Text to Speech
Speech to Text

Елена Тупарова, 4MI3400132
Петър Петров, 2MI3400168

Text to Speech Use Cases

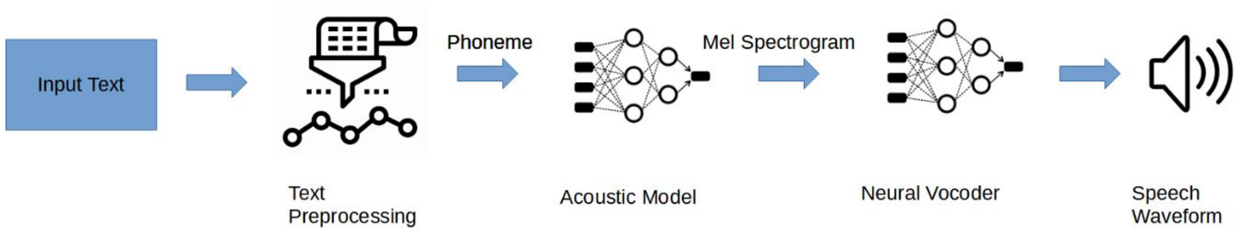
- Personal Virtual Assistants
- Creating audiobooks/podcasts
- Way to help users with speech disabilities communicate freely

Brief History Overview of TTS

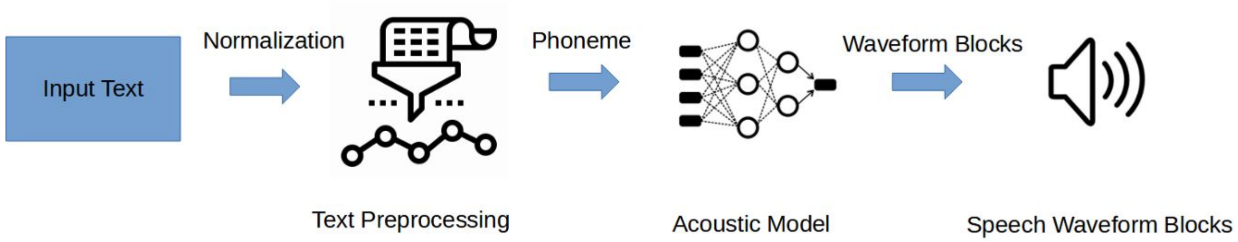
- 1961 - the song “Daisy Bell” is generated by an IBM 704 in Bell Labs
- 1968 - first general English text-to-speech system developed by Noriko Umeda et al. in Japan
- Main issue for decades - audio quality
- Deep Neural Networks

