terms are shown as columns and documents as rows. This is often referred to as "Bag of words" (BOW), which simply means counting the words and placing them in a bag, regardless of their structure or sequence. The text will be transformed into a sparse matrix by this count vectorizer. Then, using the input as a sparse matrix, the tf-idf transformer will determine the inverse document frequency for each word in the term document frequency. In this paper, the text data was normalized using the tf-idf transformer [19].

Unsupervised Document Clustering:Grouping the Documents into distinct Groups is the aim of unsupervised document clustering. Hierarchical clustering is one technique that can be used to group documents.

Hierarchical Clustering: Using a top-down (divisive) or bottom-up strategy (Agglomerative), it produces a hierarchy cluster of documents with a predetermined order from top to bottom. In this paper we are using Hierarchical Agglomerative clustering to form the clusters.

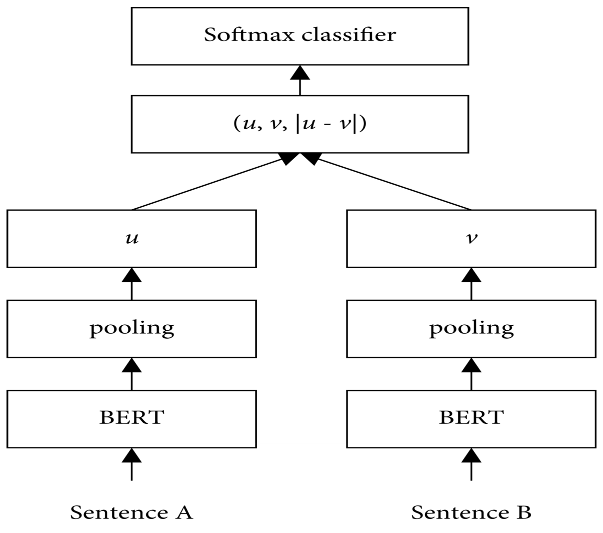
Agglomerative clustering: An unsupervised technique called agglomerative hierarchical clustering begins by placing each document in its own cluster.As contrast to partitional algorithms, which build the solution from top to bottom by first assigning each document to its own cluster, agglomerative algorithms build the hierarchical solution by continually selecting and merging pairs of clusters to establish a single all-inclusive cluster. Agglomerative algorithms build from the bottom up as a result. The most important factor is the process used to determine whether two clusters in agglomerative algorithms should be combined at each phase. The majority of agglomerative algorithms accomplish this by selecting the most similar pair of clusters, and numerous techniques have been developed to ascertain how similar two clusters are [20].Hierarchical Agglomerative clustering begins by treating each observation as a separate cluster and then iteratively combines clusters until all of the data points are combined into a single cluster. Clusters are merged depending on their distance from one another, and there are various types of connections that can be used to compute that distance, including single, full, ward, and average links. Therefore, we calculated the nmi score in this study using three different linkage measures, including average, complete, and ward, for the ground-truth dataset.

At each stage, the algorithm then connects the most related documents until there is only one cluster left. The objective is to categorize a document that has never before had a topic assigned to it. To determine how different the documents are from one another, this method employs a wide range of similarity measurements. Each document is represented as a term vector when employing TF-IDF, and the similarity between the documents may be determined by calculating the cosine angle between the vectors [9].

The amount of biomedical literature is expanding quickly in the modern period due to the biomedical field's rapid expansion. Many works of literature are retrieved when a user enters a search query for a single area of interest. The complexity of the time frame is another significant influencing factor. In this paper, we use agglomerative clustering based on mesh terms of various documents to construct clusters that only retrieve relevant papers in order to solve this challenge.

Bidirectional Encoder Representations from Transformers (BERT) is said to have better performance while calculating semantic textual cosine similarity scores between different sentences [17]. In BERT model, the desired value is estimated using such a cross-encoder whenever the transformer network is given two texts. Nevertheless, due to the significant processing expense, this arrangement is inappropriate for a number of pair regression applications. Each sentence is mapped to a vector space in order to make semantically comparable sentences close together, which is a typical approach to the problems of clustering and semantic search. When one sentence is entered into BERT, it generates fixed-size sentence embeddings. This method results in extremely poor sentence embeddings. Sentence-BERT (SBERT) creates a fixed-size sentence embedding vector by adding a sharing procedure using BERT's result.

