News Culstering

Kibtia Chowdhury

1 Data Preprocessing

In the initial phase of the code, the necessary packages and modules are imported, and the NLTK data required for tokenization, stopwords, and WordNet lemmatizer are downloaded. The preprocess_text function is defined to handle various text preprocessing tasks such as converting the text to lowercase, removing punctuation and numerals, tokenizing the text into individual words, eliminating stopwords, and lemmatizing the words using WordNet lemmatizer. Moving on to the dataset processing part, the JSON file "News_Category_Dataset_v3.json" is loaded into a pandas DataFrame using the pd.read_json function. The loaded dataset is stored in the df DataFrame. A print statement is used to display the contents of the dataset, providing an overview of the data.
Next, the processed_text column is created in the DataFrame df by applying the preprocess_text function to the concatenation of the "headline" and "short_description" columns. This column contains the preprocessed text data for each row. To verify the preprocessing results, the processed_text column is printed, allowing us to observe the transformed text.

2 Word2Vec Model Training

The Word2Vec model is imported using this from the gensim package. The text in the processed_text column is converted into a collection of lists of words (sentences). A Word2Vec model is constructed and trained using the sentences. The trained Word2Vec model is kept in a file named "news_category_w2v.model".

In this step, the Word2Vec model is loaded from the saved file. This enables the model to be used once more in subsequent phases.

3 Text & Document Vectorization

Using the imported Word2Vec model, the vectorize_text function is developed to preprocess the text, divide it into words, and then turn each word into a word

vector. The function is run across the processed_text column, and the output vectors are saved in the dataset's vector column.

Using the loaded Word2Vec model, the document_vector function is developed to calculate the mean vector of a document (a list of words). Applying the document_vector function to every document in the processed_text column results in the creation of an array X. The vectorized representation of each document in the dataset is represented by this array.

4 Clustering

Imported is the KMeans algorithm from the sklearn.cluster module. Five clusters are included in the KMeans model, which is then fitted to the vectors in the dataset's vector column. The cluster column of the dataset is given the cluster labels that are produced.

5 Evaluation Metrics & Best clusters

The clustering analysis assessment metrics for Sklearn. It divides the dataset into five clusters using the KMeans method, then saves the labels that are produced. The calculated Silhouette Score of 0.0802 suggests that the clusters may be poorly separated or overlapped. With a Davies-Bouldin Index of 2.6786, considerable separation and compactness are indicated. 16258.7428 is the Calinski-Harabasz Index, which suggests a rather strong cluster separation.

This section involves finding the best value of k (number of clusters). A loop is used to iterate over values of k from 2 to 9. For each value of k, the silhouette score is computed using the cluster labels. The best k value and corresponding silhouette score are tracked. The highest silhouette score achieved is approximately 0.10287840036372516.