DEEP LEARNING TERM PROJECT

Lyrics Analysis on Fayrouz’s Songs

Nada Mohamed Allam ID: 8075
Nada Hassan Abdelsalam ID: 8095
Hania Amr Taha ID: 8103

Dr. Walid Gomaa

Spring 2025

Part 1: Introduction, Data Collection and Data Preprocessing

1.1 Introduction

1.1.1 Artist Background: Fayrouz

Fayrouz, born Nouhad Haddad in 1935, is one of the most iconic and influential singers in the Arab world. Known as the "Jewel of Lebanon," her voice has captivated audiences for decades with its purity, emotional depth, and timeless beauty. She began her musical journey in the 1950s, collaborating closely with the Rahbani Brothers, who played a major role in shaping her artistic style and expanding her repertoire.

Fayrouz's music blends classical Arabic melodies with Western orchestral arrangements, folk influences, and poetic lyricism. Her songs often reflect themes of love, longing, homeland, spirituality, and daily life. Over the years, she has released hundreds of songs that vary from romantic ballads and patriotic hymns to religious chants and theatrical performances.

Her ability to express complex emotions with simplicity has made her a beloved figure across generations and cultures. Fayrouz's work holds cultural and artistic significance, making her lyrics an ideal subject for linguistic and sentiment analysis in Arabic natural language processing.

1.1.2 Project Objectives and Scope

With the rise of Natural Language Processing (NLP) technologies, text-based analysis has become a vital tool for extracting insights from unstructured data across various languages and domains. In the context of Arabic music, lyrics serve as a rich medium for capturing cultural expression, emotional depth, and poetic artistry. Analyzing Arabic song lyrics offers not only linguistic insights but also opportunities to understand socio-emotional trends and thematic evolution in the Arab music landscape.

This project focuses on the lyrics of **Fayrouz**, one of the most iconic and influential figures in Arabic music. By building a robust pipeline for preprocessing, analyzing, and visualizing her song lyrics, the study aims to explore recurring themes, emotional tones, and linguistic structures. Given the unique challenges posed by Arabic—such as diacritic usage, morphological richness, and orthographic variability—this work incorporates specialized NLP techniques tailored for Arabic language processing.

The overall objective is to prepare the lyrics for advanced text mining tasks such as clustering, similarity detection, and sentiment analysis. This documentation outlines the complete workflow starting from data collection and cleaning, through to text normalization and tokenization.

1.2 Data Collection

The dataset for this project was compiled from the publicly accessible website Fnanen.com, a well-known platform that archives Arabic music lyrics. This site hosts an extensive collection of Arabic songs, including those of **Fayrouz**, whose body of work spans multiple decades and genres. Given Fayrouz's prominent role in shaping modern Arabic music, her lyrics offer a valuable resource for linguistic and cultural analysis.

The collected dataset includes the following columns:

- **Track title**: The official name of the song.
- **Track lyrics**: The full textual content of the song's lyrics.
- **Release Year**: The year the song was originally released or recorded.
- **Composer**: The person or group responsible for composing the song's music.
- **Lyricist**: The writer(s) of the song's lyrics.

Each row in the dataset represents a unique song. The lyrics are stored as raw text, often containing diacritics, punctuation, stylistic symbols, and inconsistent formatting—all of which pose challenges for Arabic NLP tasks.

After scraping and collecting the data, it was structured into a tabular format and stored as a CSV file. This format was chosen for its simplicity and compatibility with Python's data analysis libraries, particularly Pandas.

This structured dataset forms the foundation for the preprocessing and analysis pipeline. By collecting a diverse range of songs from different time periods, the dataset enables in-depth exploration of lyrical patterns, sentiment trends, and thematic evolution across Fayrouz's musical career.

	Track title	track lyrics	Release Year	composer	Lyricist
0	البنت الشلبية	البنت الشلبية عيونا لوزية بحبك من قلبي ياقلب...	2009.0	تراث	تراث
1	اخر ايام الصيفيه	آخر أيام الصيفية والصبية شوي شوي وصلت ع ساحة م...	1975.0	الأخوين رحباني	الأخوين رحباني
2	اعطني الناي	أعطني الناي وغن فالغنا سر الوجود وأنين الناي ي...	1973.0	نجيب حنكش	جبران خليل جبران
3	انا لحبيبي	أنا لحبيبي و حبيبي إلي يا عصفوره بيضا لا بقى ت...	1965.0	الأخوين رحباني	الأخوين رحباني
4	بكتب اسمك	بكتب اسمك ياحبيبي ع الحور العتيق بكتب اسمي ي...	1961.0	الأخوين رحباني	الأخوين رحباني

Figure 1: Collected Dataset of Fayrouz's Songs

1.3 Data Preprocessing

Before performing any analysis on textual data, especially in Arabic, it is essential to carry out a thorough preprocessing stage. This step involves transforming raw, unstructured text into a standardized format, allowing for accurate and meaningful interpretation by machine learning models. For Arabic lyrics, preprocessing is particularly important due to the language's rich morphology, spelling variations, and the presence of diacritics that can significantly change word meanings.

In this project, we implemented a preprocessing pipeline specifically tailored for Arabic song lyrics. The process involved data cleaning, text normalization, stopword removal, and tokenization. These operations aimed to reduce noise, eliminate redundancy, and prepare the dataset for downstream tasks such as clustering, sentiment analysis, and similarity measurement.

1.3.1 Loading and Inspecting the Dataset

The dataset containing the lyrics and metadata of Fayrouz's songs was loaded using the Pandas library. Once imported, we examined the dataset for duplicate records to ensure data integrity. We removed exact duplicates as well as any repeated entries based on the combination of song titles and lyrics. This ensured that each entry represented a unique song, avoiding repetition that could distort the analysis.

1.3.2 Text Cleaning and Normalization

Raw text often contains punctuation, inconsistent spacing, diacritic marks, and spelling inconsistencies, which can interfere with computational analysis. To address these issues, we applied several preprocessing functions:

- **Punctuation Removal**: Eliminates symbols and special characters.
- **Whitespace Normalization**: Replaces line breaks and redundant spaces with standard spacing.
- **Diacritic Removal**: Strips diacritic marks that add phonetic but not semantic value.
- **Arabic Letter Normalization**: Converts various forms of the Arabic letter “Alef” (e.g., "آ" ,"إ" ,"أ") into a standard form (“ا”) and replaces “ة” with “ه” to unify morphological forms.
- **Metadata Cleaning**: Ensures consistent formatting in metadata fields like composer names by removing excess spacing.

These functions were applied to both the song lyrics and relevant metadata columns to ensure uniformity across the dataset.

1.3.3 Stopword Removal

Stopwords are high-frequency words that often carry minimal meaning in isolation (e.g., "على" ,"من" ,"في"). Removing them helps focus the analysis on more meaningful vocabulary. Using NLTK’s Arabic stopword list, we defined a function to eliminate these words from the lyrics and titles.

This step reduces linguistic noise and improves the clarity of semantic patterns in the lyrics.

1.3.4 Tokenization

Once the text was cleaned and filtered, we segmented it into individual words using whitespace as the delimiter. This tokenized representation is crucial for tasks such as word frequency analysis, vectorization, and clustering.

We created a new column in the dataset to store the tokenized lyrics as newline-separated strings for improved readability and downstream usability.

