Arabic Lyrics Generation

Mahmoud Hesham Shehabeldin Solyman Mohamed Sonbol

May 7, 2023

Abstract

This report explores the use of natural language processing (NLP) to generate Arabic lyrics using the HABIBI dataset, a collection of over 30,000 Arabic songs across six distinct dialects. The report outlines the data collection and preprocessing steps, word embedding, model selection and fine-tuning, and lyrics generation. The aim of the project is to showcase the potential of NLP in generating culturally nuanced and linguistically diverse lyrics that resonate with Arabic speakers worldwide.

1 Introduction

In recent years, natural language processing (NLP) has made significant advancements in text generation, and the development of language models like GPT-3 and BART have revolutionized the field. One area where these models can be particularly useful is in generating lyrics for music in different languages. In this project, we focus on generating Arabic lyrics using the HABIBI dataset, which is a collection of over 30,000 Arabic songs across six distinct dialects. Our approach involves preprocessing the data, performing word embedding, selecting a suitable model, fine-tuning it, and generating creative and original Arabic lyrics. The aim of this project is to showcase the potential of NLP in generating culturally nuanced and linguistically diverse lyrics that resonate with Arabic speakers worldwide.

2 Dataset

In this NLP project, we utilized the HABIBI [1] dataset, which serves as the foundation for our Arabic lyrics generator. The HABIBI dataset comprises a comprehensive collection of Arabic songs, encompassing a wide range of genres and styles. It consists of a total of 30,072 Arabic songs performed by 1,755 different singers and encompasses lyrics in six distinct dialects.

The dataset provides a diverse representation of Arabic music, capturing variations in dialect, singer, composer, song writer, lyrics, singer nationality and the song title. The six dialects forementioned are Meghribi, Gulf, Iraqi, Sudan, Egyptian and Levantine. By incorporating songs from various dialects, we aimed to ensure the generator's ability to produce lyrics that resonate with Arabic speakers from different regions and linguistic backgrounds.

Before going into the results and analysis, it is critical to understand the dataset's distribution between dialects. The pie chart below depicts the dataset's dialect makeup in terms of dialects. The collection includes lyrics from six diverse Arabic dialects, each of which contributes to the corpus' overall diversity and richness. We obtain insights into the language variances and cultural subtleties that our Arabic lyrics generator will capture by researching the dialect distribution.

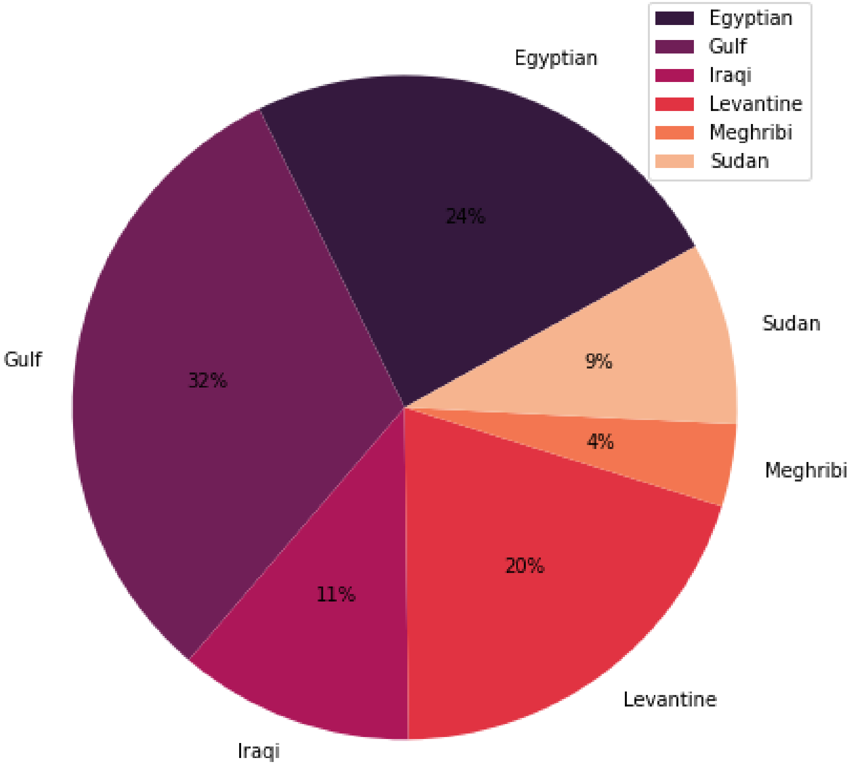
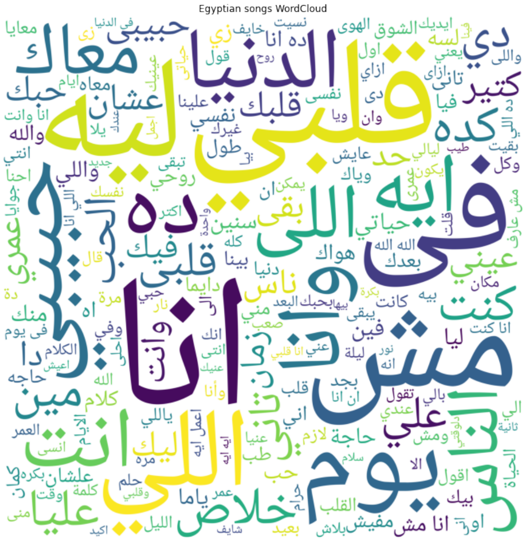
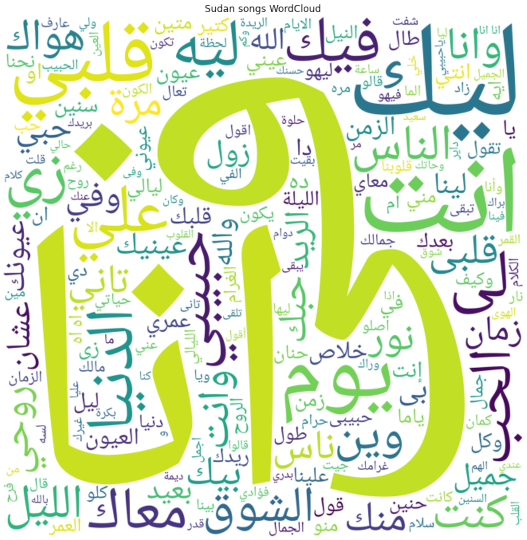


