NLP CLUSTERING

Problem Statement:

With the enclosed data, cluster the utterances using unsupervised learning techniques and explain the reasons for choice of algorithms you use. Assume there is noise in data. Identify important clusters and assign a suitable name to non-noise clusters.

<

h3>Solution:

Unsupervised models used for generating word vectors: Word2vec and TF-IDF

The data is taken from a conversation with a Bank's chatbot and the clusters identified are:

- Loan requirement questions
- Asking the bot to clear bills
- Outstanding balance-check questions
- Money Transfer queries
- FAQs

The Inference for all approaches are mentioned at the bottom of the repective ipynb files and below

TF-IDF with K-means pipeline :

Data -> Delete duplicates -> vectorization with stop words, stemming and tokenization -> Creation of Elbow plot -> identifying clusters -> PLotting clusters using different dimentionality reduction and visualization techniques -> Grouping words by their respective clusters -> Inference

alt text

alt text

Inference:

The elbow curve tells us that the optimal number of clusters are 5,6 or 7, and When we go through the word groupings we understand that the data is talking about the following queries

- Loan requirement questions
- Asking the bot to clear bills
- Outstanding balance-check questions
- Money Transfer queries
- FAQs

According to the TF-IDF word embedding technique which we just used, these five groups of data are classified in 6 clusters, since there is a significant amount of noise in the dataset and because TF-IDF is essentially a scoring algorithm which creates a score for every word, based on their occurences. The distribution of these scores, given their weights, creates a slight bias for the noisy data as well. This is vizualized in the various plots in the ipynb file. Although, the ideal number of cluster remain at 5

Word2Vec with K-means pipeline :

Data -> Creating a corpus -> tokenizing the sentences -> convert to list -> Creating the Word2Vec model -> Training the model -> Saving the trained data -> plotting the k-means elbow curve -> identifying cluster numbers -> Hyperparameter tuning -> Repeat -> Plotting clusters using different dimentionality reduction and visualization techniques -> proving word similarities using the cosine similarity function -> Inference

alt text

alt text

The elbow curve tells us that the optimal number of clusters are 4,5. Plotting these cluster values shows us a clear distiction in the intents. Visualization from the Word2Vec model shows us how the model is more capable of differtiating between noisy data points, due to its nature of going deeper into the dataset and creating contexts based on their semantic meanings.

The results of the word2vec technique combined with the TF-IDF word embedding technique, provides a clear conclusion to the number of intents and clusters we have in our data. It solidifies our understnading of the clusters(5) and their names mentioned above.

The unsupervised learning technique that performs best for this dataset is: Word2Vec, using K-means clustering with PCA and T-SNE for visualization