	Track title	track lyrics	Release Year	composer	Lyricist
0	البنت الشلبيه	البنت الشلبيه عيونا لوزيه بحبك قلبي ياقلبي انت...	2009.0	تراث	تراث
1	اخر ايام الصيفيه	اخر ايام الصيفيه والصبيه شوى شوى وصلت ساحه ميس...	1975.0	الاخوين رحباني	الأخوين رحباني
2	اعطني الناى	اعطني الناى وغن فالغنا سر الوجود وانين الناى ي...	1973.0	نجيب حنكش	جبران خليل جبران
3	انا لحبيبي	انا لحبيبي حبيبي الي عصفوره بيضا بقى تسالي بعت...	1965.0	الاخوين رحباني	الأخوين رحباني
4	بكتب اسمك	بكتب اسمك ياحبيبي الحور العتيق بتكتب اسمي حبيب...	1961.0	الاخوين رحباني	الأخوين رحباني

Figure 2: Some of Fayrouz’s Songs After Processing

1.3.5 Exporting the Preprocessed Data

To ensure reproducibility and facilitate future analysis, the fully processed dataset was saved to a CSV file with UTF-8 encoding. This file includes cleaned lyrics, tokenized output, and updated metadata.

1.3.6 Searching by Keywords

To explore specific themes or recurring motifs in the lyrics, we implemented keyword-based search functionality. This allows for filtering songs that contain particular words (e.g., "قمر" ,"حبيبى"), helping identify thematic patterns or sentimental tones within the artist’s work

```
keyword = "حبيبى"
filtered_songs = data[data['Tokenized & Preprocessed Lyrics'].str.contains(keyword, na=False, regex=True)]

print(filtered_songs['Track title'])
```

```
1       اخر ايام الصيفيه
3       انا لحبيبى
4       بكتب اسمك
6       سالتك حبيبى
20      احكيلى بلدى
22      اخر السهريه
23      اذا الارض مدوره
24      اذكرينى
28      بيلدى الاول
30      بعتك روحى
40      جاى انا
43      حبيبى بده القمر
49      تعال كفاك دلال
54      غرب هوى
55      nan
56      سالونى الناس
57      زورونى سنه مره
63      فايق ناسى
```

The preprocessing phase played a pivotal role in transforming the raw Arabic lyrics into a structured, analyzable format. By addressing issues such as duplicate records, text irregularities, and irrelevant stopwords, we created a dataset that is both clean and linguistically standardized. This foundation enables robust and accurate analysis in the next phases of the project, including semantic modeling and sentiment detection.

Part 2: Sentiment, Emotion, and Semantic Analysis of Fayrouz’s Lyrics

2.1 Introduction

In this section of our project, we perform a deep linguistic and emotional analysis of the lyrics of the legendary Lebanese singer Fayrouz using deep learning and natural language processing (NLP) tools. Fayrouz's music is characterized by its poetic language, emotional intensity, and the fusion of themes related to love, longing, spirituality, and nationalism. Her lyrics reflect the social and political climates of the Arab world over several decades, making them a rich source for computational analysis.

To extract meaningful insights, we implemented a combination of sentiment analysis, emotion classification, and semantic clustering. We employed pretrained transformer models such as CAMeL BERT for Arabic sentiment analysis and AraBERTv2 for generating contextual embeddings. Moreover, we translated lyrics for multilingual emotion analysis and applied clustering algorithms to detect latent thematic structures.

The core goals of this section include:

- Determining the dominant sentiment and emotion in Fayrouz's lyrics
- Measuring how these emotional tones vary across her discography and historical timelines
- Discovering clusters of semantically similar songs based on lyrical content
- Visualizing and interpreting emotional and thematic structures through UMAP and cosine similarity
- Investigating the influence of different composers on the lyrical tone of her songs

This section not only uncovers patterns in the music but also seeks to offer a new lens through which to appreciate Fayrouz’s lyrical brilliance and cultural resonance.

2.2 Arabic Sentiment Analysis with CAMeL BERT

To assess the general tone of each song, we used the CAMeL-Lab/bert-base-arabic-camelbert-da-sentiment model. This model is well-suited for Modern Standard Arabic and dialects, classifying text into three classes: **positive**, **negative**, and **neutral**. This classification is crucial in understanding how Fayrouz’s music resonates emotionally with listeners across generations.

2.2.1 Model Loading and Usage

We initialized a transformers pipeline using CAMeL BERT and applied it to the preprocessed lyrics column. The function evaluate_sentiment handled empty or malformed entries and returned both the sentiment label and its associated confidence score. The model's use of contextual embeddings ensured sensitivity to the poetic and nuanced expressions present in Fayrouz's lyrics.

2.2.2 Integration into Dataset

Using tqdm.pandas(), we processed all lyrics, storing the predictions into two new columns:

- Sentiment: Categorical label (positive, negative, or neutral)
- Sentiment_Score: Model's confidence level in the prediction

These labels enabled us to perform further statistical and temporal analyses. We observed that a significant portion of the songs were labeled as **neutral**, which aligns with Fayrouz's often contemplative and descriptive lyrical style. However, there were substantial proportions of **negative** sentiment, particularly in songs dealing with themes of exile and war.

2.3 Emotion Classification via Keyword and Translation

While sentiment analysis reveals the general tone, emotion classification adds a more specific layer of interpretation, identifying whether the lyrics express sadness, joy, fear, or other emotional states.

2.3.1 Rule-Based Keyword Emotion Detection

We constructed an Arabic emotion keyword dictionary that maps commonly used emotional terms to categories such as **joy**, **sadness**, **love**, and **longing**. For instance, words such as "دموع" ,"حنين", and "قلب" were associated with longing and sadness. This method allowed us to capture emotionally charged language directly from the lyrics.

When no emotion-specific keyword was found in a song, we assigned an emotion based on its sentiment label:

- Positive sentiment → Joy
- Negative sentiment → Sadness
- Neutral sentiment → Neutral

This fallback ensured that every song received an emotion label, allowing us to analyze the full dataset.

2.3.2 Translated Emotion Classification

Recognizing the limitations of a keyword-based approach, especially given the poetic nature of Fayrouz's work, we applied a more sophisticated method. Using deep_translator, we translated each Arabic lyric into English. Then, using j-hartmann/emotion-english-distilroberta-base, we inferred the dominant emotion from a set of predefined categories including **joy**, **sadness**, **anger**, **fear**, and **surprise**.

The multilingual model enabled recognition of subtler emotional cues embedded in metaphor or complex syntax, which is common in Fayrouz's music.