The BERT model's performance is improved by SBERT's updating of the weight parameters with the Siamese and triplet networks to ensure that the resulting sentence vector has semantic information. Sentences with similar meanings have a closer distance between their embedding vectors, making it possible to calculate similarity between them. Different similarity measures like Cosine similarity, Manhattan Distance and Euclidean Distance are used to measure the level of relevance between the vectors. Figure below depicts the SBERT structure when classification is the objective function.



This method is used to provide the context in which the word appears in the sentence. The sentence-transformers python library is used to reproduce the SBERT model [14]. Once the model is embedded using the corpus, cosine similarity is calculated to find the resemblance between the query and the documents. In this method, sentence embeddings are found for sentences within abstracts using model encoding. Then it is used to calculate the semantic similarity with the query embedding to rank them accordingly.

2. Materials

Text clustering has become more significant in recent years as a result of the rapid growth of online documents. One of the most advanced unsupervised classification approaches is clustering, which involves grouping various documents into clusters with unknown class labels [6].The idea of document clustering is a crucial application in the field of text clustering that makes it easier for individuals to find the papers they're interested in. The performance of the document clustering directly depends on the ability to detect text similarity. By identifying the node correlation in the structure, semantic similarity between concepts in an ontology may be measured. Concepts are represented as nodes in a hierarchical framework called an ontology. The Medical Subject Heading (MeSH) semantic data has been used to cluster MEDLINE papers in the field of biomedical text mining. It is one of the most widely applied ontologies, and its primary components are a structured language, a MeSH Tree, and MeSH descriptors [11].

3. Methods

This section discusses the improvised methods that are adapted in this paper to fetch the most similar documents. The most common procedure of classifying and identifying documents is to cluster together comparable documents. Hierarchical clustering has been chosen to address the issue of handling unknown data because it offers data-views at various levels of abstraction, making it perfect for users to interactively examine huge document collections and visualize the concepts generated. On the other hand, when the documents to be compared are clustered, semantic similarity is calculated between documents in a pairwise manner to obtain most similar documents. We start by introducing hierarchical clustering and then discuss the embedding models and show how the two concepts when linked together yield better and faster results.

3.1. Gathering Data

This paper proposes working with different datasets prepared from TREC 2005 Genomics and PubMed database in order to build and validate the model in the unsupervised setting.

- Prepared dataset from PubMed:

  The purpose of this dataset is to help overcome certain difficulties while dealing with unsupervised clustering methods. For preliminary analysis, the PubMed dataset[2] prepared consists of 10 non-overlapping general topics related to Bioinformatics. 10 articles based on the search results are selected from each of the topics to create clusters in the further steps. To fetch the articles, an account on MEDLINE is created and a particular topic (eg: text mining in Bioinformatics) is searched. There are multiple ways to save the search results in Pubmed such as Csv, pmids, PubMed format. The Pumed format contained all the necessary information like title, abstract, Mesh terms etc. Which could be saved as a text file. With the help of bio and Entrez utilities [7] available in python, the downloaded text files were parsed using Medline parser to extract abstract, title and mesh terms in order to save it as Csv for further analysis. The following diagram shows the steps followed while creating the dataset by this method.

