Arabic Lyrics Generation

Mahmoud Hesham Shehabeldin Solyman Mohamed Sonbol

May 26, 2023

TEAM 10

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.

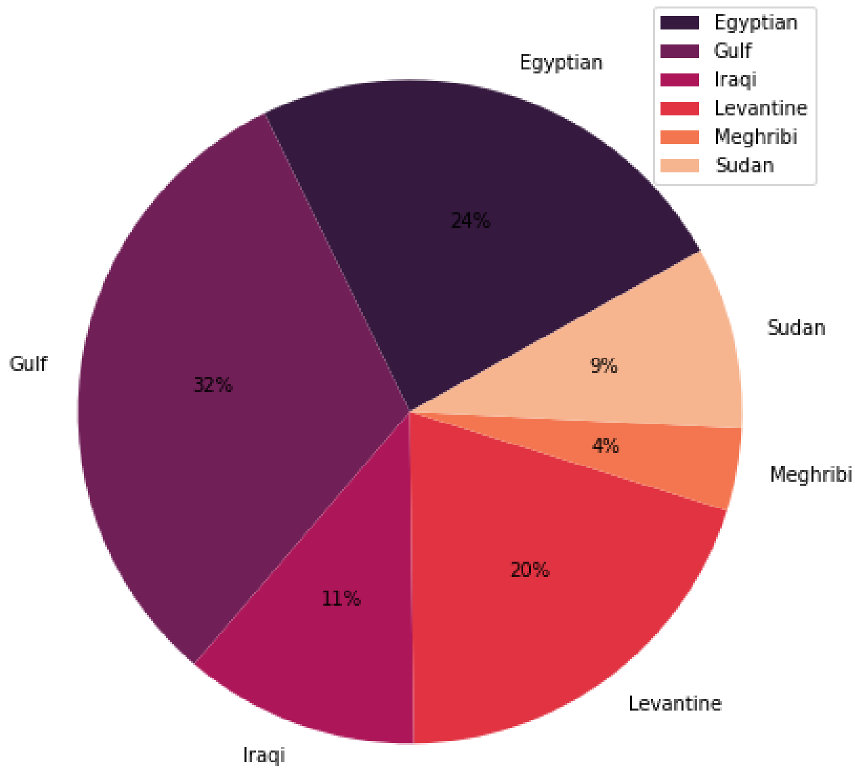

Figure 1

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 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

We focus on generating Arabic lyrics using the GPT-2[3] language model. GPT-2, short for "Generative Pre-trained Transformer 2," has emerged as a powerful tool in natural language processing (NLP) and text generation tasks. It is a transformer-based model that has demonstrated remarkable performance in various language generation tasks, including lyrics generation. We proceed to fine-tune the GPT-2 model on the Arabic lyrics dataset. Fine-tuning involves training the model on our specific dataset to adapt it to the task of generating Arabic lyrics. By exposing the model to a large volume of Arabic lyrics data, it learns the patterns, structures, and linguistic nuances of Arabic songs.

The advantage of using GPT-2 for lyrics generation lies in its ability to capture the intricacies of language and generate text that aligns with the style and characteristics of Arabic songs. It has the potential to produce culturally nuanced and linguistically diverse lyrics that resonate with Arabic speakers worldwide.

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: BLEU[4] (Bilingual Evaluation Understudy) score which is a widely used metric for evaluating the quality of machine-generated text, particularly in the field of machine translation. It measures the similarity between the generated text and one or more reference texts by computing the precision of n-grams (contiguous sequences of n words) in the generated text.

The BLEU score ranges from 0 to 1, with a higher score indicating a better match between the generated text and the reference texts. The score is calculated based on the precision of n-grams, where higher n-gram precision values indicate a better match.

When using BLEU score as an evaluation metric, it's essential to have one or more reference texts that serve as the ground truth for comparison. These reference texts should be carefully selected to represent the desired quality and characteristics of the generated text.

Overall, BLEU score provides a quantitative measure of the similarity between generated and reference texts, offering a valuable evaluation metric for assessing the quality of machine-generated text in tasks like machine translation, text summarization, and text generation.