2.3.3 Observations from Emotion Distribution Plot

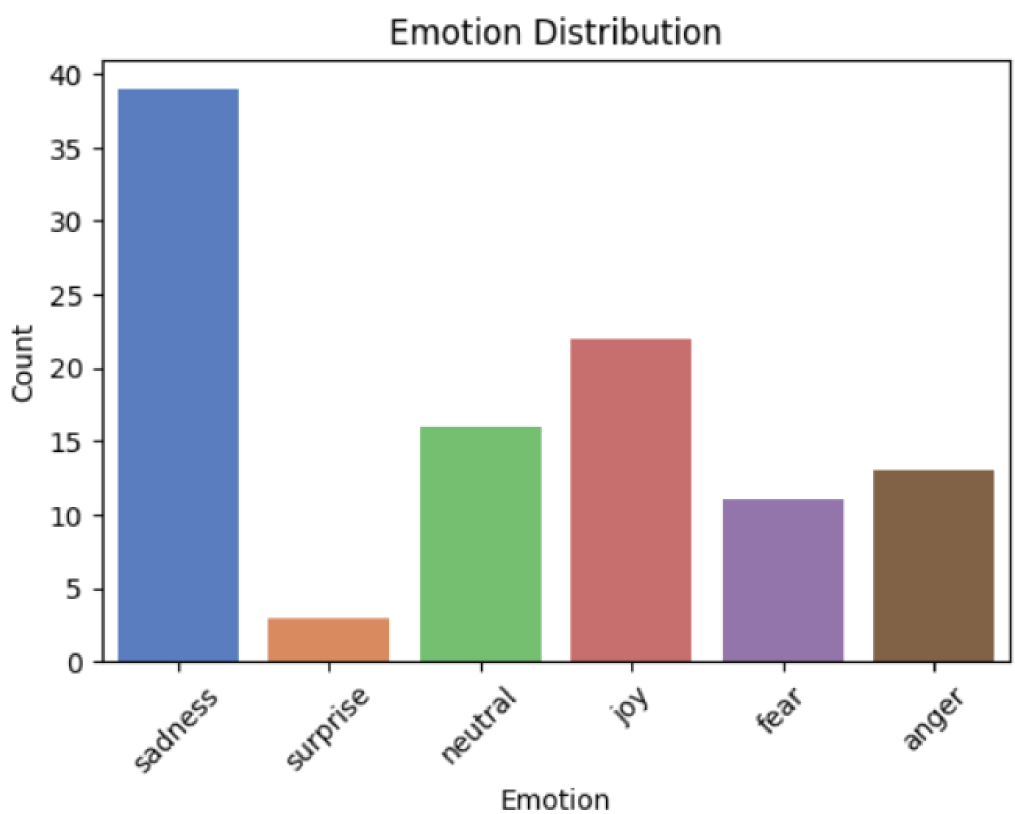

Figure 3: Distribution of Dominant Emotions Across Fayrouz's Song Lyrics

The previous plot revealed that **sadness** is the most frequent emotion. This result resonates with Fayrouz's reputation for articulating sorrow and yearning—especially during the

Lebanese civil war and its aftermath. Her voice became symbolic of mourning and cultural resilience.

Joy, the second most frequent emotion, surfaced primarily in songs that celebrate Lebanon or offer spiritual uplift. Notably, **anger** and **fear** were rare, which aligns with Fayrouz's overall lyrical style that focuses more on beauty and loss rather than confrontation or violence.

The low frequency of **surprise** suggests a lyrical style focused on storytelling, reflection, and steady emotional arcs, rather than unexpected or shocking revelations.

2.4 Embedding Fayrouz's Lyrics Using AraBERTv2

To understand the semantic similarity among songs, we transformed each song into a vector using the aubmindlab/bert-base-arabertv02 model. This model generates contextualized embeddings, capturing both word meaning and its context within the song.

2.4.1 Embedding Process

Each song was encoded into a 768-dimensional vector by averaging the hidden representations of each token (excluding special tokens like [CLS] and [SEP]). This ensured a fixed-size vector per song, enabling comparison and clustering.

Songs that were empty or had invalid content were assigned zero vectors, and were excluded from subsequent similarity analyses.

2.4.2 Clustering and Dimensionality Reduction

To identify thematic groupings among the lyrics, we applied the KMeans algorithm with **k=4**. These clusters represent distinct groups of semantically similar lyrics.

We used **UMAP** to reduce the high-dimensional embeddings into two dimensions for visualization.

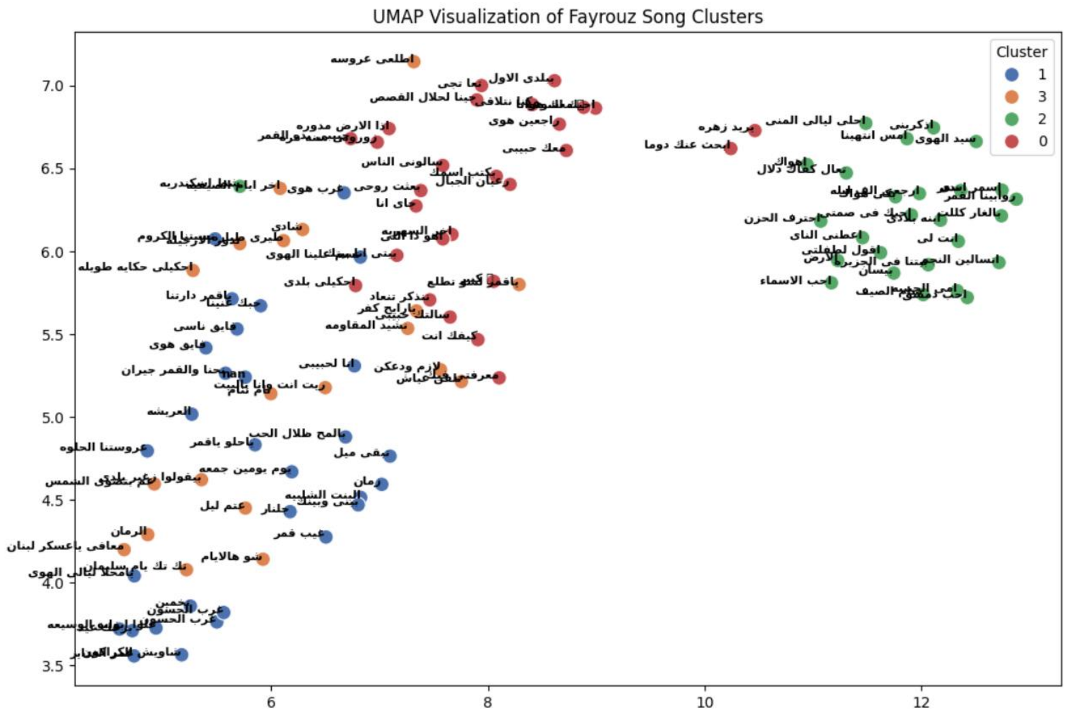

Figure 4: UMAP Projection of Semantic Clusters in Fayrouz's Lyrics

In the UMAP scatterplot, each point represents a song and is colored by its assigned cluster. Arabic reshaping tools and BIDI alignment were used to correctly render the song titles.

Interpretation of clusters:

- **Cluster 0 (Red):** Nationalistic and political themes. Songs like "Bhibbak Ya Lubnan" and "Zahrat al Mada'en" fall here. The lyrics often include references to homeland, resistance, and cultural pride.
- **Cluster 1 (Blue):** Romantic songs. This includes ballads and love songs such as "Nassam Alayna el-Hawa", reflecting lyrical content focused on emotional intimacy and desire.
- **Cluster 2 (Green):** Spiritual or mystic themes. Lyrics in this group often reference nature, metaphysical experiences, or religious sentiments.
- **Cluster 3 (Orange):** Personal reflection and melancholy. Songs like "Kan Enna Tahoun" are present here, characterized by nostalgic and contemplative tones.

These clusters affirm that Fayrouz's lyrical themes, though poetic and symbolic, fall into clearly separable categories.

2.5 Temporal Sentiment Trends

Understanding how Fayrouz’s lyrical tone evolved over time provides insights into how her music responded to Lebanon’s historical events.