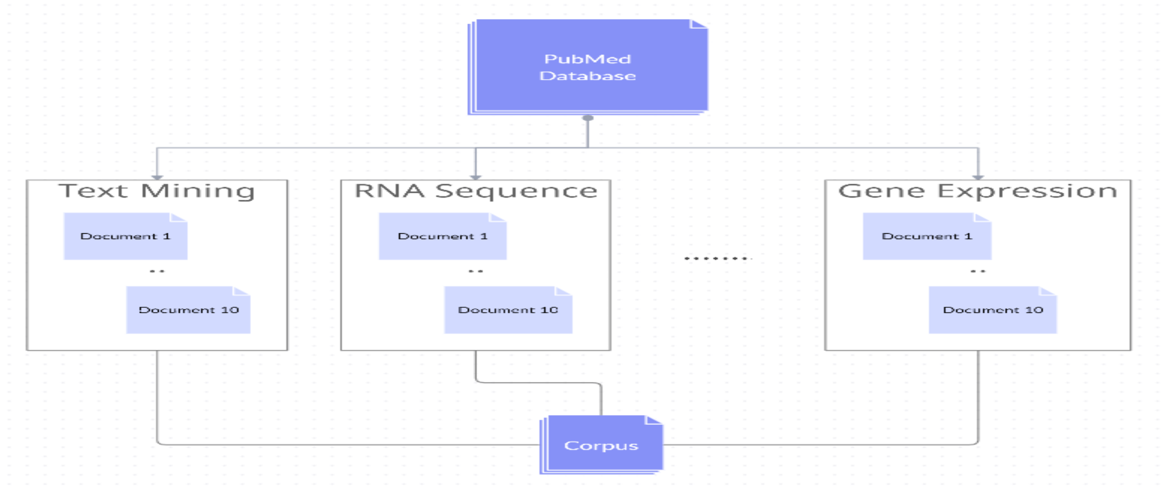


Figure 1: Steps for creating Corpus from PubMed Database for various topics and merging all these individual documents to produce the final Corpus.

This method assists in evaluating the performance of the clustering model. It also assists in obtaining balanced samples for training the model.

- TREC Genomics 2005 Dataset

  The TREC 2005 Genomics Dataset is another corpus taken as a validation set.It provides an excellent platform that makes it easy to compare the effectiveness of various retrieval methods for locating crucial biomedical materials. Medline document clustering research frequently uses this dataset. TThe database's TREC 2005 Genomic Track contained 4,591,000 documents.10 years of complete citations, from 1994 to 2003.. Genomics has created 5 generic topic templates to help biologists in their research. There are 10 topics in each topic template, the topics and documents that fall under each topic and finally contains 50 topics. These 50 topics, which we recognised as true clusters in our studies, were given as queries to all competitive data retrieval systems and imitate actual data needs in the biomedical area. A relevancy score has been assigned to each topic. If the relevance score is 0 then the document is marked as "not relevant", 1 means the document might or might not be relevant, and 2 means the documents are definitely relevant. We are considering only definitely relevant documents for our research. Out of 4,591,000 documents we have selected, only 4,583 documents having relevance score 2 were included in the study. [8].

  To create the corpus of data from TREC, the PubmedIDs along with the topic ID and the relevance score were saved as CSV. Then the Title, Abstract and MESH terms are appended to the CSV by iterating through each Pubmed ID then using Medline parser to fetch the relevant data by parsing the JSON elements. MESH follows a tree hierarchy in which it has major and subtopics. To distinguish the major topic starts with an asterisk symbol and subtopic is appended to the left of major topic (ex: chemoinformatics/*trends). The study consists of both major and subtopic while clustering because the corpus of text is small and does not hinder the efficiency of the algorithm.

3.2. Performance evaluation

The normalized mutual information function is one of the evaluation techniques used to assess the clusters' quality. The NMI approach, which stands for normalized mutual information, can be used to assess the clustering's purity. This method is followed because the ground truth is assumed to be the topics that the document is fetched from. Since the class labels for TREC data are available, the same metric is used to assess the clusters' purity. We are also employing time-complexity analysis to evaluate how long each method takes to process the cluster in this study because quick information retrieval is another crucial factor. For determining the cluster's quality, we also employed the Silhouette score.The efficiency of embedding sentences is assessed by determining the Cosine Similarity scores. There are two embeddings that the model takes, in this case, one is from the query and the other is from the cluster that the sentence is mapped to. These two tensor objects are mapped to find cosine similarity, showing how similar the corresponding documents are.

3.3. Proposed Solution

The dataset is prepared by combining the CSV produced by parsing each PUBMED format text file to form a single CSV containing information (PMID, Abstract, Title, Mesh Terms) related to all articles.

Data pre-processing: While we were extracting the title, abstract, and mesh for each PubMed id from the Medline database, a couple of the abstracts were missing for a few topics. As a result, in our research, we dropped those documents for the TREC dataset, and instead of getting 4,583 documents as previously, we obtained 4,333 records.

Text Pre-processing: Before submitting text data to any clustering algorithms, it must be cleaned and encoded to numeric values. This cleaning and encoding procedure is known as Text preprocessing. In this, we performed the activities like Removing Punctuation, HTML tags from the texts of the review. We have removed the stop words like (and, is, are, etc) from the text as it does not help the model in understanding the sentiment of the user and will also help the model to learn the more weighted words within the reviews. We have applied a technique called stemming which stems the words to their root so that all the words which are of common but the change in their tense will be treated as one word and will help in reducing the vector dimension and converting the text into the lower case for encoding purposes. For the TREC dataset, the Mesh Terms are converted from the list to string format by concatenating each element in the list and removing Html tags or JSON elements present in the text using regular expressions, if any. The feature extraction approach of the transformer model of term frequency-inverse document frequency (TF-IDF) has been applied to the cleaned text since it is the standard weighted information retrieval scheme and is used widely in document clustering[12] to convert text to numeric vectors for calculation. To calculate TF-IDF transformers it is required to use a Countvectorizer to perform term frequency. So, first we calculated the CountVectorizer for the cleaned text and then passed this to the TF-IDF transformers.