4 Methodology

In this section, we will outline the methodology employed in this project, which includes the utilization of word embeddings and the fine-tuning of a pre-trained model, namely "GPT2."

GPT-2, or Generative Pre-trained Transformer 2, is a state-of-the-art language model developed by OpenAI. It has achieved remarkable success in various natural language processing tasks such as text generation, translation, summarization, and question answering.

It is a transformer-based model, meaning it employs a self-attention mechanism to capture relationships between different words in a sentence. This allows the model to understand the context and dependencies between words, resulting in more coherent and meaningful outputs.

One of the significant advantages of GPT-2 is its pre-training phase. Before fine-tuning, GPT-2 is trained on a large corpus of text from the internet, enabling it to learn grammar, syntax, and various linguistic patterns. This pre-training allows GPT-2 to have a strong foundation in language understanding.

The subsequent subsections will provide a more comprehensive and detailed examination of these approaches.

4.1 Pre-processing

Our chosen dataset[1] consisted of a collection of Arabic song lyrics obtained from various sources. To prepare the dataset for training, we performed several pre-processing steps. Firstly, we removed unnecessary metadata such as song titles, songwriters, composers, and singer nationalities, as they were irrelevant for the lyric generation task. Additionally, we cleaned the text by removing punctuation to ensure that the model focuses solely on the textual content of the lyrics.

4.2 Word Embedding

For this project, the tokenizer that accompanies the pre-trained GPT2 model was chosen due to its technical capabilities and compatibility with the GPT2 architecture. The GPT2 tokenizer is based on the Byte-Pair Encoding (BPE) algorithm, which allows for efficient and effective tokenization of text data.

The BPE algorithm used by the GPT2 tokenizer breaks down the input text into subword units, enabling the model to capture both word-level and subword-level information. This is particularly useful for languages with complex morphological structures, such as Arabic, as it helps the model handle variations in word forms and generate more accurate and contextually relevant lyrics.

The GPT2 tokenizer also incorporates a pre-defined vocabulary that was learned during the pre-training phase of the GPT2 model. This vocabulary consists of a set of subword units and special tokens, such as start-of-sentence and end-of-sentence markers, which are essential for proper sequence generation. By using this pre-defined vocabulary, the tokenizer ensures consistency and compatibility with the GPT2 model, allowing for seamless integration and efficient processing.

In addition, the GPT2 tokenizer provides useful functionalities for handling text data, such as encoding and decoding. The encoding process involves mapping each token in the input text to its corresponding numerical representation, enabling the GPT2 model to process the data as numerical sequences. Conversely, the decoding process involves converting the model's generated numerical output back into human-readable text.

By utilizing the GPT2 tokenizer, we leverage its technical capabilities to handle the intricacies of Arabic language tokenization. This tokenizer is specifically tailored to the GPT2 model, ensuring that the input data is properly prepared for subsequent fine-tuning and lyrics generation processes. It simplifies the tokenization process by taking into account language patterns, subword units, and special tokens, leading to improved accuracy and quality of the generated lyrics.

4.3 Fine-Tuning

The training and fine-tuning of the language model were crucial steps in the development of the lyric generation system. We leveraged a pre-trained GPT-2 model, a state-of-the-art language model, as our base model. To fine-tune the model for generating Arabic song lyrics, we utilized the HABIBI[1] dataset comprising a collection of Arabic song lyrics. Our dataset consists of 6 different so we trained 7 different models one for each dialect and one trained on the whole corpus.

Next, we implemented a custom SongLyrics class to convert the lyrics into a suitable format for training the GPT-2 model. The class utilized the GPT2Tokenizer to tokenize the lyrics into subword units, which are the input representation expected by the model. We limited the maximum sequence length to ensure efficient training and prevent memory limitations.

