**NLP Project Milestone 1 Report**

Ziad Amr Shamseldein (52-0295)

Hazem Sherif (52-5272)

Youssef Bayoumi (52-11219)

## **1. Introduction**

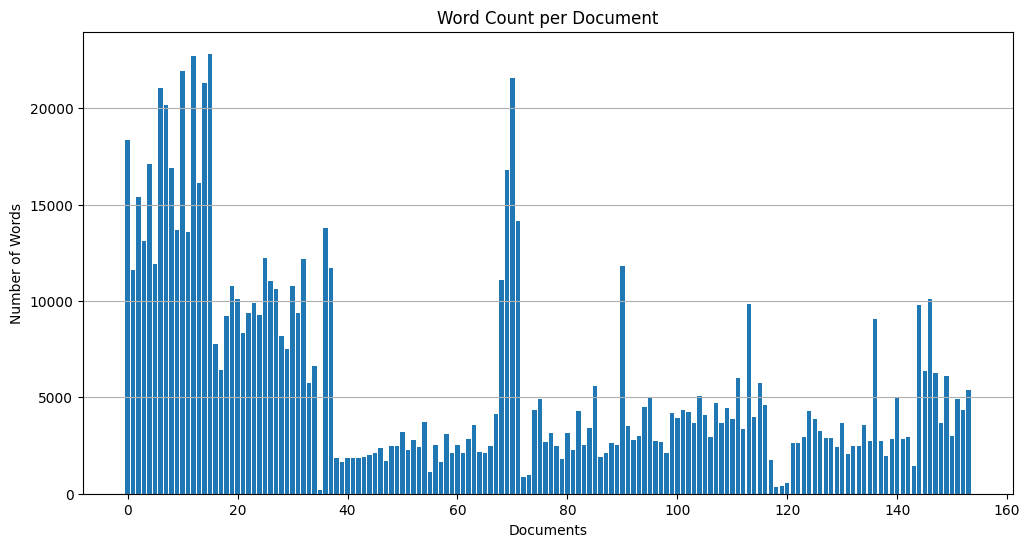
This project aims to explore and analyze transcripts from a collection of Arabic Spotify podcasts collected by Gasser Ali. The workflow involves loading the data from text files, conducting initial data exploration to understand the dataset's size and basic statistics, performing multiple preprocessing steps including lowercasing, tokenization, stemming, lemmatization, and stopword removal, and conducting exploratory data analysis using word frequency counts, TF-IDF, and co-occurrence analysis. Additionally, embeddings are visualized using t-SNE and UMAP, followed by clustering using KMeans.

Several design choices were made to optimize Arabic text processing. Libraries such as arabic-reshaper, python-bidi, camel\_tools, and NLTK’s ISRIStemmer were used to handle the complexities of Arabic script. The combination of stemming (ISRIStemmer) and lemmatization (Camel Tools Analyzer) ensures consistent word reduction, which is essential for morphological languages like Arabic. TF-IDF representation was chosen to emphasize discriminative terms by weighting words based on frequency and rarity across episodes. Dimensionality reduction techniques such as t-SNE and UMAP were used to visualize data. KMeans clustering was selected as a simple yet effective method to identify initial topic similarities among transcripts.  
  
By performing exploratory data analysis, we aimed to understand the underlying patterns in the transcripts, which would later support more advanced NLP tasks such as topic modeling.

## **2. Data Loading and Exploration**

The dataset was loaded from a designated folder in Google Drive, where the script identified text files and aggregated them into a list. Basic statistics, including the number of podcasts, total documents, and word counts per document, were extracted to gain an overview of the dataset.





Through this exploration, logging files and word counts helped identify anomalies such as extremely short or unusually long transcripts. For example, the episode “لیه الدایت بیبوظ - خسر کل فلوسه فی مطعم - کریم امساعیل - Foodcast 2” has the highest number of words with 22836, while the episode “إيه المشكلة في الشتيمة ؟” only contains 170 words. The word frequency per document averaged at 5993 words. Understanding the word count provided insights into whether shorter or longer transcripts dominated the dataset, which is crucial in shaping preprocessing and analysis strategies.