Figure 1

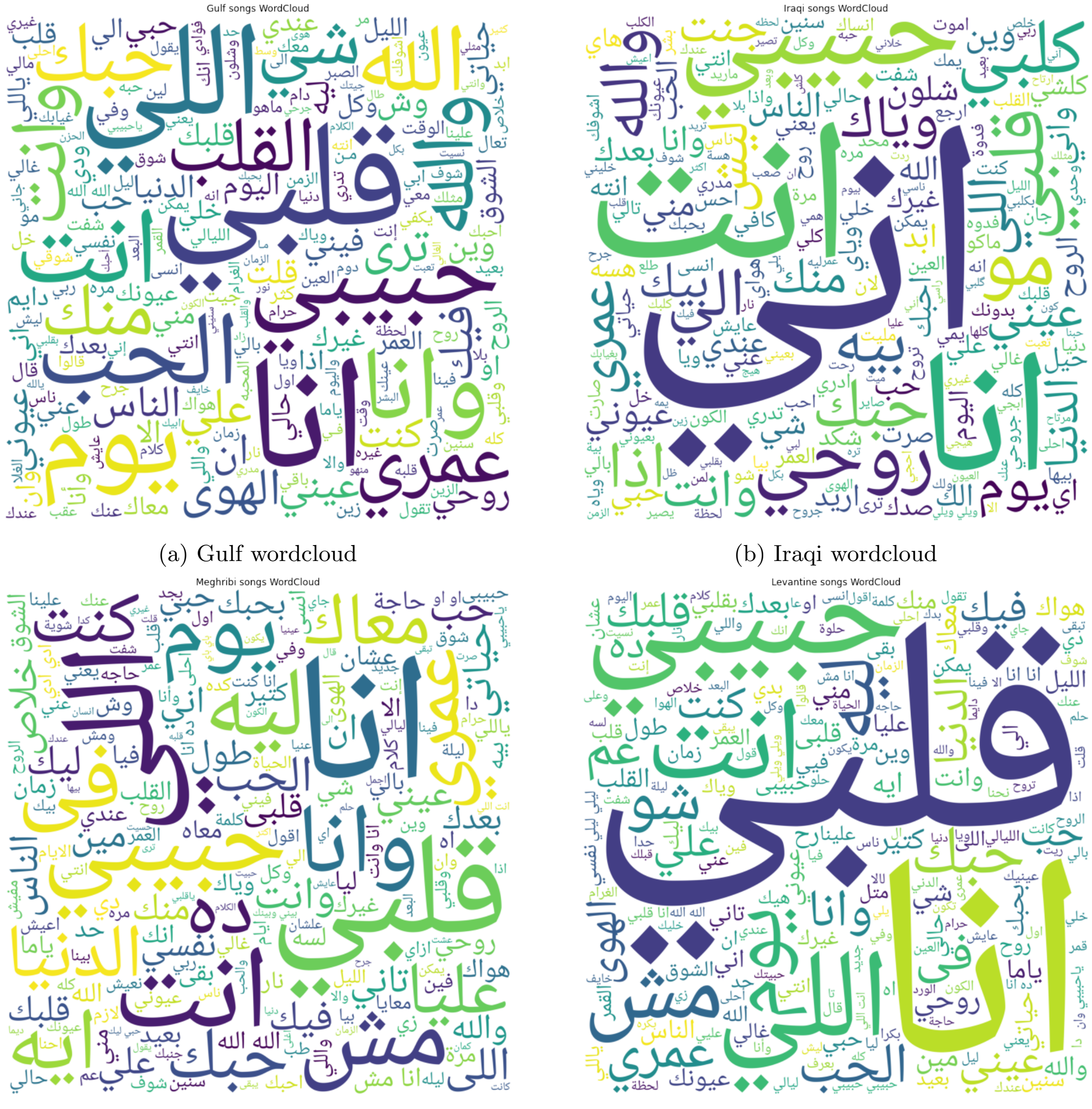
The following are word-clouds of each of the six dialects represented using NotoNaskhArabic-Regular font



