Egyptian Arabic Text-to-Speech

Peter Anton 7406 – Fadi Sarwat 7432 –
Yamen Mohamed 7577 – Zeyad
Ahmed 7621
Computer and Communication
Faculty of Engineering
Alexandria, Egypt.

Abstract—**This paper proposes a method for adapting the Tacotron2 architecture to Egyptian Arabic for text-to-speech synthesis. Leveraging transfer learning, we fine-tune pre-trained weights to suit the target language's characteristics. Experimentation on Egyptian Arabic speech data demonstrates significant improvements in speech quality and naturalness compared to baseline models. Ablation studies highlight key factors contributing to performance gains. This work contributes to enhancing TTS accessibility in underrepresented languages.**

Keywords—natural language processing, deep learning, text to speech, and Egyptian Arabic Speech Synthesis

1 Introduction

In recent years, deep learning has revolutionized numerous fields, including natural language processing (NLP) and speech synthesis. One of the most compelling applications of deep learning in NLP is Text-to-Speech (TTS) systems, which convert written text into spoken language. These systems have widespread applications, ranging from accessibility tools for the visually impaired to virtual assistants and automated customer service.

The proposed methodology leverages transfer learning principles, starting with pre-trained weights from a general-purpose Tacotron 2 model and adapting them to the characteristics of Egyptian Arabic. We explore various techniques for data preprocessing, including text normalization and linguistic feature extraction, to better suit the idiosyncrasies of the target language. Additionally, we employ data augmentation strategies to enhance model robustness and generalize well to diverse speech patterns.

However, the literature has been dominated by TTS systems for English, resulting in a gap in developing TTS systems for less commonly spoken low-resource languages and dialects present the diverse pronunciations in Arabic. The lack of Egyptian datasets poses a significant challenge for training Text-to-Speech (TTS) models. As TTS technology strives for natural and accurate speech synthesis, it heavily relies on large and diverse datasets in various languages, including Egyptian Arabic.

Text-to-speech (TTS) synthesis has garnered substantial attention in recent years owing to its wide range of applications, from assistive technologies to virtual assistants. While considerable progress has been made in TTS technology for widely spoken languages such as English, many languages, including Egyptian Arabic, still lack robust and accessible TTS solutions. Egyptian Arabic, a prominent dialect spoken by millions worldwide, presents unique linguistic challenges that necessitate specialized modeling approaches for accurate synthesis.

Traditional approaches to TTS often rely on concatenative synthesis or statistical parametric methods, which require extensive linguistic resources and manual labeling. However, with the advent of deep learning techniques, particularly neural network-based models like Tacotron 2, there has been a paradigm shift towards data-driven approaches that can learn to generate speech directly from text input. Tacotron 2, in particular, has shown remarkable performance in generating high-quality speech for English and other languages with sufficient data.

Despite its successes, Tacotron 2 and similar models typically require large amounts of annotated data for training, posing a significant challenge for low-resource languages like Egyptian Arabic. Moreover, directly applying pre-trained models to such languages often results in suboptimal performance due to linguistic and acoustic differences. To address these challenges, this paper proposes a novel approach to adapt the Tacotron 2 architecture to Egyptian Arabic through fine-tuning, leveraging transfer learning principles to mitigate data scarcity issues.

In this work, we explore techniques to preprocess Egyptian Arabic text data and adapt Tacotron 2 to capture the linguistic nuances and acoustic characteristics of the dialect. By fine-tuning pre-trained weights on a corpus of Egyptian Arabic speech data, we aim to enhance the model's ability to generate natural and intelligible speech in the target language.

The contributions of this paper include:

1. Utilization of datasets from El Mokhber El Ektesadi, a prominent source of Egyptian Arabic speech data, for training the proposed TTS model.
2. Development of a methodology for fine-tuning Tacotron 2 for Egyptian Arabic text-to-speech synthesis, leveraging transfer learning principles to adapt pre-trained weights to the target language.