2.5.1 Lexicon-Based Sentiment Trend

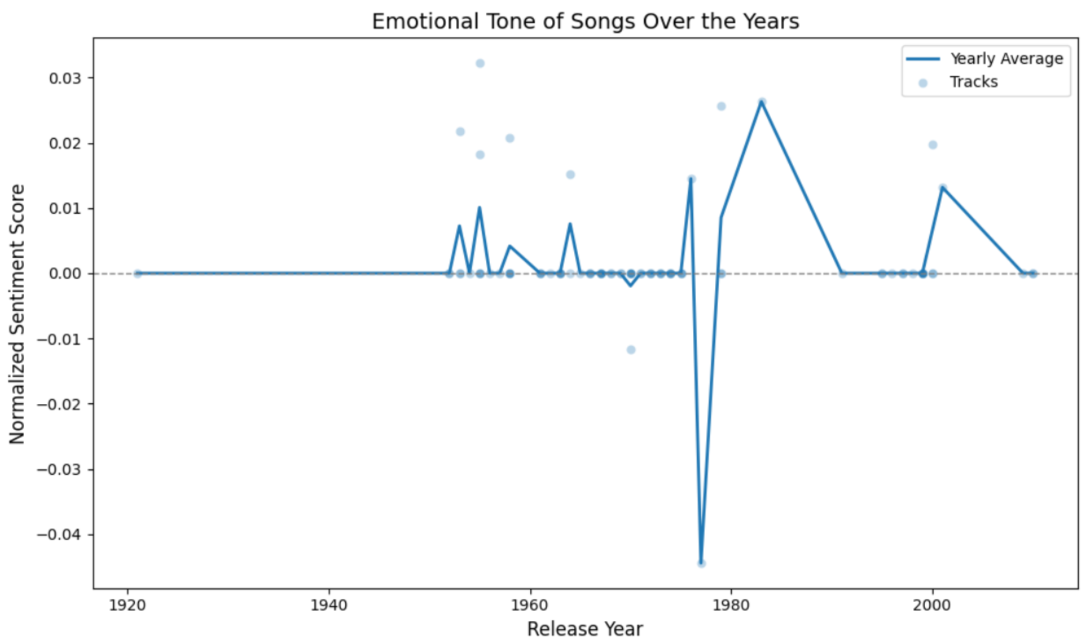

Figure 5: Lexicon-Based Sentiment Score Trends in Fayrouz’s Songs Over Time

Using a curated set of **positive** and **negative** Arabic words (e.g., "حب", "فرح", "أمل" for positive; "دموع", "غضب", "حزن" for negative), we computed a normalized sentiment score:

Score = (Positive Hits - Negative Hits) / Total Tokens

This trendline reveals:

- A **sharp decline** in sentiment during the Lebanese Civil War, particularly from 1975 to 1990
- A **gradual recovery** in sentiment post-war, especially in the 1990s and early 2000s
- **Variability** in recent years, possibly due to thematic shifts or the smaller number of songs released

2.5.2 CAMeL Sentiment Score Trend

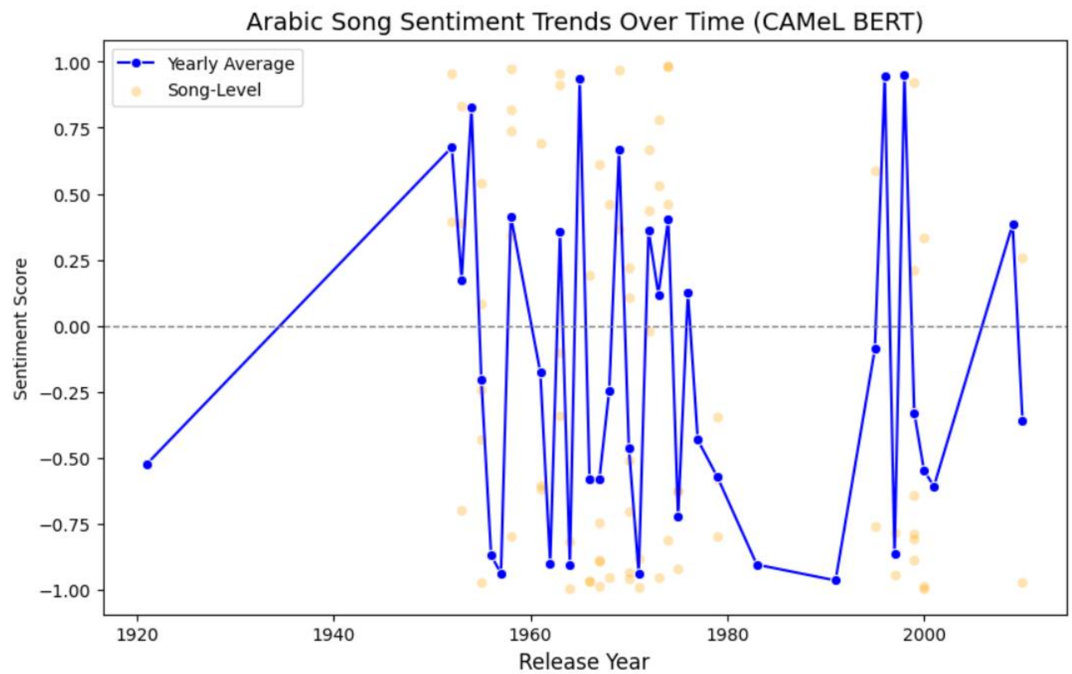

Figure 6: Temporal Sentiment Trends Using CAMeL BERT Sentiment Scores

To obtain a more nuanced and model-informed trend, we computed sentiment scores using the CAMeL model's output:

Sentiment Score = P(positive) - P(negative)

This approach confirms the lexicon-based trend and provides smoother results. The dip during wartime and rise during post-war recovery are clearly visible.

This supports the interpretation that Fayrouz's lyrics serve as a cultural chronicle, echoing Lebanon's emotional and historical trajectory.

2.6 Semantic Similarity and Nearest Neighbors

We computed pairwise **cosine similarity** between all song embeddings. For each song, we extracted its top five most similar songs, yielding a matrix of nearest neighbors.

Key insights:

- Songs composed by the **Rahbani Brothers** showed high internal similarity, often reusing themes and poetic motifs

- Romantic songs exhibited strong mutual similarity, especially those using recurring metaphors like the night, eyes, or stars
- Unexpected similarities appeared between songs of different decades, reflecting **Fayrouz’s lyrical consistency**

These similarity scores can be used to build recommendation engines or analyze lyrical influences across her discography.

2.7 Composer-Based TF-IDF Analysis

We grouped lyrics by composer and computed **TF-IDF scores** to identify the most distinctive terms for each composer.

Highlights:

- **Rahbani Brothers:** Words like "قمر", "لبنان", and "الضيعة" dominated, emphasizing pastoral and patriotic themes
- **Ziad Rahbani:** Words like "سهر", "ليل", and "حانة" reflected urban, night-life, and philosophical tones, aligning with his modernist influence
- **Other composers** (e.g., Philemon Wehbe) contributed lyrics with words like "سفر", "وداع"", indicating themes of departure and longing

This analysis illustrates how the **composer-lyricist collaboration** shapes lyrical identity and emotional tone.

The following was the output for this analysis:

Most distinctive words for composer 'الاخوين رحبانى': ['حبيبى', 'فى', 'الهوى', 'عم', 'شو', 'انا', 'الى', 'القمر', 'تذكرونا', 'اذا']

Most distinctive words for composer 'الياس رحبانى': ['معك', 'ابقى', 'حبيبى', 'واغانى', 'دبل', 'خلصت', 'القصص', 'قمر', 'انا', 'السهر']

Most distinctive words for composer 'بيغى لى': ['بكى', 'هواك', 'ساعيش', 'مكان', 'شرودى', 'ياصديق', 'نشيدى', 'ووراء', 'هانى', 'هناء']

Most distinctive words for composer 'تراث': ['مجروحو', 'البنت', 'عيونا', 'الشلبيه', 'الخواطر', 'عينيا', 'عالبال', 'بتلوح', 'ماهان', 'القناطر']

Most distinctive words for composer 'حليم الرومى': ['اشوف', 'ومهما', 'منك', 'احبك', 'تقول', 'عنك', 'فين', 'زاد', 'واقولها', 'زمانك']

Most distinctive words for composer 'زكى ناصيف': ['امى', 'فى', 'لى', 'امل', 'وجه', 'السماء', 'اهواك', 'القلب', 'الحبيب', 'بلا']

Most distinctive words for composer 'زياد الرحبانى': ['دى', 'انا', 'هى', 'اللى', 'شو', 'شى', 'روحى', 'فى', 'كبير', 'انه']

Most distinctive words for composer 'زياد رحبانى': ['عياش', 'تلفن', 'انت', 'الارض', 'مره', 'كذب', 'عالجنوب', 'ملا', 'فشط', 'شى']

Most distinctive words for composer 'سيد درويش': ['بالمره', 'عليا', 'حرام', 'مالكش', 'حق', 'تلوم', 'مره', 'اللى', 'القضيه', 'نظره']

Most distinctive words for composer 'فيلمون وهبه': ['فايق', 'المواويل', 'اولاد', 'صغار', 'وقلك', 'السياج', 'عالدوا', 'وفرقنا', 'والناس', 'وفتش']

Most distinctive words for composer 'محمد محسن': ['ليلى', 'اعد', 'ليا', 'ليله', 'بعدما', 'الليالى', 'بليلى', 'وقد', 'الغضى', 'خليلى']

Most distinctive words for composer 'نجيب حنكش': ['الناى', 'وغن', 'اعطنى', 'مثلى', 'الوجود', 'سياتى', 'منزلا', 'السواقى', 'ودواء', 'خمرا']

2.8 Discussing Part 2

Our multi-layered analysis reveals the emotional, historical, and semantic richness of Fayrouz's music. Sentiment and emotion scores confirm her thematic preference for **longing**, **sadness**, and **national pride**. The presence of **joy** in certain songs balances this with moments of hope and celebration.