Model Training: We Utilized the Kmeans and agglomerative clustering clustering methods for this project. Clustering techniques are fitted with the vectors that result from TF-IDF Transformers. For the dataset we've prepared, Agglomerative clustering uses Euclidean distance as the metric to assess the relationship between cluster pairs, and there are a set of 10 clusters in total because of the existence of 10 topics [13]. The identical set of clusters were utilized in k-means as well. Since there are 50 topics in the TREC 2005 genomics dataset, which is treated as a true cluster, there are 50 clusters in the ground truth dataset. In order to measure the distance between two clusters in agglomerative clustering and merge them, we used hyperparameter tuning for the linkage calculation and passed three different metrics through it, including average, complete, and ward. We also used Euclidean distance as the metric to evaluate the relationship between cluster pairs and compared it with NMI score, time taken, and silhouette score. There are the same number of clusters for the k-means clustering technique.

Semantic Similarity: After forming the clusters, the query has to be mapped to the clusters based on the semantic similarity which is computed using the pre-trained sentence transformers model "all-MiniLM-L6-v2" [15]. The clustered labels are stored as a column having cluster numbers to specify which cluster the article belongs to. For a query, the rows are filtered based on the label, and the abstracts are taken as an input list for corpus embedding then the cosine similarity metric under pytorch utility is used to rank the top k articles based on the similarity scores.

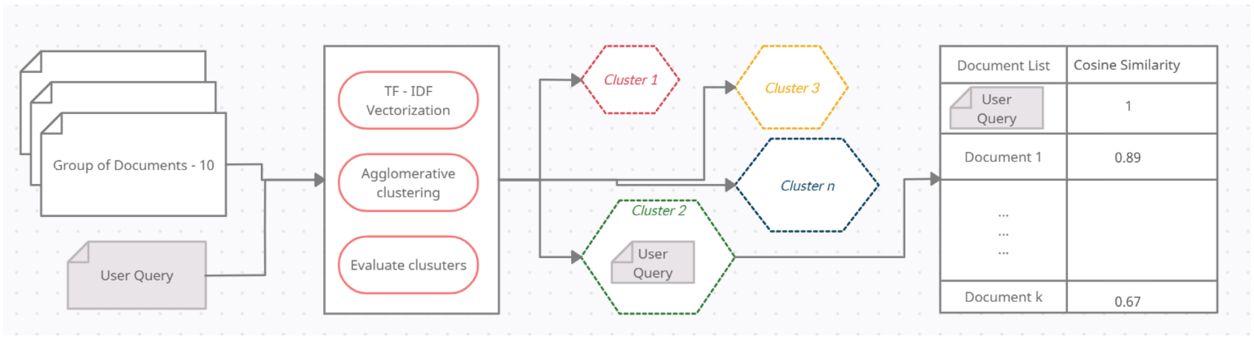


Figure 3: Framework of the of the suggested approach

The suggested solution is shown in the figure above, the group of documents along with the user's query is fed as an input for the model. The input is passed through the TF-IDF vectorization model to build a matrix of dense vectors. The vectors are fit transformed into an Agglomerative clustering algorithm to form the desired number of clusters, as per the requirement we need 10 clusters since we have 10 topics so n_clusters parameter is set to 10 for prepared dataset. The model output is evaluated using the NMI (Normalized Mutual Information) metric available in the sklearn metrics package. Once the clusters are formed, it is seen from the figure that the cluster size might carry based on the number of documents which fall into those clusters. The user query will fall into a cluster which is known to be relevant to the documents within that particular cluster. To calculate the similarity of user query and the set of documents within a cluster, cosine similarity score is used, which will give a score on how similar the documents are based on the embeddings computed using Sentence BERT model. The documents are sorted based on the similarity score with respect to user query and the top-k documents are retrieved from a cluster as per the request made by the user through tkinter UI.

