Team members and contributions

Tao Yao: Comparisons between Tacotron 2 and Transformer TTS
Abhijit Bhatnagar: Comparison between Transform TTS and Fast Speech
Kam Chuen Mak: Comparison between FastSpeech and Fash Speech 2/2s

Selected option: Literature review [suggested length: 6-8 pages]

1. Introduction

In this report we will make a literature comparison between a few different model artectures that aim to achieve the same task, TTS (text to speech). The goal is to input a sequence of words and the trained model will generate a realistic human speech audio file speaking the same input sentence.

The reason why we think this is important is because this type of tasks are very on-demand in the current market throughout all products across major tech companies, and affect our daily task; from converting a reading book into an audio book, and expedites the book-lovers reading speed. In addition, to provide a neutral feel of audio while listening to audio books, the generated audios are indistinguishable from human recording.The technology also improves the accessibility to the people with disabilities, including people who are blind, using specialized TTS software called screen readers. Screen readers provide important functionality such as navigating through headings, speaking image alternatives, and identifying internal and external links. It is exciting to see how technology evolved to improve our daily lives as well as those people with disabilities.

It's exciting because up until now our NLP class has only covered text tagging related tasks but this is barely even the tip of the iceberg in the field of NLP as many research are involved with audio recognition/synthesis as well. As we dig into more details of each paper we realize it's even more exciting to see many model architectures introduced during class were used in some TTS models we're about to cover.

In summary, our project contribution will focus on comparisons of how those models taught in class were different in terms of improvement of accuracies and training time throughout a very short timeline.

2. Text to Speech Model Comparison

Tacotron2 and Transformer TTS

The first comparison is between Tacotron 2 (RNN/LSTM based) and Transformer TTS. Both models divide the whole NN into front-end NN, which takes a sequence of letters and generates mel spectrogram data (contains both time and frequency domain info). and a back-end NN synthesizer that turns mel spectrogram data into human speech audio data. In this report we only focus on the front-end as audio synthesis itself is an even larger topic and somewhat not too relevant to this text-focused NL class.

Both Tacotron 2 and Transformer TTS use modified WaveNet as the back-end to synthesize from generated mel spectrogram, a vocoder trained by a Google team. However Tacotron 2 uses RNN/LSTM based design for the front end model as mentioned earlier. Although the outcome itself is very impressive with almost indistinguishable human speech results. It suffers from a big problem which is very common among all RNN based models. First due to dependencies of the current node to its previous nodes, it's very difficult to train such a model with any form of parallelism. In the end it doesn't gain any benefit by utilizing GPU parallel computing like other models result in severe slow training time. The second problem is even with the latter LSTM design, the window duration from a long term sequence is still limited. Transformer TTS solved both problems by replacing the RNN with Transformer design, as its name suggests.

Most of the submodules and big pictures from both LSTM based Tacotraon 2 and Transformer have already been repeatedly covered in our NLP lecture. So here I will just highlight some part of the algorithm. Both models' front-end network are divided further into an encoder and decoder part. The Encoder converts the character sequence to internal feature representation, the decoder converts feature representation to frames of mel spectrogram.

In Tacotron 2 (fig 1.1). For the encoder part, the sequence of characters are embedded into a 512 long vector, followed by a 3 CNN layer then a Bi-directional LSTM which generates encoded features. Encoded features are then passed to attention units which convert features to fixed context vectors. For the decoder part. The prediction from previous time steps are passed to a pre-net consisting of 2 FC layers, both attention context vector and pre-net output are concatenated by uni-directional LSTM layers which will then be projected by a linear transformation to get predicted spectrogram frame. Then pass to a 5 layer CNN postnet to finalize the mel spectrogram before passing it to the wavenet vocoder.

In Transformer TTS (fig 1.2), the overall architecture is similar. Except the input word sequences are represented with an input and position embedding. Both encoder and decoder use a multi

head attention network followed by normalization and feed-forward network to replace LSTM and convolutional layers.

We chose both models because they are the best representation of two different designs in the field of TTS.and they are the perfect examples of how LSTM is gradually replaced by Transformer.

