1. Data Preparation:

The first step in our analysis involved preprocessing the raw text data from the provided TXT files. Each TXT file contained information about the document-word frequency, which we parsed to create a suitable representation for clustering.

Specifically, we read each TXT file and converted it into a list of sets. Here's how the conversion process worked:

- Each index in the list represents a document ID, starting from 1.
- Within each list index, we stored a set containing all the words present in the corresponding document.

2. Methodology

To cluster the documents effectively, we employed the K-means algorithm with the Jaccard index as the similarity measure. The methodology can be summarized as follows:

Centroid Initialization: We began by randomly selecting K centroids from the document collection. These centroids served as the initial representatives for each cluster.

Document Assignment: Each document in the collection was then assigned to the nearest centroid based on the maximization of the Jaccard index. Specifically, we calculated the Jaccard index between each document set and each centroid set, assigning the document to the centroid that yielded the highest similarity.

Centroid Update: After assigning documents to clusters, we updated the centroids iteratively. The new centroids were determined by selecting words that appeared more frequently within the cluster than the average frequency across all documents. This process ensured that the centroids were representative of the documents in their respective clusters.

Termination Condition: We defined a termination condition to control the number of iterations. The algorithm continued iterating until either 50 iterations were completed or the average Jaccard index between the old

and new centroids exceeded 0.8. This termination condition ensured convergence or stability of the clusters.

3. Results

Nips Dataset

Dataset Characteristics: The NIPS dataset comprised 1,500 documents, each represented as a bag of words. The vocabulary size was 12,419 words, and the total number of non-zero frequency entries was approximately 1,900,000.

Clustering Performance: The K-means clustering algorithm was executed on the NIPS dataset, with different values of K tested to find an optimal value. The algorithm took approximately 18 minutes to run on the entire dataset.

Optimal Value of K: After experimenting with various values of K, we found that the optimal number of clusters for the NIPS dataset was determined to be 7. This value of K resulted in clusters that effectively captured the underlying structure and patterns within the documents.

KOS Dataset

Dataset Characteristics: The NIPS dataset consisted of 3,430 documents, each represented as a bag of words. The vocabulary size was 6,906 words, and the total number of non-zero frequency entries was approximately 467,714.

Clustering Performance: The K-means clustering algorithm was executed on the NIPS dataset, with different values of K explored to determine an optimal value. The algorithm required approximately 25 minutes to complete the analysis on the entire dataset.

Optimal Value of K: Through experimentation with various values of K, it was determined that the optimal number of clusters for the NIPS dataset was 8. This value of K yielded clusters that effectively captured the underlying structure and similarities among the documents.

Enron Dataset

Dataset Characteristics: The Enron Emails dataset consisted of 39,861 documents, each represented as a bag of words. The vocabulary size was 28,102 words, and the total number of non-zero frequency entries was approximately 6,400,000.

Clustering Performance: The K-means clustering algorithm was executed on the Enron Emails dataset, with different values of K explored to determine an optimal value. The analysis took approximately 45 minutes to complete on the entire dataset.

Optimal Value of K: After experimentation with various values of K, it was determined that the optimal number of clusters for the Enron Emails dataset was 7. This value of K resulted in clusters that effectively captured the inherent structure and similarities among the documents.