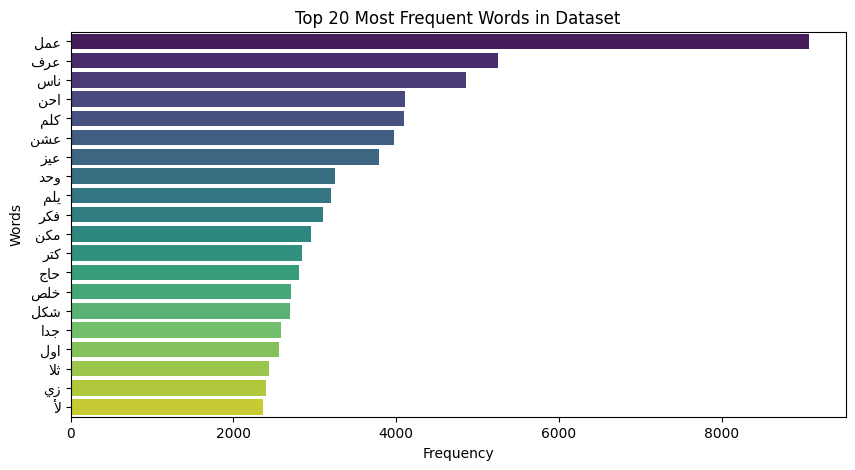
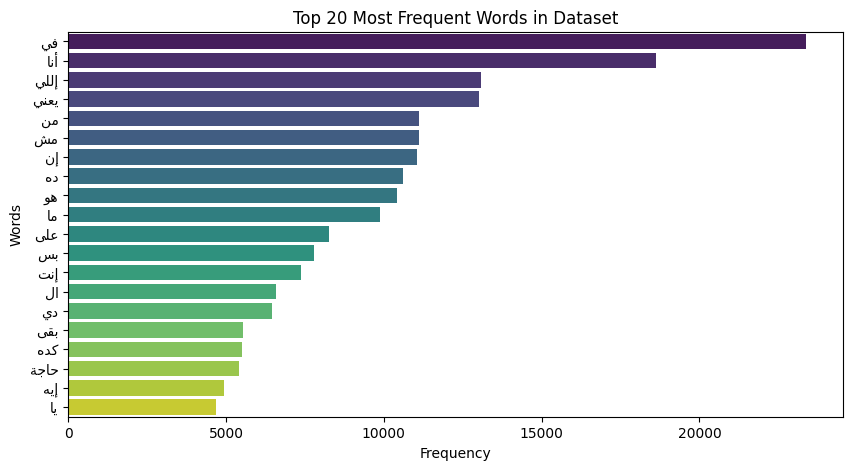
## **3. Data Preprocessing and Cleaning**

To standardize the text, all characters were converted to lowercase, ensuring consistency across variations of the same word. Tokenization was performed using NLTK’s word\_tokenize function, effectively splitting text into words and punctuation marks. Stemming was applied using ISRIStemmer to reduce inflected Arabic words to their root forms, while lemmatization with the Camel Tools morphological analyzer provided precise word normalization. Additionally, diacritics were removed using dediac\_ar to unify different forms of words. While lemmatization was more accurate, it took way more processing time than stemming. Thus, for the rest of the preprocessing stemmed words were used.

Stopword removal was carried out using a custom Arabic stopword list, supplemented with NLTK’s default Arabic stopwords, to eliminate common but non-informative words such as “أنا,” “مش,” and “ده.” We were required to add more stopwords to the NLTK’s default list because some words were misspelled or used the Egyptian dialect (eg. “أن” vs. “ان”). This multi-step preprocessing approach was necessary to address the morphological richness of Arabic. Iterative updates to the stopword list may be required to further refine the dataset based on exploratory analysis findings.

## **4. Exploratory Data Analysis**

Word frequency analysis was conducted by aggregating all tokens and calculating word frequencies. The most common words were visualized using bar plots, highlighting dominant terms across the dataset. Frequent word analysis provided insight into key themes while identifying residual non-informative tokens. The vocabulary size of the dataset was also analyzed to estimate corpus diversity and lexical richness.