(a) Egyptian wordcloud



(b) Sudan wordcloud



(a) Gulf wordcloud

(b) Iraqi wordcloud

(c) Meghribi wordcloud

(d) Levantine wordcloud

Figure 3: Word-Clouds of the 6 different dialects

3 Project Architecture

3.1 Data Collection and Preprocessing

To facilitate the training of the Arabic lyrics generator, we performed preprocessing steps on the dataset. These steps involved cleaning , dropping the columns:

- 'SongTitle' for not being relevant to the text generation process.
- SongWriter and Composer since both columns almost had 50% null values.
- SingerNationality dropped since we care more about the dialect of text not the nationality of the singer.

3.2 Word Embedding Preprocessing

In the word embedding preprocessing step, the preprocessed Arabic lyrics undergo a transformation into numerical representations that are suitable for embedding. This involves assigning a unique index to each word in the vocabulary. By associating each word with a specific index, the lyrics can be effectively represented as sequences of numbers, facilitating further processing. Additionally, to capture the semantic nuances and contextual relationships between words, each word is encoded as a dense vector. This encoding can be accomplished through either utilizing a pre-trained word embedding model specifically designed for Arabic language or training a new embedding model on the Arabic lyrics dataset. By encoding the words as dense vectors, the lyrics generator can better understand and leverage the semantic associations between words, thereby enhancing the coherence and quality of the generated Arabic lyrics.

3.3 Model Selection and Fine-tuning

For model selection, the BART[2] model is a suitable choice for our Arabic lyrics generator. BART is a pre-trained sequence-to-sequence model that has demonstrated promising performance in various text generation tasks. While it's ideal to have an Arabic-specific BART model, BART can also be considered to be used as a multilingual model which is trained on Arabic and other languages. Such models can still capture language patterns and generate coherent Arabic lyrics.

To fine-tune the BART model, we will start by initializing it with pre-trained weights. These weights contain knowledge learned from a large-scale dataset. We will then further train the model using our preprocessed Arabic lyrics dataset. Fine-tuning allows the model to adapt and specialize for the task of generating Arabic lyrics. During this process, the model's parameters are adjusted to optimize its performance and align it with the specific characteristics of Arabic lyrical language.

3.4 Lyrics Generation

When it comes to lyrics generation, we will provide a starting prompt or seed input and we may consider an option to choose the dialect of the generated lyrics, which will be passed through the fine-tuned BART model. The model will then generate lyrics by sampling from its output probability distribution, producing creative and original text.

3.5 Evaluation Metrics

In evaluating the performance of our Arabic lyrics generator, we can utilize several metrics that measure the quality, diversity, and coherence of the generated lyrics. Some of these metrics include: The Perplexity which measures the ability of the model to predict the next word in a sentence. Lower perplexity indicates better performance in predicting the next word, indicating that the model has learned the language patterns of the dataset more effectively.

References

[1] University of Lancaster. (n.d.). HABIBI dataset. Retrieved from http://ucrel-web.lancaster.ac.uk/habibi/

[2] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 7871-7880).