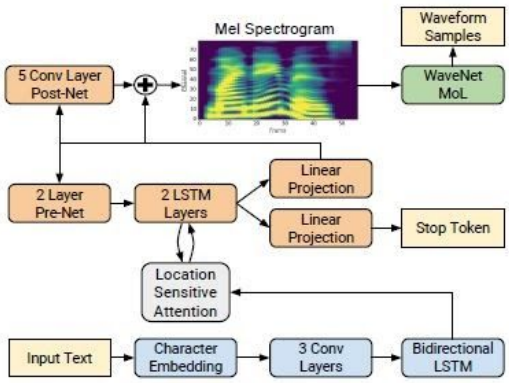


fig 1.1

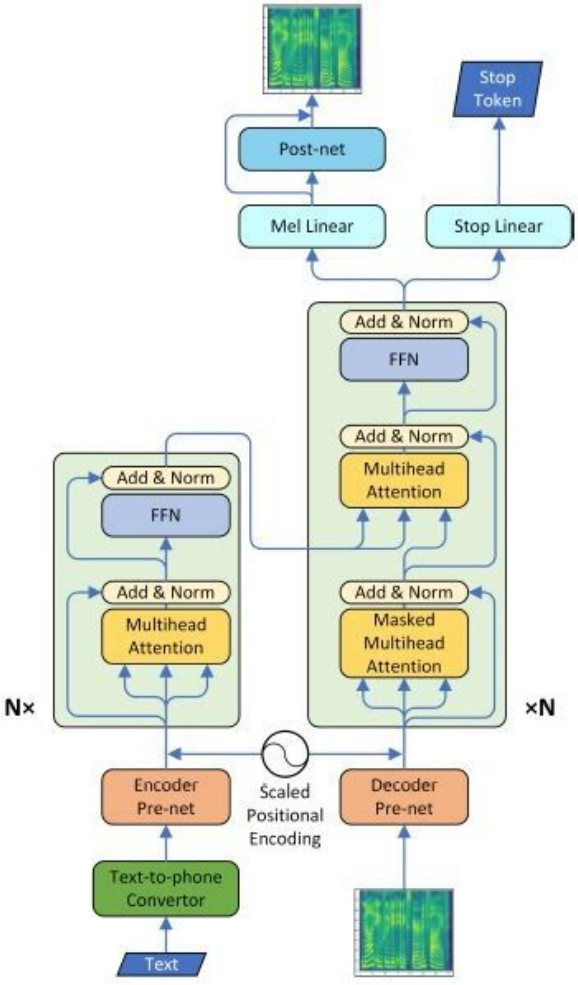
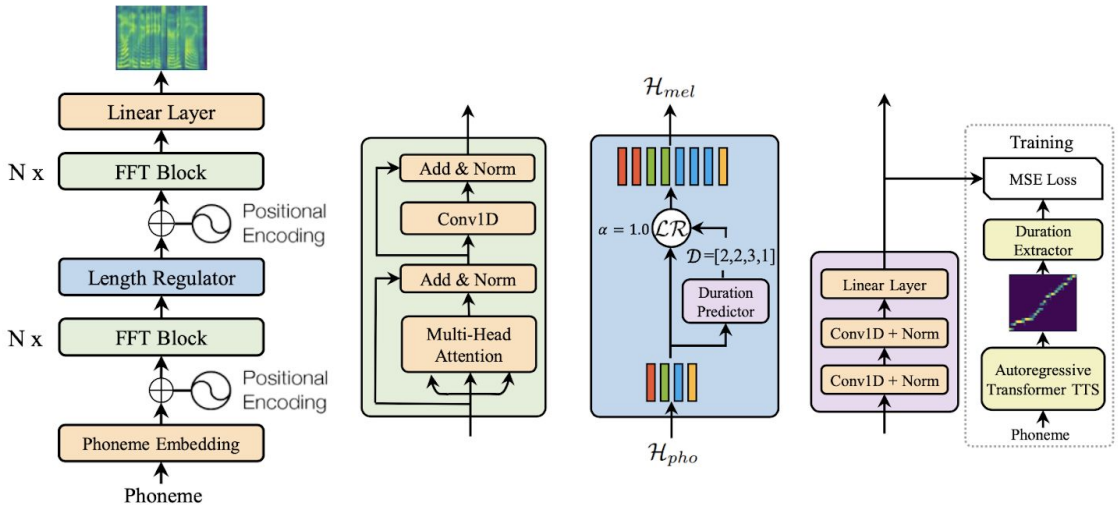


fig 1.2

Fast Speech and Transformer TTS

Fast Speech (fig 1.3) is a non-autoregressive model based on transformer architecture with multi-headed attention. It replaces the recurrent neural network component with a feed-forward network to parallelize training, making it a non-autoregressive model. This approach addresses low inference, less robustness and lack of controllability problems seen by traditional deep learning methods. Unlike Transformer, the generated mel-spectrograms depend purely on the

input phonemes, and not on previously generated mel-spectrograms. This model uses a length regulator and duration predictor blocks to solve the challenge of aligning a mel-spectrogram with phoneme duration accurately.