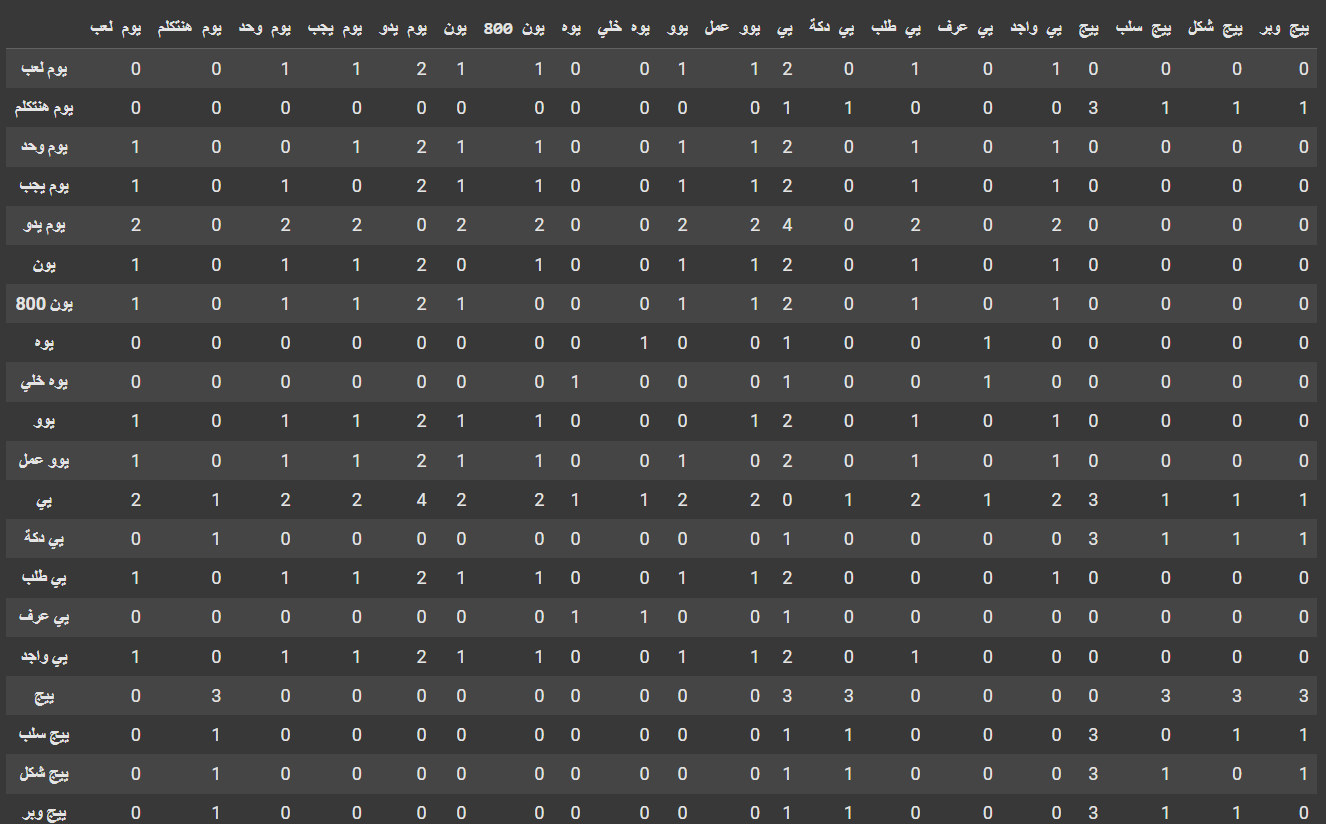


*Word Frequency Before and After Stopword Removal*

TF-IDF analysis was performed using TfidfVectorizer from sklearn, which transformed the text into weighted vector representations. This approach highlighted words that were uniquely significant within specific documents while down-weighting those that were frequently used across all documents. High TF-IDF scores revealed key terms that helped distinguish different episodes, whereas extremely high or low scores indicated the presence of domain-specific jargon or overly common stopwords.

