FOOCTTS: Generating Arabic Speech with Acoustic Environment for Football Commentator

Massa Baali[1], *Ahmed Ali*[2]

[1]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[2]Qatar Computing Research Institute, HBKU, Doha, Qatar

massa.baali@mbzuai.ac.ae, amali@hbku.edu.qa

Abstract

This paper presents FOOCTTS, an automatic pipeline for a football commentator that generates speech with background crowd noise. The application gets the text from the user, applies text pre-processing such as vowelization, followed by the commentator's speech synthesizer. Our pipeline included Arabic automatic speech recognition for data labeling, CTC segmentation, transcription vowelization to match speech, and fine-tuning the TTS. Our system is capable of generating speech with its acoustic environment within limited 15 minutes of football commentator recording. Our prototype is generalizable and can be easily applied to different domains and languages.

Index Terms: speech recognition, text-to-speech

1. Introduction

Generating speech with its corresponding acoustic environment remains a big challenge. Traditional text to speech (TTS) systems require high-quality, large-scale spoken-written pair training examples. However, the quality of TTS is still not sufficient for the generation of speech with its corresponding acoustic environment, especially for under-resourced languages or dialects, unlimited domains, or different speaking styles. In this paper, we present the FOOCTTS web application, which allows the user to input text and generate speech with its corresponding acoustic environment, which is a football crowd noise in our case. We propose a strategy of pre-training with source domain data followed by fine-tuning with target domain data. Our approach is fully unsupervised, where we collected videos from Youtube for a football commentator in the Tunisian dialect. Then, we labeled the data using a large pre-trained Arabic Automatic Speech Recognition (ASR). Our pipeline involved different preprocessing strategies such as CTC segmentation, voice activity detection (VAD), and transcription vowelization to match the speech. In speech synthesis, we use transfer learning broadcast news domain and finetune it for the football commentating domain. Finally, we evaluate our model on a new text for one of the previous games and reproduced the commentating in a different style which was hard to tell that this was generated by a model. Our main contributions fall in generating the acoustic environment with high-quality speech and with the data labeling using Arabic ASR. We use the following ESPNet-TTS recipe[1].

2. System Architecture

The FOOCTTS system is composed of three independent components, namely: the web application, the TTS inference, and the Farasa [1] toolkit application server. The complete system workflow diagram is shown in figure 1. It performs the following steps:

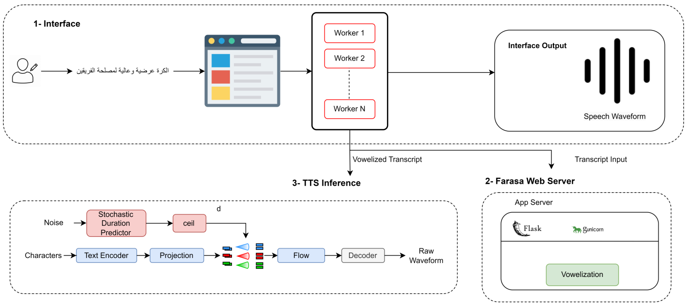

Figure 1: *System Overview*

- Capture input from user's keyboard through the interface.
- Spawn a worker that sends the text to the Farasa web server.
- Send vowelized text to TTS inference where we show in the figure the inference mode for our pre-trained variational inference with adversarial learning for end-to-end text-to-speech (VITS) [2] model.
- Play the final audio output on FOOCTTS.

2.1. Web Application

The web application is comprised of a front-end and a back-end. The front-end presents the users with an interface to enter the text through a textbox. The front-end is primarily built with CSS and Javascript. For the backend, we used Flask and Nginx. A worker passes the text input to the Farasa server through its web API and receives back the vowelized transcript as an output. Once the transcribed output is received, the data is sent to the TTS inference through Flask API.

3. Implementation

Figure 2 illustrates the data collection, preprocessing strategies, and the overall architecture of the TTS model.

3.1. Building TTS Corpus from Football Commentator

We collected videos from Youtube for football games for a Tunisian commentator reflecting the main use cases such as (shooting the ball, passing the ball, scoring, etc..).