For the actual training process, we employed the train function. This function encapsulated the necessary steps to train the model on the prepared dataset. We utilized techniques such as batch processing, where the dataset was divided into smaller batches for efficient processing. Gradient accumulation was used to accumulate gradients over multiple batches before performing the optimization step, which helped stabilize the training process. We employed the AdamW optimizer, which combines the Adam optimizer with weight decay regularization, to update the model parameters and minimize the training loss. During the training phase, the training data is processed to ensure it fits the input length requirements of the GPT-2 model. Since GPT-2 has a maximum sequence length, any input sequences exceeding this length are dropped. This ensures that the training data aligns with the model's input constraints, enabling effective learning and processing of the text.

To ensure optimal training, we incorporated a learning rate schedule using the get_linear_schedule _with_warmup function. This schedule gradually increased the learning rate during the warm-up steps and then decreased it linearly over the remaining training steps. This approach helped prevent overfitting and improved generalization.

During training, we monitored the loss value, which represents the discrepancy between the predicted lyrics and the ground truth. By minimizing this loss, the model learned to generate lyrics that closely resemble the training data. We trained the model for a fixed number of epochs, iterating over the dataset multiple times to enhance the model's learning capabilities.

To evaluate the model's performance and checkpoint the training progress, we had the option to save the model's parameters at each epoch using the save_model_on_epoch flag. This allowed us to retain the best-performing model or resume training from a specific checkpoint if needed.

Once the training process was completed, we saved the trained model, which could be later loaded and utilized for generating Arabic song lyrics based on user prompts. The fine-tuned model captured the language patterns and lyrical characteristics specific to Arabic songs, enabling it to produce coherent and contextually relevant lyrics.

4.4 Output Generation

The lyric generation process involved utilizing the fine-tuned GPT-2 model to generate Arabic song lyrics based on user prompts.

To generate lyrics, we implemented the generate function. This function took several parameters, including the fine-tuned model, a tokenizer for text encoding, a prompt as input, and other settings such as the desired number of generated entries and the entry length. The function utilized the GPT-2 model to iteratively predict the next token in the generated lyrics, based on the previously generated tokens.

During the generation process, the model made use of probability distributions to balance between exploring new possibilities and adhering to the context of the input prompt. The top_p parameter controlled the diversity of the generated lyrics, determining the cumulative probability threshold for selecting the next token. A higher top_p value allowed for more diverse outputs, while a lower value encouraged the model to stick to high-probability tokens.

To ensure the generated lyrics were in line with the desired style and tone, a temperature parameter was used to control the randomness of the model's predictions. A higher temperature value introduced more randomness, leading to more creative and varied lyrics, while a lower value made the model more deterministic, resulting in more focused and predictable outputs.

The generate function iteratively generated lyrics until it reached the desired number of entries or the specified entry length. It incorporated a stopping criterion based on the presence of a specific token "<|endoftext|>", to determine when an entry was considered complete.

Once the lyrics were generated, we evaluated the outputs and compared them with the true ending lyrics from the test set. This comparison allowed us to assess the quality and accuracy of the generated lyrics.

By leveraging the fine-tuned GPT-2 model and the generation function, we were able to generate Arabic song lyrics that exhibited lyrical patterns, semantic coherence, and thematic relevance.

4.5 Output processing

In the process of lyric generation using the GPT-2 model, the output of the model is in the form of tokens, which are integer representations of words or subwords. To obtain readable text from these tokens, a decoding step is necessary. The tokens are converted to text using a tokenizer, which maps each token to its corresponding word or subword in the target language. By applying the decoding process to the generated token sequence, the output lyrics are transformed into human-readable text. This conversion enables the comprehension and evaluation of the generated lyrics in the desired language, in this case, Arabic.

5 Experiments & Results

In this section, we present the experimental setup and results of our fine-tuned models. We conducted experiments by training two separate models: one trained on a single dialect and another trained on the entire corpus.

5.1 Experimental Setup

To train our models, we utilized the fine-tuning approach on the GPT-2 architecture. We divided the dataset into subsets based on dialects and created a separate training set for each dialect. Additionally, we created a combined training set using the entire corpus to train a model on the overall language patterns and variations.

5.2 Results

We evaluated the performance of our models using 2 different metrics, including BLEU score and human evaluation for 10 songs of each dialect that we excluded from the training dataset.