*Snippet of the TF-IDF Matrix*

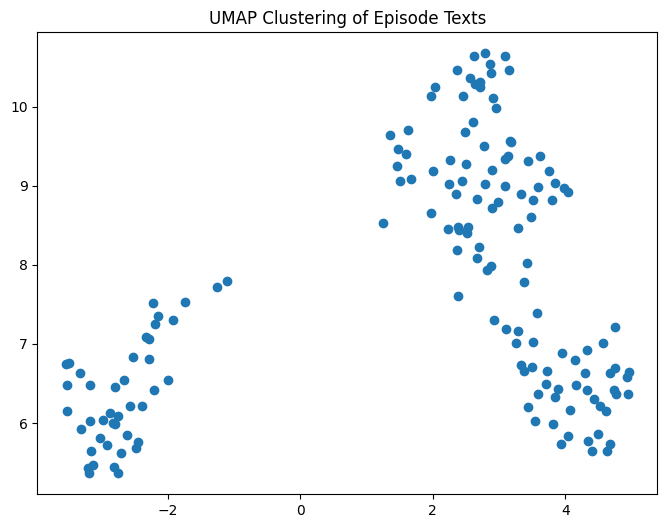
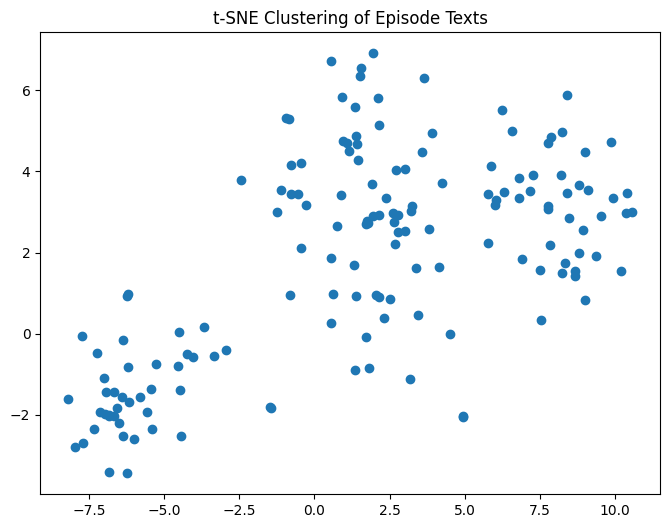
Co-occurrence analysis was conducted using CountVectorizer with n-grams, focusing on unigrams and bigrams to identify frequently occurring word pairs. This analysis helped detect commonly used phrases in spoken content, which is useful for recognizing idiomatic expressions or repeated patterns within the dataset.



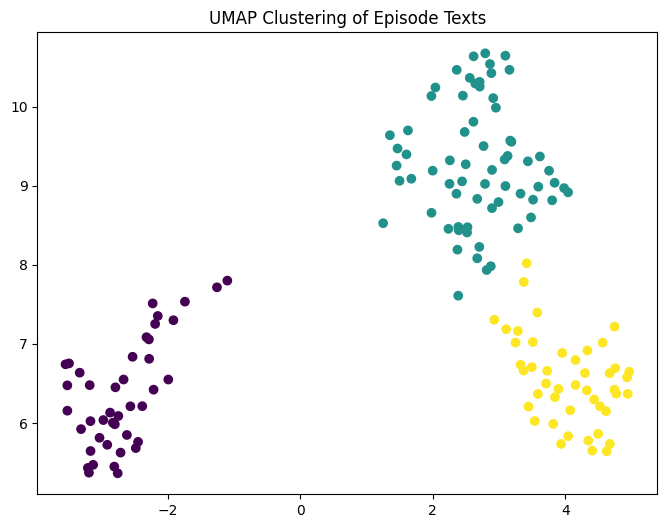
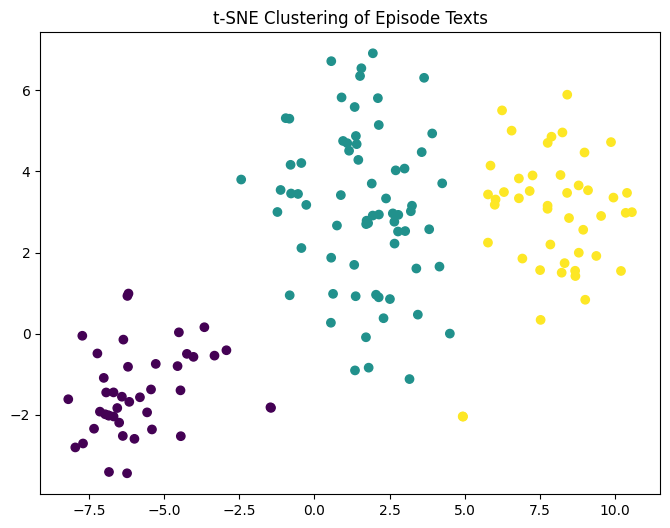
*Snippet of The Co-occurrence Matrix*