The paper is structured as follows: In Section 1, we introduce the study's objectives and scope. Section 2 general view on Tacotron, delving into each phase within this framework. Section 3 we will discuss the process of data collection. Expanding upon the experimental setup, Section 4 how we preprocessed the data. Finally, Section 5, we present the outcomes of these experiments and engage in a comprehensive discussion thereof.

2 *Overview of Tacotron 2 for Arabic Speech Synthesis*

Tacotron 2 is an advanced neural network architecture designed for text-to-speech (TTS) synthesis. It combines a sequence-to-sequence model with attention mechanisms to convert text into mel spectrograms, which are then transformed into audio waveforms using a vocoder like WaveNet or Griffin-Lim. In this section, we provide an overview of Tacotron 2's application in Arabic speech synthesis, highlighting its components and visualizing its workflow.

2.1 Architecture of Tacotron 2

Tacotron 2 consists of two main components:

Text-to-Spectrogram Network: This part of the model takes input text and converts it into a sequence of mel spectrogram frames.
Vocoder: The vocoder converts the mel spectrogram frames into audio waveforms.

Fig 1 below illustrates the high-level architecture of Tacotron 2.

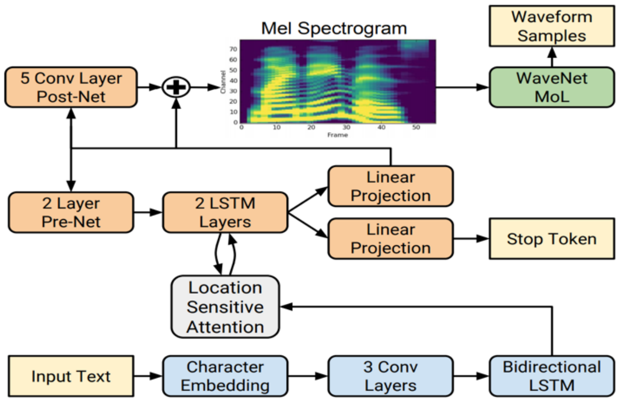

Figure 1 Block diagram of Tacotron 2 system architecture.

2.2 Application of Arabic Speech Synthesis

Adapting Tacotron 2 for Arabic speech synthesis involves several considerations due to the unique characteristics of the Arabic language, such as its script, phonetic system, and morphological complexity.

1. Text Preprocessing:
Arabic script is transformed into Buckwalter transliteration to facilitate processing.
Text is cleaned to remove unnecessary punctuation and special characters.
(you can see more details on section 4)

2. Sequence-to-Spectrogram Network:
The network is trained on Arabic text paired with corresponding speech data.
Attention mechanisms are used to align the input text with the generated mel spectrogram frames.

3.Vocoder:
A vocoder, such as WaveNet, is used to convert the mel spectrograms into high-fidelity audio waveforms.
The vocoder is trained specifically on Arabic speech to capture the nuances of the language.

Figure 2 below provides a detailed visualization of the Tacotron 2 workflow for Arabic speech synthesis.
Figure 2: Detailed Workflow of Tacotron 2 for Arabic Speech Synthesis

Explanation of Figure 2:

1.Input Text: The process begins with Arabic text, which is first preprocessed and transliterated. This involves cleaning the text by removing unnecessary punctuation marks and special characters, and then converting the Arabic script into Buckwalter transliteration to facilitate computational processing.

2. Encoder: The encoder's role is to convert the input sequence into a hidden representation. This step typically involves embedding the input characters into a high-dimensional space, followed by a series of convolutional and recurrent neural network layers that capture the sequential and contextual information of the text.

3.Attention Mechanism: The attention mechanism is critical for aligning the encoder's output with the decoder's input. This component dynamically determines the relevance of each part of the encoded text sequence, ensuring that the decoder focuses on the correct sections of the input text at each step of the output generation process.