Furthermore, it is important to acknowledge that the GPT2 model, without fine tuning, tends to generate English words as it is not specifically pre-trained on the Arabic language. Conversely, the model trained on the entire corpus demonstrates the capability to discern between dialects and generate lyrics consistent with the prompt's dialect.

Table 1: Bleu Scores for Different Dialect Models

Model	**Full Corpus**	**Gulf**	**Egyptian**	**Levantine**	**Iraqi**	**Sudan**	**Meghribi**
Avg Bleu Score	0.6810	0.6817	0.693	0.687	0.668	0.693	0.698

In Table 1, it is evident that across the 10 randomly selected test songs for each dialect, the average Bleu scores obtained by the different models exhibit a close similarity.

Singer	**Song Title**	**Prompt**	**True end lyrics**	**Single dialect model**	**Full corpus model**
مصطفى كامل	ام عيالي	يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه بسم الله انا قلبي اختارك تبقى حبيبتي وتبقى مراتي ام عيالي بسم الله انا جيت على دارك سبت القلب وجبت المهر حرير من الغالي بسم الله انا قلبي اختارك تبقى حبيتي وتبقى مراتي ام عيالي بسم الله انا جيت على دارك سبت القلب وجبت المهر حرير من الغالي حتت بت جننت اهلى حتت بت بس تستهلي حتت بت جننت اهلى حتت بت بس تستهلي انا	مقدرتش اجي على مهلي جتها جري جري جري جري جتها طوالي بسم الله انا قلبي اختارك تبقى حبيتي وتبقى مراتي	قلبي اختارك تبقى حبتي وتبقى مراتي ام عيالي بسم الله انا جي<|endoftext|>	قلبي اختارك تبقى حبتي وتبقى مراتي ام عيالي بسم الله انا جي<|endoftext|>
مصطفى كامل	ام عيالي	ام عيالي بسم الله انا جيت على دارك سبت القلب وجبت المهر حرير من الغالي يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه يوه مين قلب على قلبي ونداه ميت قلب لكن محتش الا ده خلاص من النهارده انا عمري ابتدى خلاص عمري ابتدى مين قلب على قلبي ونداه ميت قلب لكن محتش الا ده خلاص من النهارده انا عمري ابتدى خلاص عمري ابتدى حتت بت جننت اهلى حتت بت بس تستهلي	حتت بت جننت اهلى حتت بت بس تستهلي انا مقدرتش اجي على مهلي جتها جري جري جري جري جتها طوالي	قلب سلم عيالي بسم الله انا جيت على دارك سبت القلب وجبت المهر حر<|endoftext|>	بسم الله انا عيالي بسم الله انا عيالي بسم الله انا عيالي بسم الله انا �<|endoftext|>
عمرو دياب	ايام و بنعشها	ايام وبنعيشها أيام و بنعيشها هنعمل ايه يا قلبي في ناس ماهماش ناس مفيش احساس ومش بيحسوا بجروحنا دي دموعنا هنحوشها عشان مش صح نبقى ضعاف نبان ساعة الفراق بنخاف يا ريت نغصب على روحنا وعلى ايه دي حكاية خلصت من بدري والله خلصت من بدري والغدر حسيت بيه وعلى ايه على ايه هنبكى على الماضي أيام وعدت على الفاضي في ايه هنبكى عليه مش فارقة نتكلم نعيد ونزيد في ايه على ايه معدش يفيد بكانا عليه خلاص الجرح أهو معلم ويا ريتنا نتألم ولا نقاسي وداع بوداع محدش ساب حبيبو وضاع وأدينا خلاص بنتعلم وعلى ايه دي حكاية خلصت من بدري والله خلصت	من بدري والغدر حسيت بيه وعلى ايه على ايه هنبكى على الماضي أيام وعدت على الفاضي في ايه هنبكى عليه	من بدري والغدر حسيت بيه وعلى ايه على ايه هنبكى عليه مش فا<|endoftext|>	من بدري والغدر حسيت بيه وعلى ايه على ايه هنبكى عليه مش فا<|endoftext|>

Figure 4: Egyptian dialect results

In figure 4 the sample results for the Egyptian dialect. we can see that for The first and the second row both models (the one trained on the whole corpus and the one trained on the Egyptian songs only) performed similarly which is unexpected since one has more diverse data than the other.

