Arabic TTS using Tacotron 2

Students

Andrew Botros	20201322493
Amgad Mohammed	20201446888
George Ayman	20201446826
Katherine Ashraf	20201378779

Date

28-Dec-23

Course title

Natural Language Processing

Dependencies

The implementation relies on PyTorch, Google Colab, and various helper functions and classes provided in the repository.

Code Overview

Data Downloading:

The code starts by mounting Google Drive to the Colab environment and then proceeds to download and unzip the Tacotron 2 GitHub repository, as well as a pre-trained model and the ClArTTS dataset (that's been uploaded manually on one of our own Google drive accounts) for fine-tuning the model to our own purpose.

Preprocessing:

a. Reading and Formatting Text Data

The training and validation datasets are read from text files, split into filename, and text pairs, and stored in Pandas DataFrames.

```
# Display the data frame
train_data.head(10)
```

	filename	text
0	ch_00_arabic_tts_dataset_1	.... هَذَاالتَّسْجِيل .مِنْ طَرَفِ لِرِبَّرِي بُوكْس.
1	ch_00_arabic_tts_dataset_10	...خَاتَمِ الرُّسُلِ وَالأَنْبِيَاءِ .وَعَلَى آ.
2	ch_00_arabic_tts_dataset_11	...أَمَّا بَعْدُ .فَإِنَّ شَرَفَ الْمَطْلُوبِ .ب.
3	ch_00_arabic_tts_dataset_12	وَبِحَسَبِ مَنَافِعِهِ تَجِبُ الْعِنَايَةُ بِهِ.
4	ch_00_arabic_tts_dataset_13	...وَعَلَى قَدْرِ الْعِنَايَةِ يَكُونُ اجْتِنَاء.
5	ch_00_arabic_tts_dataset_14	...وَأَعْظَمُ الْأُمُورِ خَطَرًا وَقَدْرًا وَأَع.
6	ch_00_arabic_tts_dataset_15	...مَا اسْتَقَامَ بِهِ الدِّينُ وَالدُّنْيَا .و.
7	ch_00_arabic_tts_dataset_16	...لِأَنَّ بِاسْتِقَامَةِ الدِّينِ تَصِحُّ الْعِ.
8	ch_00_arabic_tts_dataset_17	.... وَبِصَلَاحِ الدُّنْيَا تَتِمُّ السَّعَادَةُ.
9	ch_00_arabic_tts_dataset_18	...وَتَفْصِيلَ مَا أُجْمِلَ مِنْ أَحْوَالِهِمَا.

The text is then pre-processed, converting Arabic text into phonemes using helper functions that converts to the Buckwalter transliteration system, then dropping the useless Arabic text column.

```
train_data.head()
```

	filename	Phonemes
0	ch_00_arabic_tts_dataset_1.wav	h a * aa aa tt a s j ii0 l + m i1 n + T A r a ...
1	ch_00_arabic_tts_dataset_10.wav	x AA t a m i0 + rr u0 s u0 l i0 + w a l < a n ...
2	ch_00_arabic_tts_dataset_11.wav	< a mm aa + b a E d u0 + f a < i0 nn a + $ a r...
3	ch_00_arabic_tts_dataset_12.wav	w a b i0 H a s a b i0 + m a n aa f i0 E i0 h i...
4	ch_00_arabic_tts_dataset_13.wav	w a E a l aa + q A d r i0 + l E i0 n aa y a t ...

b. Saving Processed Text Data

The processed data is saved into new text files for both training and validation sets, with filenames and corresponding phonemes.

c. formatting the data type.

The paths for the training and validation audio datasets are defined, and an ArabDataset class is used to pair text and audio data.

d. forming a mini batch of Tensor

A collate function designed to process and organize a batch of samples during data loading for text-to-speech training. It ensures that the sequences are properly padded and sorted, making them suitable for efficient batch processing during training.

Model Training:

a. Configuring Training Parameters

A configuration class is defined, specifying parameters such as batch size, number of epochs, learning rate, and others.

b. Tacotron 2 Model Initialization

The Tacotron 2 model is initialized with specific configurations and loaded onto the GPU if available.

c. Training Loop

The main training loop is implemented, including the optimization process, loss calculation, and logging. The training loop also includes validation steps to evaluate the model's performance on the validation set.

Inference:

The code includes an inference step where the trained Tacotron 2 model is used to generate speech from a given Arabic text. The resulting waveform and mel spectrogram are displayed and can be listened to using IPython's Audio class.

Conclusion

The provided code offers a comprehensive implementation of Tacotron 2 for Arabic TTS, covering data preprocessing, model training, and inference. Users can leverage this implementation for training their own Arabic TTS models or adapting it for other languages by making necessary adjustments.