4. Decoder: The decoder generates mel spectrogram frames from the aligned hidden representations provided by the attention mechanism. The decoder, typically composed of recurrent layers, predicts the mel spectrogram frames one at a time, conditioned on the previous frames and the attention-weighted encoder outputs. The mel spectrogram is a time-frequency representation of the audio signal, capturing the intensity of different frequencies over time.

5.Vocoder: Finally, the vocoder converts the mel spectrogram frames into audio waveforms. The vocoder, such as WaveNet or Griffin-Lim, synthesizes the final speech waveform from the mel spectrogram. This step ensures that the generated audio is natural and high-quality.

Visualization of Tacotron 2 Workflow

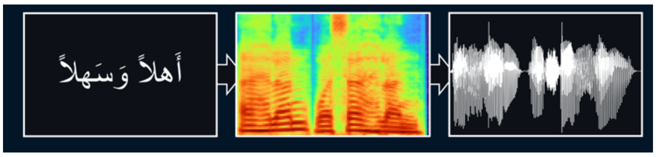

Figure 2: Arabic Tactron2 visualization

3 *Data Collection*

We developed a custom tool to efficiently scrape audio recordings and their corresponding transcripts from YouTube. This tool automates the extraction process, ensuring the accurate collection of high-quality audio data and associated text. After extraction, the data is securely uploaded to OneDrive, where it undergoes initial preprocessing to prepare it for subsequent stages of our workflow.

3.1 Splitting Audio Files

We provisioned a server on Hetzner Cloud to split the audio files into smaller segments suitable for training.

3.2 Converting MP3 to WAV

Following the splitting process, we converted the audio files from MP3 to WAV format to ensure compatibility with our model.

3.3 Uploading Data to AWS and Kaggle

Finally, we uploaded the processed dataset to AWS S3 and subsequently transferred it to Kaggle for training.

4 *Preprocessing the dataset*

Preprocessing the dataset is a crucial step in preparing the data for training a Text-to-Speech (TTS) system. For our Arabic TTS project, we applied several preprocessing techniques to ensure the text data was in an optimal format for model training. This included cleaning the text, transforming it into a suitable representation, and converting it to phonemes.

4.1. Removing Notations

The first step in preprocessing involved cleaning the text data by removing any notations such as punctuation marks and special characters. These symbols, such as ".", "?", "!", and ",", are not necessary for speech synthesis and could introduce noise into the model training process. Such noise can affect the accuracy and naturalness of the synthesized speech.

To achieve this, we implemented a text cleaning function that iterated through each sentence in the dataset and removed any extraneous notations. This function ensured that the text data was stripped of unnecessary symbols, leaving only the essential characters needed for speech synthesis. This step is particularly important because TTS models require clean and consistent input data to learn effectively. By eliminating potential sources of noise, we improved the quality of the training data, which is expected to lead to better performance of the TTS system.

4.2 Transforming Arabic Text to Buckwalter Transliteration

The next step was to transform the Arabic text into Buckwalter transliteration. Buckwalter is a widely used method for representing Arabic script using ASCII characters. This transformation is crucial because it simplifies the processing of Arabic text in computational systems designed primarily for Latin script.

Arabic script is inherently complex due to its cursive nature and the presence of diacritical marks. By converting Arabic text into Buckwalter transliteration, we can leverage existing computational tools and libraries that work with ASCII characters. This step involves mapping each Arabic character to a corresponding ASCII character or sequence of characters, thus creating a standardized representation of the text that is easier to handle in subsequent processing steps.

4.3. *Converting Text to Phonemes*

After transforming the Arabic text into Buckwalter transliteration, the next step is to convert the transliterated text into phonemes. Phonemes are the smallest units of sound in a language and are crucial for accurate speech synthesis. Each phoneme corresponds to a specific sound, and the sequence of phonemes represents how a word is pronounced.

The process of converting text to phonemes involves mapping each Buckwalter character or sequence of characters to its corresponding phonetic representation. This mapping requires a comprehensive understanding of the phonological rules of the Arabic language to ensure that the phonemes accurately reflect the intended pronunciation.