For the third prompt we have almost accurate output for both models as they generated the same true end of the lyrics.

<table>
<tr><th>Singer</th><th>Song Title</th><th>Prompt</th><th>True end lyrics</th><th>Single dialect model</th><th>Full corpus model</th></tr>
<tr><td>بشير شنان</td><td>ياغاية المطلوب</td><td>ياغاية المطلوب ياصافي الاوجان رميتني وسط البحور الغزيره ترى الذي غرك على غير برهان قصده يولع وسط قلبي السعيره قصده يغير كل عزك بحقران وان طعنتي يازين لاتستشيره اشرح لي الموضوع وان كنت غلطان حقك علي ياسيد كل العشيره انت الذي في داخل القلب سكان وانت الذي ابدي عليك السريره انت الذي لك بسقل القلب بستان وانت الذي لاجلك دموعي نثيره وانت الذي تشبه قمر شهر شعبان لازل منه ايام ماهي كثيره وانت الذي جرعتني المر الوان وانت الذي روحي من اجلك خطيره انت الذي اسقيتني سم ثعبان وذبحتني بعيونك</td><td>المستديره ممشاك متغير وطبعك معي شان وكل الامور الياسره جت عسيره رجواي فالي لاعطى غير منان يعطي برايه مايشاور وزيره</td><td>براب بحرا proven حقلبه وأعوره وابل شهر شعبان لازل منه ايام ماهي ك <|endoftext|></td><td>وشودك بهران وانت رجيان انت الذي تشبه قمر شهر شعبان لازل منه<|endoftext|></td></tr>
<tr><td>عوض الدوخي</td><td>هو يا مال</td><td>ما غفر أنا خطاك حبك من ضميري إنزاح والسبب جرح بقلبي ما برى وإنزاح أنسيتيني ليش وانسيت اللعب وإمزاح قلبك علي قسى شعملت أنا من بد ما بين الأحباب تحجي علي بد والضرس لي من رقل من شلعته لابد أنظر إلى الغيم لي هب الشمال إنزاح دمعي تحدر علي وجناي وأستاهل واسراي همي أبد واستاهل هذا جزى من رابع</td><td>الإنذال يستاهل رابعت بدر ذهب واليوم فضة قلب ان كان هذا وذا ولا لي قلب أنا الذي استحق الصلب واستاهل</td><td>ة أنظر إنزاح أنا من خطاك حبك من ضميري إنزاح والسب جرح بق<|endoftext|></td><td>بعد كله انسيت بد ما بخد في السب شعملت لي هب الشمال إنزاح دمعي <|endoftext|></td></tr>
<tr><td>عبد الرحمن محمد</td><td>لو كان بيننا</td><td>يا ربي قد غفلنا وكلنا طمع في عفو وفي كرم وفي اعتاق وجنة مع سيد البشر ندعوك ندعوك ربي من الأعماق لو كان بيننا الحبيب لدنا القاصي والقريب من طيبة قبل المغيب طالبا قرب الحبيب بقربه النفس تطيب وتدعو الله فيجيب أنوار طه لا تغيب بلغنا لقاه يا مجيب هداك الكون الرحيب رحمة الهادي القريب حديثك النهر العذيب جوارك الغصن الرطيب فدتك روحي يا حبيب محمد مكرم الغريب بقربك</td><td>الروح تطيب يا رحمة للعالمين يا حبيبي يا محمد يا طبيبي يا ممجد أنت ذوالفضل المؤيد جل من صلى عليك</td><td>الرقيب عني عفو المغيب لغيبه بالعذيب أنوار طه لا تغيب بلغن<|endoftext|></td><td>النفس تطيب وتدعو الله فيجيب أنوار طه لا تغيب بلغنا لقاه يا<|endoftext|></td></tr>
</table>

Figure 5: Gulf dialect results

In figure 5 the sample results for the Gulf dialect. In the first row the model trained on one dialect performed poorly even generating an English word while the model generated on the whole corpus performed better which is expected. For the remaining sample data, the full corpus model exhibited a slight improvement, generating sentences that are more legible sentences compared to the other model.