[1] https://github.com/espnet/espnet/tree/master/egs2/qasr_tts/tts1

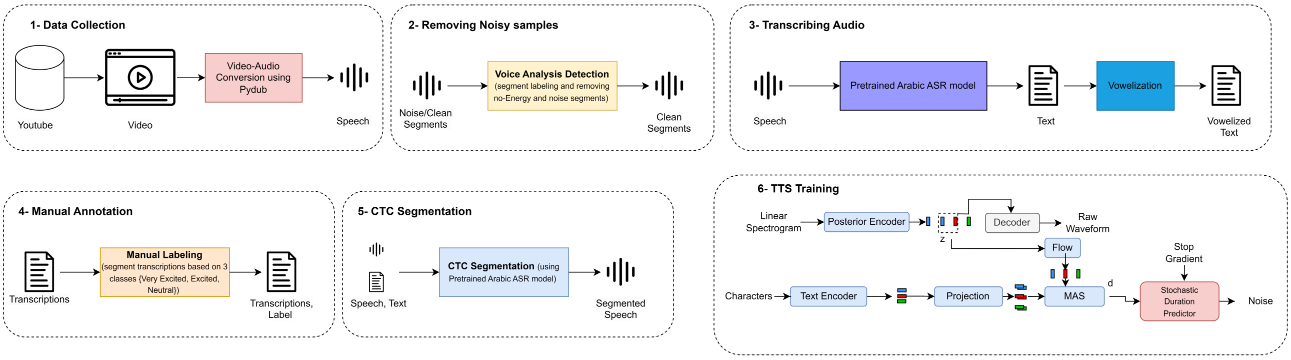

Figure 2: *Our proposed automatic pipeline*

3.1.1. Voice Activity Detection

We extract the audio from the video samples at 22Khz. Then, we process the audio files of each football video. We investigate the impact of removing noise labels using voice activity detection (VAD). We use the inaSpeechSegmenter pre-trained model [3] that extracts meta-data from each audio file; speech, noise, music and noEnergy (silence). We remove the noise, music, and noEnergy segments.

3.1.2. Transcribing Audio

We use a large pre-trained code-switching Arabic ASR model to transcribe the speech samples. Our ASR model was able to transcribe the French words that are spoken by the commentator and the player's names. We replace the French words with Arabic words by applying transliteration using QCRI's transliteration API. [2]

3.1.3. Vowelization using Farasa

In FOOCTTS, we employ the diacritizer provided by Farasa. One of the main issues is that vowelization is applied based on grammatical rules whereas if we want to perfect our output we should have the text vowelized according to how it was spoken.

3.1.4. Data Classification

We apply manual annotation for 15 minutes of dialectal Arabic dataset. We ask the annotators to label the data into three different classes: neutral, excited, and very excited. We instruct them to classify the samples based on the pitch.

3.1.5. CTC Segmentation

In the data classification part 3.1.4, we got segmented transcriptions with their corresponding emotion. We investigate the impact of segmentation by applying CTC segmentation using the large pre-trained Arabic ASR model to segment the big audio chunks into smaller chunks to match them with the segmented transcriptions. Our goal is to have better training samples by having one sample labeled with one emotion in order to help the model learn faster.

3.2. Building Speech Synthesizer for Commentator

For baseline model, we start with previous work in [4] and fine-tune it with 15 minutes of Tunisian commentator.

3.2.1. Model Architecture

We train VITS as our text to waveform (T2W) model which adopts a conditional variational autoencoder (CVAE) with normalizing flows and GAN-based optimizations. We train the model without the need for the two-stage pipelines which remain problematic because they require fine-tuning or training a neural vocoder. we show the architecture in this figure 2.

3.2.2. TTS Training

Our training process requires first training the text-to-waveform using 1 hour of one anchor male speaker from QASR corpus [5]. The text-2-wavform models were pre-trained as the character-based model with 1 hour from QASR dataset. We used 1150 utterances for the training and 25 utterances for the validation in pre-training. After pre-training, we fine-tune the model using commentator corpus. We used 25 utterances for development and 25 for testing and the rest for training.

4. Conclusions

This paper presents FOOCTTS, a speech synthesizer for a football commentator that generates speech with background crowd noise. Currently, the system works well without the user's ability to specify the emotion during inference time. For future work, we aim to allow the user to add the emotion with its corresponding text to generate speech in a certain tone.

5. References

[1] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for Arabic," in *The North American chapter of the association for computational linguistics: Demonstrations*, 2016.

[2] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*, 2021.

[3] D. Doukhan, J. Carrive, F. Vallet, A. Larcher, and S. Meignier, "An open-source speaker gender detection framework for monitoring gender equality," in *ICASSP*.

[4] M. Baali, T. Hayashi, H. Mubarak, S. Maiti, S. Watanabe, W. El-Hajj, and A. Ali, "Unsupervised data selection for tts: Using arabic broadcast news as a case study," *arXiv*, 2023.

[5] H. Mubarak, A. Hussein, S. Chowdhury, and A. Ali, "QASR: QCRI Aljazeera speech resource–a large scale annotated Arabic speech corpus," *ACL*, 2021.

[2] https://transliterate.qcri.org/