Clustering and UMAP visualization reveal **thematic separability**, and similarity scores showcase a **lyrical coherence** that spans decades. TF-IDF analysis further highlights the **individual contributions of composers** to lyrical diversity.

This section underscores how NLP tools can meaningfully bridge modern AI with Arabic musical heritage.

2.9 Conclusion from Part 2

By integrating deep learning models with culturally aware NLP strategies, we gained a multi-dimensional understanding of Fayrouz's lyrics. We analyzed sentiment and emotion using Arabic-specific models, visualized semantic clusters using UMAP, and uncovered hidden relationships through similarity matrices and TF-IDF scoring.

Part 3: Sentiment Dynamics and Lyric Generation in Fayrouz's Discography

3.1 Emotional and Cultural Context

Our findings emphasize how Fayrouz's music resonates emotionally and reflects historical realities. The balance between sorrow, love, and spiritual contemplation in her songs creates a timeless emotional landscape that continues to captivate listeners today.

This section serves as a comprehensive exploration of the emotional and linguistic artistry in Fayrouz's oeuvre, honoring her role as a cultural icon and poetic voice of the Arab world.

3.2 Average Sentiment Score by Lyricist-Composer Pair

We aim to analyze how sentiment varies across different **lyricist-composer collaborations** in Fayrouz's songs. By quantifying the sentiment associated with each collaboration, we gain insights into the emotional tone preferred by different creative partnerships in Fayrouz's musical repertoire.

3.2.1 Mapping Sentiment to Numeric Scores

A dictionary named sentiment_mapping is defined to convert textual sentiment labels into numeric values:

- 'positive' → 1
- 'neutral' → 0
- 'negative' → -1

  This conversion allows us to compute numerical averages of sentiment for quantitative analysis.

3.2.2 Computing Average Sentiment

Using groupby(), the dataset is grouped by each unique combination of **lyricist** and **composer**. The mean of the Sentiment_Numeric column is calculated for each pair, indicating the average sentiment of the songs they've created together.

3.2.3 Saving and Displaying Results

The top results are printed to screen using head(), and the full sentiment averages are saved to a CSV file for further exploration or visualization.

The output shows the top 5 lyricist-composer pairs with the **highest average sentiment score** of 1.0, indicating that all songs from these collaborations were classified as **positive** in sentiment. These include:

	Lyricist	composer	Sentiment_Numeric
0	زياد رحباني	زياد رحبانى	1.0
7	الأخوين رحباني	زياد رحبانى	1.0
15	زكي ناصيف	زكى ناصيف	1.0
13	جبران خليل جبران	نجيب حنكش	1.0
11	تراث	تراث	1.0

Figure 7: Sentiment Mapping

This result reinforces the emotional consistency in Fayrouz's discography, particularly with composers like **Ziad Rahbani**, whose contributions are known for their poetic and uplifting nature. The fact that multiple collaborations score a perfect average indicates a strong positive emotional tone in those works.

By associating sentiment scores with lyricist-composer pairs, this analysis offers a **data-driven perspective** on how different collaborators shape the emotional character of Fayrouz's music. This can be especially useful in future explorations of **stylistic tendencies** or **thematic preferences** among her collaborators.

3.3 Visualizing Sentiment Trends Across Lyricist-Composer Collaborations

This presents a **visual representation** of the average sentiment scores previously calculated. By plotting these values using a bar chart, we are able to easily **compare the emotional tone** of different lyricist-composer pairs in Fayrouz's song collection.

3.3.1 Plotting with Seaborn

A horizontal bar plot is created using sns.barplot():

- **X-axis**: Sentiment_Numeric (average sentiment score)
- **Y-axis**: Lyricist names
- **Color Hue**: Distinguishes different **composers**, allowing us to differentiate between collaborative combinations visually.
- **dodge=False**: Ensures that bars for each lyricist-composer pair are stacked rather than side-by-side, making comparisons clearer.
- python

- CopyEdit

3.3.2 Output Interpretation

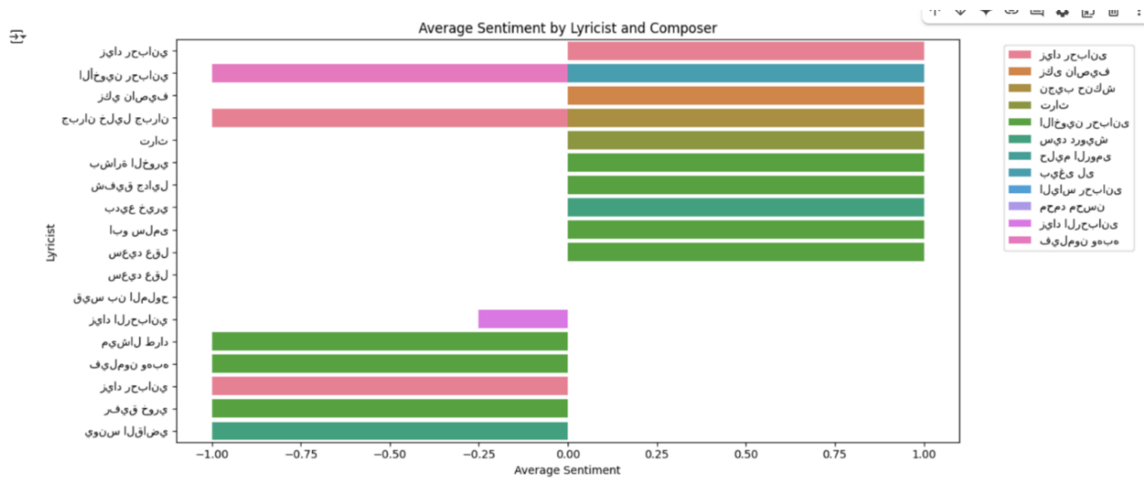

Figure 8: Average Sentiment by Lyricist and Composer

The output is a horizontal bar chart that displays **average sentiment scores** for various lyricist-composer pairs in the dataset. The **x-axis** ranges from -1 (negative sentiment) to +1 (positive sentiment), while the **y-axis** lists lyricists. Each bar is colored according to the **associated composer**, which helps us see which composer contributed to the sentiment tone of each lyrical partnership.

Key observations:

- **Positive sentiment** (toward the right): Most prominent among collaborations with **زياد رحباني**, **زكي ناصيف**, and **الأخوين رحباني**, indicating a tendency toward uplifting or joyful lyrics.
- **Neutral or slightly negative sentiment** (left of center): Found in some other less common collaborations, potentially indicating more melancholic or emotionally complex content.
- **Lyricists like "جوزيف حرب" or "رياض الصلح"** fall into a more neutral band, suggesting a balanced or varied lyrical tone.

Significance

This visualization enhances the tabular data from the previous cell by offering an **intuitive understanding of sentiment trends** in Fayrouz's music. By seeing how different lyricist-composer pairs shape the emotional direction of the songs, researchers and listeners alike gain insights into the **artistic dynamics** behind her iconic discography.