4. Results and Discussion

In the two datasets explained above, we compare predicted clusters with actual classes. We make sure that the clustering method does not use the topic classes while clustering. There are numerous prominent external metrics, including mutual information, average entropy, and F measure. The NMI is used to evaluate how well the two data distributions match up. The NMI index was discovered to be able to achieve a good evaluation effect of clustering in the study given by Ghosh [16]. As a result, this study incorporates additional assessment of clustering performance by applying NMI scores.

Results	Time Taken	NMI	Silhouette Score
K Means with TF-IDF (200 features)	0.079	0.835	0.125
Agglomerative with TF-IDF(200 features)	0.016	0.878	0.133
K Means with TF-IDF (max features)	0.116	0.748	0.042
Agglomerative with TF-IDF(max features)	**0.035**	**0.879**	0.054
TREC Kmeans(max features)	32.15	0.787	0.06
TREC Agglomerative(max features)	**15.66**	**0.851**	0.075

Table 1: Comparing results from different models

Before testing the proposed method on the ground truth dataset, we checked the efficiency of the clustering model by running it on the prepared dataset. First, the clustering was performed using the K Means algorithm by varying the maximum features. Agglomerative is known to be a widely used hierarchical clustering method for text clustering. The results observed for Agglomerative are comparatively better than Kmeans in terms of time taken and the NMI scores as shown is the table below. Once the predicted output produced by the Agglomerative clustering algorithm was comparatively better than Kmeans on the prepared dataset, then the same algorithms were validated on ground truth TREC dataset. It is seen from the table below that the time taken by agglomerative clustering is almost half of that of k-means and the NMI score is 0.851 for Agglomerative whereas Kmeans has 0.787 as NMI score. Therefore, the agglomerative has better performances with regard to timing and accuracy of predicted values.

Title	Similarity score
Chemoinformatics-Driven Design of New Physical Solvents for Selective CO2 Adsorption.	1.0000
Chemoinformatics at IFP Energies Nouvelles: Applications in the Fields of Energy, Transport, and Environment.	0.7987
Machine learning in chemoinformatics and drug discovery.	0.7361
Chemoinformatics-based enumeration of chemical libraries: a tutorial.	0.7335

Intelligently Applying Artificial Intelligence in Chemoinformatics.	0.7264
Chemoinformatics for the Safety of Energetic and Reactive Materials	0.7220

Table 2: Representation of the semantic similarity score

Visualization: Since text data is nominal in nature, it is inherently challenging to visualize numerically. For at least ten years, word clouds have been a popular method of visually presenting textual data to both laypeople and scientists. The most common words in a text are found by using a text mining technique called a word cloud[21]. The word cloud's primary functions are keyword extraction, word clustering, and keyword filtering. In this research, word clouds were used to visualize text information in relation to the clusters that were produced. We created a word cloud for our prepared dataset using clusters of 10 different topics. The word cloud representation of one of our topics, "Food drug interaction," is shown in the figure below. It extracts the most popular keyword associated with that particular topic.