Singer	Song Title	Prompt	True end lyrics	Single dialect model	Full corpus model
جمال فرفور	الفرقه	اريتني اسلم واطيب الفرقه قسمه ونصيب اريتني اسلم واطيب الفرقه قسمه ونصيب ياسنين الضياع الضياع ياليالي الوداعياليالي الوداعالبريدو في قلبي وبقى ريدو لي خداعاريتني اسلم واطيب الفرقه قسمه ونصيب يادموع الندم الندم كل البينا انهدمكل البينا انهدموالصلات البينا راحت لا امل ولا عشم اريتني اسلم واطيبياليالي الشباب ياسنين العذاب ياسنين العذاب البريدو من قلبي وبقى ريدو لي سراباريتني اسلم واطيب الفرقه قسمه ونصيبمهما جارت بي السنين السنين مهما طال بي الانينمهما طال بي الانين بكره تبرا ياقلبي وتاني ماتعيش حزيناريتني اسلم واطيبكل البينا ولا راح وراح وحدي حاضن الجراحوالغلطتو في حقي ويرضو طالب السماحالفرقه قسمه ونصيب ياريتني اسلم واطيب الفرقه قسمه ونصيبيادموع الندم كل البينا	انهدم والصلات البينا راحت لا امل ولا عشمياريتني اسلم واطيب الفرقه قسمه ونصيبيياريتني اسلم واطيب الفرقه قسمه ونصيبياريتني احلم واطيب	والديث بالعمر واعيب حني متوم ماعيب شبيب عماريتني اسلم واطيب الفر� endoft|> <|ext	قل انهدمكل البينا انهدموالصلات البينا راحت لا امل ولا عشم اريتني <|endoftext|>
زيدان ابراهيم	الزاهي لونك	اجمل حبيب لي الروح قلبي اصطفاك وما قادر اعرف ايه اسباب جفاك يا الزاهي لونك حبيت عيونك وعشقت طبعك وجمال فنونك ورد الخميله ناضره وجميله كالفل عبيرها وما لي بديلها انا ليك وافي وانت المجافي بعيش ليالي في خيالي ديمه ونعيش حياتنا هانيه ونعيمه دائما اشوفك بين الازاهار تتهادي دائما وانا طرفي ساهر فيك يا ملاكي قبلان هلاكي وايه اسباب جفاك وحياتك عيونك الباسمه ليا ونضار	خدودك الكم نديه وحيات شبابك ما لي وصيه غير بس حنانك ورضاك عليا سيب الجفا وارحم كفا ايبييه اسباب جفاك	ه وجمال فنونك ورد الخميله ناضره وجميله كالفل عبيرها وما لي بديل<|endoftext|>	ه وجميله كالفل عبيرها وما لي بديلها انا ليك وافي وانت المجافي<|endoftext|>
مصطفى حمزة	الجريف واللوبيا	عندي شوق لي نيلنا و الاهل و الطيبة الجريف و اللوبيا و باقي ذكرى حبيبة و قشة الصفصاف القايمة عاملة زريبة ضل مغطي القيف فوقو روعة عجيبة يا سواقي بلدنا وينو صوتك مالو وينو قارع مويتك غنى باسم حالو وين غسيل النيل و شرهدوم فى رمالو وينو ماضي الحلة وين ومين الشالو كان زمان نتدلى لما كنا صغار من فريع لفريع نقطع الاثمار كان بحرنا نعومو ناس كبار وصغار فينا طيبة القرية ومافى كان اسرار وينو صوت الساقية الكان يشق الليل وين نخيلنا الرامي الظلال فى النيل وين ووين دفيقو الكنا منو	نشيل وين جداول الموية الجارية زي السيل يابحر حلتنا فى رمالك ذكرى لا بتشيلا الموجة ولا بتفارق الفكرة الجريف واللوبيا	صوتك مالو وينو قارع مويتك فين ومين الشالو كان زمان نتدلى لما كن<|endoftext|>	عندي شوق لي نيلنا و الاهل و الطيبة الجريف و اللوبيا و باقي ذكرى �<|endoftext|>

Figure 6: Sudanese dialect results

In figure 6, the results of both models are different and not similar to True end lyrics however both generated lyrics are lexically correct and they rhyme to some extent.