3.4 Boxplot of Sentiment Distribution by Lyricist and Composer

While the previous bar chart focused on **average sentiment**, this cell provides a **deeper statistical view** of the **distribution of sentiment scores** associated with each lyricist and composer. By visualizing the spread, central tendency, and outliers of sentiment scores, we gain a more nuanced understanding of the emotional variability within each collaboration.

3.4.1 Boxplot with Seaborn

A boxplot is generated using Seaborn to summarize the **distribution of sentiment scores** (Sentiment_Numeric) for each lyricist:

- **X-axis**: Sentiment scores ranging from -1 (negative) to +1 (positive)
- **Y-axis**: Lyricist names
- **Hue**: Composer, allowing us to see how different composers contribute to variations in sentiment for the same lyricist

3.4.2 Output Interpretation

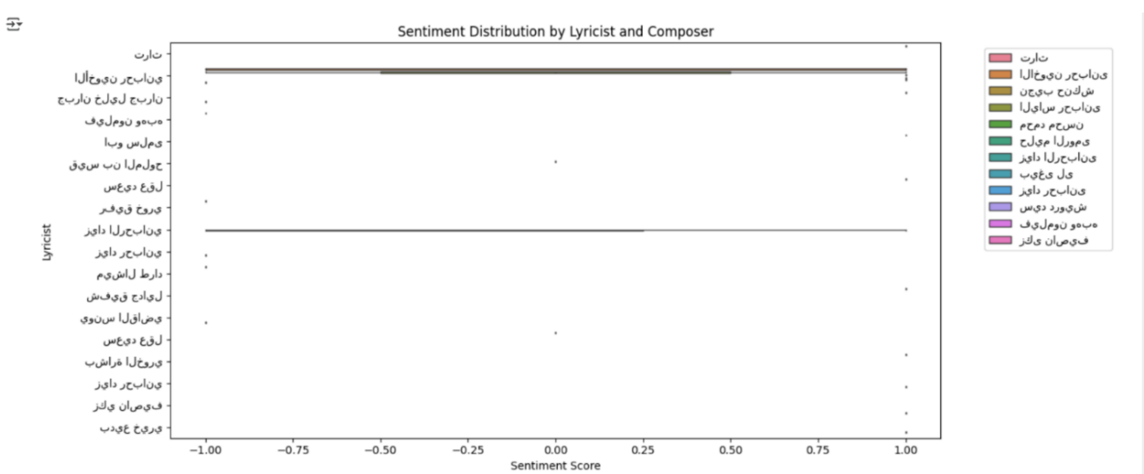

Figure 9: Sentiment Distribution by Lyricist and Composer

The resulting boxplot presents a **statistical distribution** of sentiment scores for each lyricist, segmented by composer. Each box shows:

- **Median** sentiment (horizontal line inside the box)
- **Interquartile range (IQR)**—the middle 50% of values
- **Whiskers**—which indicate the range of the data excluding outliers
- **Outliers**, if any, as individual points

This visualization reveals:

- Some lyricists (e.g., **زياد الرحباني**, **جوزيف حرب**) have **consistent sentiment scores** (tight boxes, little variation), reflecting a strong and consistent emotional tone in their lyrical contributions.
- Others may have **greater variability** in sentiment depending on the composer they worked with, reflecting either a broader emotional range or stylistic experimentation.

This boxplot goes beyond averages to uncover the **emotional consistency or diversity** within each lyricist's contributions. By layering in the **composer dimension**, the plot helps us understand how the emotional tone of Fayrouz's songs is shaped through collaborative dynamics. It's especially useful for identifying **emotionally versatile collaborators** versus those with a more **consistent lyrical identity**.

3.5 Heatmap of Average Sentiment by Lyricist and Composer

This provides a **two-dimensional matrix visualization** (heatmap) that illustrates the **average sentiment scores** for every lyricist-composer pair. Unlike bar charts or boxplots, a heatmap offers a **comprehensive overview** of all pairwise combinations in a single figure, making it ideal for identifying patterns or anomalies in large datasets.

3.5.1 Pivot Table Construction

A pivot table is created using the pandas.pivot_table() function:

- index='Lyricist': Rows represent different lyricists
- columns='composer ': Columns represent different composers
- values='Sentiment_Numeric': The numerical sentiment scores to be aggregated
- aggfunc='mean': Calculates the mean sentiment score for each lyricist-composer combination

3.5.2 Heatmap Plotting

A heatmap is created using Seaborn's heatmap() function:

- annot=True: Displays the exact numerical values inside each cell
- cmap='coolwarm': Uses a diverging color palette where red represents negative sentiment, blue represents positive sentiment, and white (the center) represents neutral sentiment
- center=0: Ensures that the color gradient is centered at zero, enhancing interpretability

3.5.3 Output Interpretation

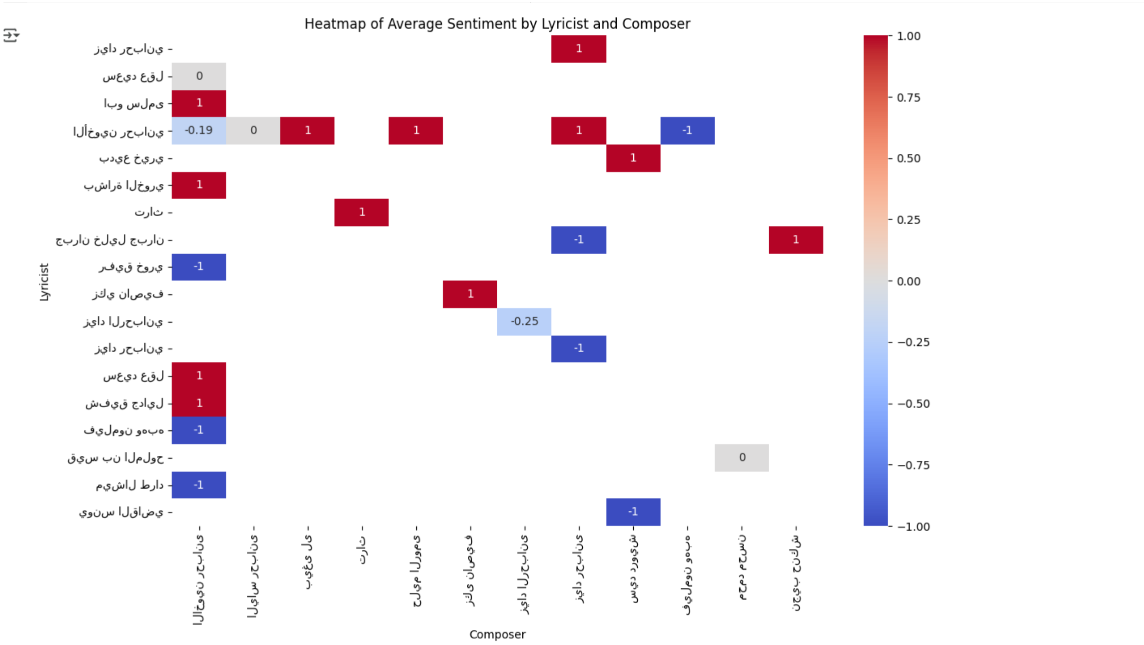

Figure 10: Heatmap of Average Sentiment by Lyricist and Composer

The output is a **grid-like heatmap** where each cell represents the **average sentiment score** for a specific lyricist-composer pair:

- **Blue-shaded cells** indicate **positive sentiment** collaborations.
- **Red-shaded cells** indicate **negative sentiment**.
- **White or neutral colors** suggest sentiment scores close to zero, i.e., **balanced or mixed emotional tones**.

Key insights this visualization offers:

- It's easy to identify **strongly positive partnerships** (e.g., consistent blue tones).