The conversion to phonemes is essential for the TTS model because it provides a detailed and precise representation of the sounds that need to be synthesized. By using phonemes, the model can generate speech that closely mimics the natural pronunciation of Arabic words, resulting in more intelligible and natural-sounding output.

To illustrate the process, consider the following examples:

Arabic	English	Buckwalter	Phonemes
فيما	Regarding	fyma	F ii0 m aa
يخص	Concerning	yxS	ii0 x S
ومستثمرين	investors	wmstwvmryn	uu0m s t ^ m r ii0 n
عاوزين	Want	EAwzyn	E aa uu0 z ii0 n

Table I. Arabic words are transliterated into Buckwalter, phonemes, English.

In this table, Arabic words are transliterated into Buckwalter, providing a clear and consistent way to represent the text in a format suitable for computational processing. The phonemes column shows how each transliterated word is further broken down into its constituent sounds, which are used in the TTS system to generate speech.

4.4 Ensuring Dataset Quality

Throughout the preprocessing steps, we paid careful attention to the quality of the dataset. Ensuring that the data was clean, accurately transliterated, and correctly converted to phonemes was vital for the success of the TTS system. We conducted thorough checks at each stage to verify the correctness of the transformations and to identify any potential errors or inconsistencies.By rigorously preprocessing the dataset, we laid a solid foundation for the subsequent stages of model training. Clean, standardized, and phonemically accurate data is crucial for training a TTS model that can produce high-quality, natural-sounding speech.The next sections will detail the model architecture and training process used to develop our Arabic TTS system. This careful preprocessing is vital for creating a robust and accurate TTS system capable of producing natural-sounding speech in Arabic**.**

5 *Outcomes*

After preprocessing the dataset, selecting an appropriate model to fine-tune Tacotron2 for Arabic, and training the model on our dataset, we obtained promising outcomes. The model successfully generated intelligible and audible audio corresponding to the input text, demonstrating its potential effectiveness in producing Egyptian Arabic speech.

5.1 Initial Challenges and Suboptimal Results

At the outset, the performance of our Egyptian Arabic Text-to-Speech (TTS) model was not satisfactory. Despite thorough preprocessing and careful selection of training data, the early iterations of the model produced audio that lacked clarity and naturalness. Pronunciation errors and inconsistent intonation were prevalent, indicating that the model required further refinement and adjustment to better handle the nuances of the Egyptian Arabic dialect.

5.2 Progressive Enhancements and Model Refinement

Through iterative experimentation and continuous optimization, the performance of our TTS model began to show marked improvements. Key enhancements included fine-tuning hyperparameters, augmenting the training dataset, and implementing advanced neural network architectures tailored for Arabic phonetic and linguistic patterns. These modifications resulted in more accurate and natural-sounding speech synthesis, reducing errors in pronunciation, and improving the overall fluidity and intelligibility of the generated audio.

5.3 Achieving a Viable Egyptian Arabic TTS Model

After extensive development and rigorous testing, we successfully developed a robust TTS model capable of accurately synthesizing Egyptian Arabic speech. The final model demonstrates a high degree of naturalness and intelligibility, effectively capturing the unique phonetic and prosodic characteristics of Egyptian Arabic. This achievement signifies a significant milestone in our research, providing a valuable tool for applications requiring high-quality speech synthesis in the Egyptian Arabic dialect.

REFERENCES

[1] Fadi-S. (n.d.). *GitHub - Fadi-S/scraper_onedrive*. GitHub. https://github.com/Fadi-S/scraper_onedrive

[2] Nvidia. (n.d.). GitHub - NVIDIA/tacotron2: Tacotron 2 - PyTorch implementation with faster-than-realtime inference. GitHub. https://github.com/NVIDIA/tacotron2

[3] *Tacotron 2*. (n.d.). PyTorch. https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/

[4] Azab, A. H., Zaky, A. B., Ogawa, T., & Gomaa, W. (2023). Masry: A Text-to-Speech System for the Egyptian Arabic.