<table>
<tr><th>Singer</th><th>Song Title</th><th>Prompt</th><th>True end lyrics</th><th>Single dialect model</th><th>Full corpus model</th></tr>
<tr><td>اسماء لمنور</td><td>عندو الزين</td><td>عندو الزين عنده الحمام دايرو في دارو وانا الحب وانا الغرام جابني تا لدارو كيقولوا عينيه زايغة عمره في حياته ما أبغى لا قال لشي واحدة زوينة تتكون صباغة وانا ولا جاني على لقانا إمتى غيجيني وانا والله</td><td>جاني على لقانا صح يبغيني كيقولوا حافظ سواري وماشي اللي جات تديه مقرية مو مزيان صعيب وحده أن توصل ليه</td><td>صحرة ولا تحبة قال له بي سعض ويه وانا تا عينيه وانا غرام جابني<|endoftext|></td><td>عاجب وملاغة بعزوم قال شباته ما قال تمعر في سيري اينا عينيه رض<|endoftext|></td></tr>
<tr><td>اماني السويسي</td><td>هي جزاتي</td><td>هي جزاتي تنسى حبي وزكرياتي هي جزاتي الحزن يمالي حياتي هي جزاتي بعد عشر العمر الي كانت بعد روحي فيك ماعانت تبدالي بكزب حياتي لوكلام الدنيا ينفع كنت اعاتب كنت الوم لا عتزارك الي يشفع لا دموع إلها لزوم لو شبابي الى راح يرجع عن هواك سهران لاصوم عن الغرام</td><td>إلى كان اسافه لما بعته بكم مكان راحت وبعت المسافه بعته وضيعت المكان اسوتك بعد لاطفه مرات مع ظلمك كمان</td><td>لو كنت الوم لا عتزارك الي يشفع لا دموع إلها لزوم لو شبابي ال<|endoftext|></td><td>عنها ينتى حبي بعد عشر العمر الي كانت بعد روحي فيك ماعانت تبد<|endoftext|></td></tr>
<tr><td>الشاب مامي</td><td>تيجي تيجي</td><td>تيجي تيجي ريحني وقولي انت عاشق هوايا دي عيونك فيها كلمة ولسه ما بتقول اه يالي رماني زماني معاك بحكاية وعمرك ما تعطي حبيبي وصبري يطول وانا اشكي لمين حبيبي غيرك انا غير عيونك ما لقيت ريحني وقولي ياريت والي تمنيتو لقيتو في قربك حبيبي تيجي تيجي ننسى الشوق تيجي تيجي بينا نروق تيجي تيجي دي الايام معدودة ويشهد عليك قلبي وقلبك ده لو يوم تغيب حبيبي ولا قبلك ولا بعدك مه انت الحبيب بعد غريب حبك ده في الدنيا تيجي تيجي ننسى الشوق تيجي تيجي بينا نروق تيجي تيجي دي</td><td>الايام معدودة تصدق ما تعرفشي تخبي ويبان عليك حبيبي ومعاك ديما قلبي وبالروح شاريك واشتق اليك وقرب ده في الدنيا</td><td>الايام معدودة ويشهد عليك قلبي وقلبك ده لو يوم تغيب حبيبي ولا<|endoftext|></td><td>الايام معدودة ويشهد عليك قلبي وقلبك ده في الدنيا ت autosalam اعده علي�<|endoftext|></td></tr>
</table>

Figure 7: Meghribi dialect results

In Figure 7, the first two rows showcase similar results from both models, although the output from the full corpus model is more legible, as anticipated. However, the results in the last row are unexpected. The model trained solely on the Meghribi dialect outperformed the model trained on the entire corpus significantly, while the latter produced notably poor results, even generating English words.

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).

[3] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI Blog.

[4] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 311-318).