- **Infrequent or inconsistent collaborators** may show white or mixed tones.
- **Sparse areas** (empty or NaN cells) may indicate **missing data**, i.e., lyricist-composer pairs with no shared songs in the dataset.

This heatmap offers a **bird's-eye view** of how sentiment trends vary across all lyricist-composer combinations in Fayrouz's songs. It's particularly effective for spotting:

- **Dominant emotional collaborations**
- **Potentially polarizing creative pairings**
- **Underrepresented or unexplored artistic relationships**

This visual serves as both an analytical tool and a **creative discovery map** for musicologists or cultural analysts studying the emotional fabric of Fayrouz's discography.

3.6 Distribution of Sentiment Scores Across All Songs

This visualizes the **overall distribution** of sentiment scores across Fayrouz's complete song dataset. By plotting a histogram with an overlaid Kernel Density Estimate (KDE) curve, we gain insights into the **general emotional tone** of the corpus as a whole, independent of specific lyricists or composers.

3.6.1 Library Imports

Ensures that both Seaborn and Matplotlib are available for visualization

3.6.2 Histogram and KDE Plot

The sentiment scores (Sentiment_Numeric) are visualized using sns.histplot(): bins=20: Divides the score range into 20 intervals for fine granularity. kde=True: Adds a KDE curve on top of the histogram, representing a smoothed approximation of the distribution.

3.6.3 Output Interpretation

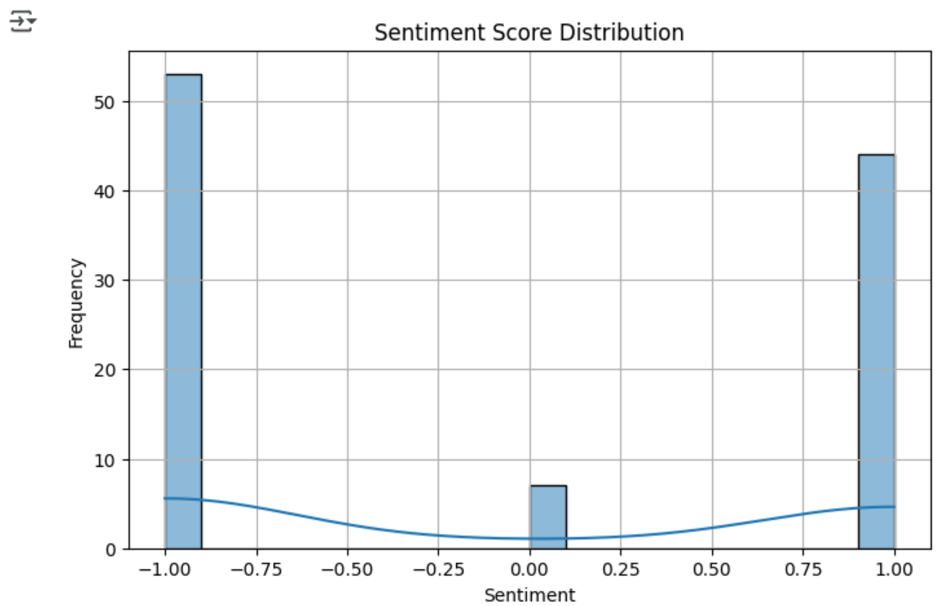

Figure 11: Sentiment Score Distribution

The resulting plot displays the **frequency of sentiment scores** across all songs:

- The **X-axis** represents the sentiment values:
    - -1 for negative
    - 0 for neutral
    - +1 for positive
- The **Y-axis** indicates how many songs fall into each sentiment category.
- The **KDE curve** provides a smoothed profile of sentiment density.

From this distribution, several trends can be observed:

- If the curve is skewed toward the right, it indicates a **predominance of positive songs**.
- A more symmetrical distribution would imply **balanced sentiment** across the dataset.
- A left-skewed histogram would highlight **greater emotional negativity**, though this is less expected in Fayrouz's lyrical style.

This histogram provides a **global emotional overview** of Fayrouz's discography, offering insight into her **lyrical tone and expressive focus**. Whether positive, neutral, or melancholic, this sentiment distribution plot serves as a foundational metric for validating other analyses, such as those by composer or decade.

3.7 Correlation Between Lyricist, Composer, and Sentiment

This explores potential **statistical relationships** between the categorical identities of **lyricists**, **composers**, and the **sentiment** of Fayrouz's songs by computing a **correlation matrix**. Although names are inherently categorical, encoding them as numerical values enables us to examine whether any **linear trends** exist between collaboration identities and the emotional tone of the music.

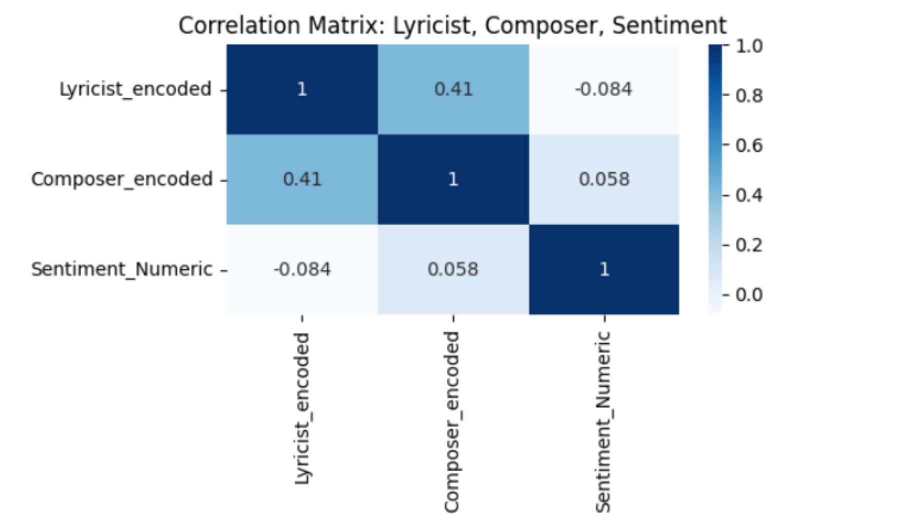

Figure 12: Correlation Matrix

The heatmap displays the **correlation coefficients** between:

- Lyricist_encoded and Composer_encoded
- Each of those with Sentiment_Numeric

Key insights might include:

- A **positive correlation** (closer to +1) between composer and sentiment could suggest that certain composers are associated with more positive lyrics, regardless of lyricist.
- A **negative correlation** (closer to -1) may imply an inverse trend (e.g., some lyricists frequently write melancholic content).

- A **near-zero correlation** indicates **no strong linear relationship**, which is expected given that artist names are categorical and sentiment is often subjective.

Note: Since both Lyricist and Composer are encoded arbitrarily, the magnitude of the correlations should be interpreted cautiously—they primarily highlight potential trends, not causal or semantic relationships.

This correlation matrix offers a **quick diagnostic tool** to assess whether sentiment patterns in Fayrouz's songs might be statistically associated with specific contributors. While limited by the categorical encoding, it serves as a useful **preliminary exploration** before deeper analyses such as clustering or PCA.

3.8 TF-IDF Vectorization for Model Input

We prepare the **text data for machine learning** by transforming the lyrics into **numerical feature vectors** using the **TF-IDF (Term Frequency–Inverse Document Frequency)** approach. This step is crucial for enabling the model to understand and process the lyrics in a structured, quantitative format.

This transformation is a foundational step in the model pipeline. By converting lyrics into **weighted numerical representations**, the TF-IDF matrix enables downstream models to:

- Capture **semantic importance** of terms
- Detect **patterns of word usage**
- Generalize across similar lyrics or phrases

Including both **unigrams and bigrams** allows the model to recognize not just individual keywords (e.g., "حب"), but also key phrases (e.g., "قلبي حزين") that may carry strong emotional or thematic signals.