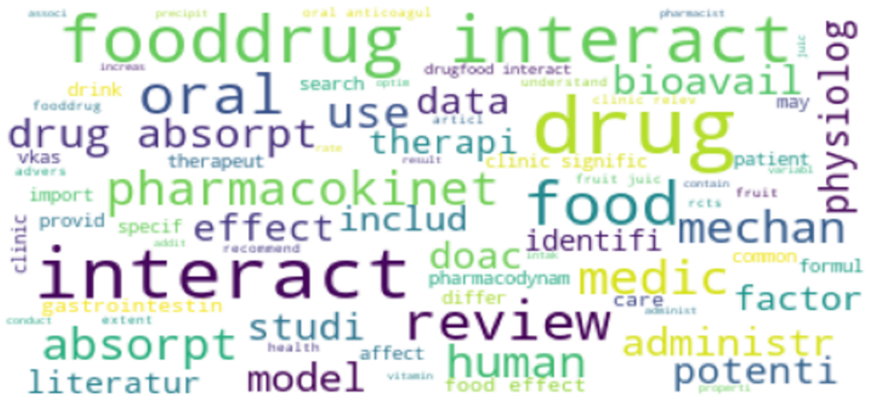


Figure 4: Word cloud

5.Conclusion

Overall, in this study, different methods were reviewed for document clustering specifically targeted for the MEDLINE and TREC 2005 datasets. The main analysis was to measure the time taken to process the clusters, calculate Normalized Mutual Information with the anticipated labels, and the Silhouette scores within different clusters. The results when applied to different datasets with different parameters have shown that Agglomerative hierarchical clustering performs better and consumes lesser time to process the clusters. Another section of research in NLP has shown the importance of capturing the context of text rather than matching keywords. The most optimal results are shown by supplementing the clusters by adding semantic meanings at the sentence level rather than the word level. Cosine similarity scores are calculated after applying Sentence BERT, between documents within a particular cluster to address this issue. Since the total pairwise comparison of SBERT is only performed within the cluster, it reduces the time taken drastically. This method has maximum weightage on the efficiency of clustering and this could be improved in the future by adding optimal solutions to enhance hierarchical clustering. Since the Medical subheadings are represented by trees, having a weighted measure for the MESH headings can also result in better clusters which eventually lead to better similarity scores.

6. Reproducibility

Code and data is available at the following link, https://github.com/sanjshroff/Biomedical
Steps to reproduce are provided as a document in the repository.

References:

[1] https://pubmed.ncbi.nlm.nih.gov/about/

[2] https://pubmed.ncbi.nlm.nih.gov/

[3] https://devopedia.org/text-clustering

[4] Fiszman, M., Demner-Fushman, D., Kilicoglu, H., Rindflesch, T.: Automatic summarization of MEDLINE citations for evidence-based medical treatment: a topic-oriented evaluation. J. Biomed. Inf. 42(5), 801–813 (1999)

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5572517/

[6] Karaa, W.B.A., Ashour, A.S., Sassi, D.B., Roy, P., Kausar, N., Dey, N. (2016). MEDLINE Text Mining: An Enhancement Genetic Algorithm Based Approach for Document Clustering. In: Hassanien, AE., Grosan, C., Fahmy Tolba, M. (eds) Applications of Intelligent Optimization in Biology and Medicine. Intelligent Systems Reference Library, vol 96. Springer, Cham. https://doi.org/10.1007/978-3-319-21212-8_12

[7] Bio and Entrez python module: https://biopython.org/docs/1.76/api/Bio.Entrez.html

[8] Pan, X., Huang, P., Li, S. et al. MCRWR: a new method to measure the similarity of documents based on semantic networks. BMC Bioinformatics 23, 56 (2022). https://doi.org/10.1186/s12859-022-04578-1

[9]https://datamathstat.wordpress.com/2020/09/15/documents-clustering-text-mining/#:~:text=Agglomerative%20 hierarchical%20 clustering%20is%20an,there%20is%

[10] Chaudhary, M., Pruthi, J., Jain, V.K. et al. A novel squirrel search clustering algorithm for text document clustering. Int. j. inf. tecnol. 14, 3277–3286 (2022)https://link.springer.com/article/10.1007/s41870-022-01078-6

[11] Keyvanpour, MohammadReza and Serpush, Fatemeh. "ESLMT: a new clustering method for biomedical document retrieval" *Biomedical Engineering / Biomedizinische Technik*, vol. 64, no. 6, 2019, pp. 729-741. https://doi.org/10.1515/bmt-2018-0068

[12] Term Frequency Inverse Document Frequency (TF-IDF) Transformer Documentation https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html

[13] Agglomerative Clustreing Documentation from scikit-learn https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

[14] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)

[15] SentenceTransformers Documentation https://www.sbert.net/

[16] J. Ghosh, “Scalable clustering methods for data mining,” Handbook of data mining, Lawrence Erlbaum, 2003.

[17] Zheng Hu, Hua Dai, Geng Yang, Xun Yi, Wenjie Sheng, "Semantic-Based Multi-Keyword Ranked Search Schemes over Encrypted Cloud Data", *Security and Communication Networks*, vol. 2022, Article ID 4478618, 15 pages, 2022. https://doi.org/10.1155/2022/4478618

[18] Grootendorst, M., “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”, <i>arXiv e-prints</i>, 2022.

[19]https://towardsdatascience.com/tf-idf-explained-and-python-sklearn-implementation-b020c5e83275

[20] Zhao, Ying & Karypis, George & Fayyad, Usama. (2005). Hierarchical Clustering Algorithms for Document Datasets. Data Min. Knowl. Discov.. 10. 141-168. 10.1007/s10618-005-0361-3.

[21] M. A. Hearst, E. Pedersen, L. Patil, E. Lee, P. Laskowski and S. Franconeri, "An Evaluation of Semantically Grouped Word Cloud Designs," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 9, pp. 2748-2761, 1 Sept. 2020, doi: 10.1109/TVCG.2019.2904683.