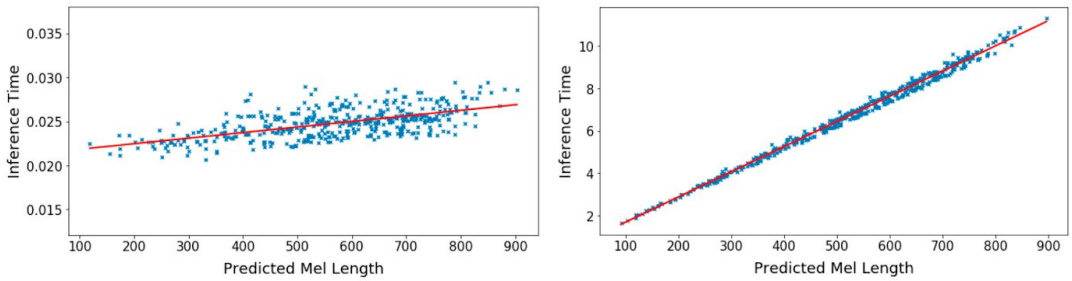


(a) Feed-Forward Transformer (b) FFT Block (c) Length Regulator (d) Duration Predictor

Fig 1.3

FastSpeech addresses the problems in Transformer TTS based models in the following manner:

1. It improves inference (Figure 2) and synthesis speeds through parallel mel-spectogram generation.
2. It improves robustness by avoiding issue of error propagation and wrong attention alignment, therefore reducing the ratio of skipped and repeated words.
3. It improves controllability by using a length regulator that can adjust voice speed by lengthening or shortening the phoneme duration to determine the length of the generated mel-spectogram.



(a) FastSpeech (b) Transformer TTS

Figure 2: Inference time (second) vs. mel-spectrogram length for FastSpeech and Transformer TTS.

The architecture of FastSpeech is a feed-forward structure based on self-attention in Transformer and 1D convolution (referred to as a feed-forward transformer). It stacks multiple feed-forward transformer blocks for phoneme to mel-spectogram transformation with N blocks on both the mel-spectogram and the phoneme sides. A length regulator bridges the gap between the phoneme and the mel-spectogram sequences.

The self-attention network in FastSpeech is different from that in Transformer TTS. FastSpeech uses a self-attention network with multi-head attention to extract cross-position information, whereas, Transformer uses a 2-layer 1D convolution network with ReLU activation. FastSpeech uses a different attention mechanism because hidden states are closely related in the character/phoneme and mel-spectrogram sequence in speech tasks. The length regulator solves the problem of length mismatch between phoneme and mel-spectrogram sequences by expanding the hidden states in the mel-spectrogram side feed-forward transformer. It assumes that the length of the phoneme sequence is much smaller than that of its mel-spectrogram, and that each phoneme corresponds to multiple mel-spectrograms. Based on the phoneme duration d, the length regulator expands the hidden states of the phoneme sequence d times such that the total length of the hidden states equals length of mel-spectrograms.

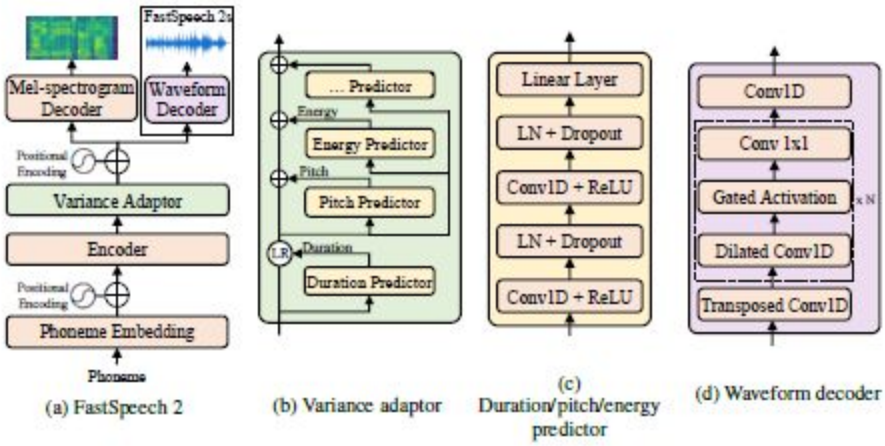
Fast Speech 2/2s and Fast Speech

This paper further improves the FastSpeech, a feed-forward network based on transformer model by addressing it's core issue in the model architecture. First, the distillation pipeline is too complicated and time consuming; Second, the data simplification in the teacher model causes the information loss and causes the model not accurate enough, and limits the voice quality.