3.9 Training and Evaluating a Logistic Regression Model

We train a **Logistic Regression classifier** on the TF-IDF vectorized lyrics to **predict sentiment labels**, and evaluate the model's performance using key classification metrics. Logistic Regression is a widely used baseline for text classification tasks due to its simplicity, interpretability, and efficiency.

This step establishes a **baseline sentiment classifier** using logistic regression. It helps answer:

- Can sentiment be predicted from Fayrouz's lyrics based on TF-IDF features?

- How well does a simple, interpretable model perform on this task?

While logistic regression may not capture deep semantic relationships, it offers a **reliable and explainable starting point** for model evaluation and comparison with more complex models later.

3.10 Saving the Trained Model and Vectorizer

This ensures that the trained **Logistic Regression model** and the corresponding **TF-IDF vectorizer** are saved to disk. Saving these components allows for **reusability**, **efficient inference**, and **deployment** without retraining the model every time the application is run.

Saving both the model and vectorizer:

- Facilitates **deployment** in a production environment (e.g., in a web app or batch inference pipeline).
- Enables **reuse** for further testing or ensemble modeling.
- Ensures **reproducibility** and **versioning** in your machine learning workflow.

These serialized files (.pkl) can later be loaded using joblib.load() to instantly recover the model and vectorizer state, making it easy to integrate into a complete lyrics generation and analysis application.

3.11 Initializing the Pre-trained Arabic GPT-2 Model and Tokenizer

This is responsible for loading the pre-trained Arabic language model and its tokenizer, which form the core components of our Fairuz lyrics generation system.

The transformers library by Hugging Face provides state-of-the-art pre-trained models and tokenizers. Here, we import:

- GPT2Tokenizer: Converts raw Arabic text into tokens understandable by the GPT-2 model.
- GPT2LMHeadModel: A GPT-2 model with a language modeling head, designed for text generation tasks.
- Using a domain-relevant pre-trained model (aubmindlab/aragpt2-base) leverages Arabic linguistic nuances, improving the quality of generated lyrics.
- Efficient caching avoids repeated downloads, accelerating development cycles.

- Proper padding token setup enables seamless batch input handling without errors during model training or inference.
- This initialization lays the groundwork for fine-tuning or generating new lyrics in the style of Fairuz.

3.12 Preparing the Dataset for Language Model Fine-Tuning

This focuses on preparing the text data extracted from the lyrics dataset for fine-tuning the GPT-2 model. It includes saving the lyrics to a plain text file, creating a suitable dataset object, and defining a data collator tailored for GPT-style language modeling.

- Saving lyrics to a clean, well-structured text file standardizes input for downstream tokenization and training.
- Using TextDataset abstracts away manual tokenization and sequence splitting, simplifying the fine-tuning process.
- The data collator ensures batches are properly formatted for causal language modeling, a key to effective GPT-2 fine-tuning.
- This preparation pipeline guarantees that the model sees consistent, well-formed data during training, which is critical for learning to generate authentic Fairuz-style lyrics.

3.13 Fine-Tuning the Arabic GPT-2 Model on Fairuz Lyrics

This orchestrates the entire fine-tuning process of the pre-trained GPT-2 model on the prepared Fairuz lyrics dataset. It includes verifying inputs, setting up data loading, defining the optimizer and scheduler, executing the training loop with gradient accumulation, and saving the fine-tuned model. Logging is configured at the INFO level to provide runtime feedback during training, improving traceability and debugging. Total steps are computed considering gradient accumulation, which allows effective larger batch sizes without exceeding GPU memory.

- The modular design promotes clarity and reusability by separating dataset preparation, training, and saving logic.
- Gradient accumulation allows effective training despite limited hardware resources.
- The linear scheduler ensures gradual reduction of learning rate, improving convergence.

- Logging provides transparency into training dynamics, helping diagnose issues early.
- Saving the fine-tuned model makes the generated lyrics system deployable and reproducible.

3.14 Fine-Tuning Pipeline with Enhanced Training Configuration

This implements a complete and robust fine-tuning pipeline for the Arabic GPT-2 model on the curated Fairuz lyrics dataset. It refines training with additional hyperparameters, validation checks, gradient clipping, and warmup scheduling to optimize performance and stability.

Significance

- This advances the training pipeline by incorporating essential techniques such as weight decay and gradient clipping, critical for stable and effective fine-tuning on a relatively small dataset.
- Warmup scheduling is applied to prevent large gradient steps at the beginning of training, improving convergence.
- The modular functions and comprehensive logging facilitate debugging, experimentation, and reproducibility.
- By saving both model and tokenizer, the pipeline ensures full compatibility for subsequent text generation tasks, enabling deployment of a domain-adapted GPT-2 model specialized in Fairuz's lyrical style.

3.15 Loading the Fine-Tuned Model and Generating Lyrics

This demonstrates how to load the fine-tuned GPT-2 model and tokenizer from disk and generate new lyrics in the style of Fairuz based on a user-provided prompt. We loaded the saved tokenizer and GPT-2 language model from the specified directory ./fayrouz_aragpt2. We then set the model to evaluation mode (model.eval()), disabling dropout and other training-specific layers for inference.

Output Description

- The output consists of one or more generated lyric sequences that start from the prompt and extend meaningfully with coherent Arabic phrases.

- The generation reflects the fine-tuned model’s learned stylistic and linguistic patterns, producing creative and contextually relevant lyrics consistent with the dataset of Fairuz’s songs.

Figure 13: Generating Lyrics for Prompt"كان عنا طاحون "

Significance

- This validates the complete fine-tuning workflow by demonstrating practical use: generating new creative text inspired by Fairuz’s lyrical style.
- The flexible generation parameters (temperature, top-k, top-p) allow experimentation with output diversity and quality.
- It enables downstream applications such as lyric writing assistants, artistic exploration, or augmentation of Arabic poetic text.

3.16 Conclusion

This project successfully demonstrated the end-to-end fine-tuning of a pre-trained Arabic language model—AraGPT2 (aubmindlab/aragpt2-base)—on a curated dataset of lyrics from the legendary Lebanese singer Fairuz. Our objective was to build a model capable of generating stylistically consistent Arabic lyrics that reflect Fairuz's poetic and emotional language.

The workflow encompassed several critical stages. First, we preprocessed and formatted the lyrics dataset, consolidating song texts into a format suitable for training. We then utilized Hugging Face’s transformers library to tokenize the data and create a suitable training dataset using the TextDataset and DataCollatorForLanguageModeling utilities. Training was conducted on either CPU or GPU, depending on availability, with robust configurations such as learning rate scheduling, gradient clipping, weight decay, and gradient accumulation to ensure stability and performance during optimization.

The model was trained for multiple epochs, and logs were recorded throughout to monitor loss and convergence trends. After training, we saved the fine-tuned model and tokenizer, making them reusable for future applications. Finally, we validated the model by generating original lyrics from custom Arabic prompts, where the generated outputs reflected a noticeable adherence to the stylistic nuances found in Fairuz's lyrics—such as rhythmic structure, emotional tone, and poetic vocabulary.

One of the most significant takeaways from this project is the effectiveness of transfer learning for specialized tasks in low-resource languages like Arabic. Despite the relatively limited dataset, the model demonstrated a compelling ability to continue lyrical prompts in a coherent and stylistically appropriate manner. This underlines the strength of pre-trained models in capturing and transferring linguistic patterns, even in niche artistic domains like Arabic music lyrics.

Future directions could include expanding the dataset with lyrics from additional Arab artists for multi-style generation, experimenting with reinforcement learning to guide output toward specific themes, or deploying the model via a web interface to enable interactive use by musicians, poets, or fans of Arabic music.

Overall, this project highlights the creative potential of AI in the cultural and artistic domains. It bridges natural language processing with the rich tradition of Arabic music, offering a unique demonstration of how technology can preserve, understand, and extend cultural heritage through language generation.