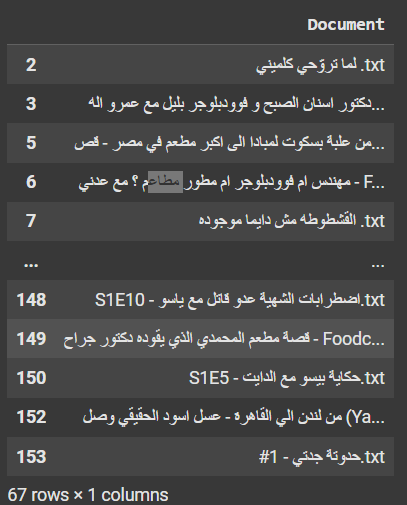
## **5. Dimensionality Reduction and Clustering**

Dimensionality reduction was applied using t-SNE and UMAP to visualize high-dimensional TF-IDF representations in two dimensions. Scatterplots generated from these embeddings helped reveal potential clusters within the topic distribution of transcripts. When clusters were not clearly defined, it indicated a high vocabulary overlap across episodes, suggesting additional preprocessing might be necessary.



KMeans clustering was used to group transcripts, based on their embeddings, into 3 clusters. The distribution of documents across clusters provided insights into thematic similarities.





The above images showcase the documents belonging to cluster 0, 1 and 2 respectively. Based on the episode titles in Cluster 0, it can be inferred that these transcripts primarily focus on movies and pop culture. Cluster 1 contains discussions related to cooking and food, highlighting themes around recipes and culinary techniques. Cluster 2, on the other hand, revolves around religious topics.

## **6. Limitations and Opportunities for Improvement**

The dataset contained a mix of dialectal Egyptian Arabic, Modern Standard Arabic (MSA), and English code-switching, which posed challenges for existing NLP tools optimized primarily for MSA. This language variation could impact the accuracy of preprocessing and clustering results. The stopword list, while extensive, required iterative refinement to ensure that it excluded only non-informative words while preserving meaningful content.

The dataset's size and diversity also influenced the effectiveness of dimensionality reduction techniques. Larger datasets typically yield more reliable results, whereas small datasets can lead to misleading clusters. KMeans clustering, while useful, assumes spherical cluster structures, which may not accurately reflect real-world text data. Exploring alternative methods such as hierarchical clustering, DBSCAN, or topic modeling (LDA) could improve clustering results. Additionally, the interpretability of extracted features required manual inspection, as automated clustering does not inherently assign meaningful labels to groups. Future iterations could benefit from integrating language models like AraBERT to enhance feature extraction and topic interpretation.

## **7. Conclusion**

This project successfully outlined the key steps for analyzing Arabic podcast transcripts. The use of Arabic morphological tools, stopword handling, TF-IDF analysis, and clustering provided a solid foundation for exploring the thematic structure of the dataset. However, refining techniques—especially for handling dialect-specific content—can further enhance the results.