FastSpeech 2 addresses the problems in FastSpeech based models in the following manner:

1. It removes the distillation pipeline and the phoneme duration from a teacher model
2. It uses the ground-truth training instead of simplified output from a teacher model to avoid information loss in distilled mel-spectrograms and increase voice quality.
3. It takes more different variations of inputs like pitch, energy to provide more variance information, and ease the one-to-many mapping problem in the TTS.

As shown in the figure below, FastSpeech2 first converts the inputs into phoneme hidden sequence, and then passes the output of encoder into the variance adapter and adds variance information like duration, pitch and energy, etc. Finally, the decoder will convert the hidden sequence into mel-spectrogram sequence in parallel.



FastSpeech 2S further improves the architecture without cascading the mel-spectrogram generation to improve the waveform generation. It discarded the mel-spectrogram decoder and simplified the model further. However, this approach has other disadvantages where the output waveform has more variance and the information gap is larger, it is also harder to train due to the missing intermediate mel-spectrogram layer.

3. Dataset

The dataset LJSpeech is used on Tacotron 2, Transformer TTS, FastSpeed, FastSpeed2 and FastSpeed2s. It consists of 13100 sentences written in text form and their exact matching recorded human speech audio files (ground truth), with the total audio length of about 24 hours.

4. Experiments & Results

In our experiment we mainly measured all models based on 1. training time of a single step with a batch size of 16 samples. 2. MOS (mean opinion score, human evaluations on how accurate the generated speech is comparing to ground truth). 3. CMOS. The following table shows overall result:

Model Name	Training Time (h)	MOS	CMOS
Tacotron 2	152.6	3.70 ± 0.08	0
Transformer TTS	38.64	3.72 ± 0.07	0.048
FastSpeech	53.12	3.68 ± 0.09	-0.885
FastSpeech 2	17.02	3.83 ± 0.08	0.000
FastSpeech 2s	92.18	3.71 ± 0.09	-
Ground Truth	-	4.30 ± 0.07	-

5. Conclusion

Throughout our exploration we get the conclusion that RNN/LSTM based model is gradually all replaced by the novel Transformer type model due to much higher parallelizable flexibilities with higher speed and broader past context window to produce slightly better accuracies and result for the goal of TTS. But both papers were published not too far apart from each other and the improvement was already so significant. It further proved that the field of AI is moving at a very fast pace. The future work will be to find the next successor which will replace the transformer and analysis for those even newer designs.

FastSpeech is a fast, robust and controllable neural TTS system, which uses key components - feedforward network, length regulator and a duration predictor, to alleviate problems faced by the Transformer TTS architecture. The generated audio nearly matches that of Transformer TTS in terms of speech quality, but the mel-spectrogram generation is sped up by 270x while end-to-end speech generation is sped up by 38x.

FastSpeech2 has addressed the problems in FastSpeech to improve both the pitch prediction and audio quality. While FastSpeech2s is the only model directly generating waveforms from phonemesquence fully in parallel, it leverage the architecture of FastSpeech and remove the mel-specgram model, and unlike other models discussed previously in the paper, which have separate models to convert input text to intermediate features, then convert these intermediate features into waveform.

6. References (not counted towards the page limit)

[Tacotron 2, Dec 2017] Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions; Authors: Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, Yonghui Wu
https://arxiv.org/abs/1712.05884v2

[Transformer TTS, Sep 2018] Neural Speech Synthesis with Transformer Network; Authors: Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, Ming Zhou
https://arxiv.org/abs/1809.0889

[FastSpeech, Nov 2019] FastSpeech: Fast, Robust and Controllable Text to Speech; Authors: Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhoy Zhao, Tie-Yan Liu
https://arxiv.org/abs/1905.09263

[FastSpeech 2, Dec 2020] Fast and High-Quality End-to-End Text to Speech; Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu
https://arxiv.org/abs/2006.04558