TTS Systems Architectures

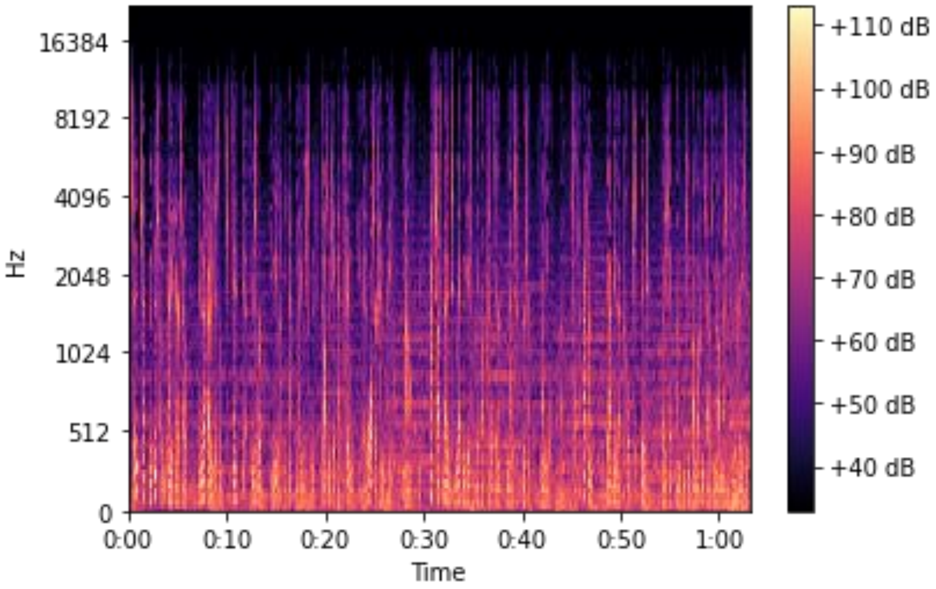
- Mainstream 2-stage



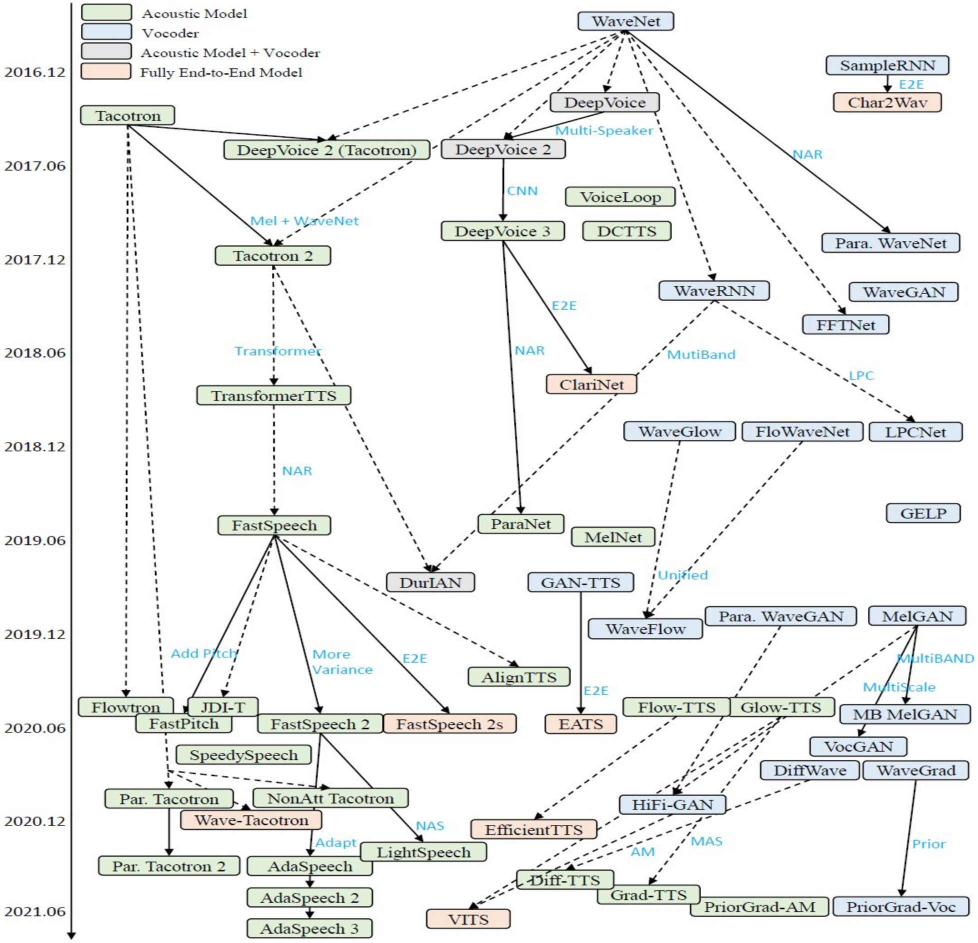
- End-to-End Text to Wave



What is a Mel-Spectrogram?



TTS Systems Evolution



Acoustic Models

- Recurrent Neural Network (RNN)
- Convolutional Neural Network (CNN)
- Transformers

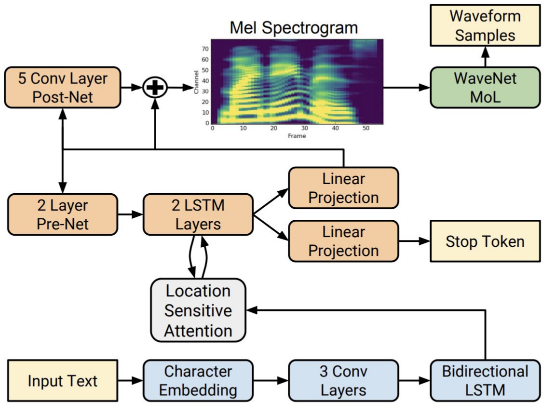
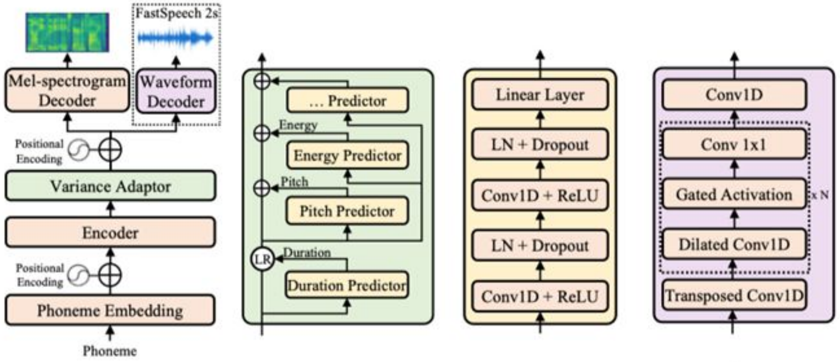


Fig. 1. Block diagram of the Tacotron 2 system architecture.



(a) FastSpeech 2 (b) Variance adaptor (c) Variance predictor (d) Waveform decoder

Neural Vocoders

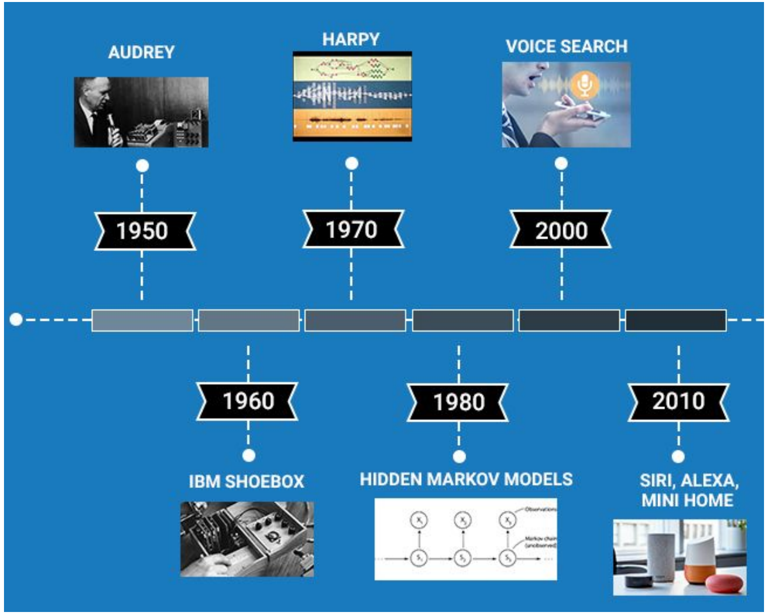
- Autoregressive
- Flow-based
- GAN-based
- Diffusion-based

TTS Frameworks

- TensorFlow TTS
- ESPnet

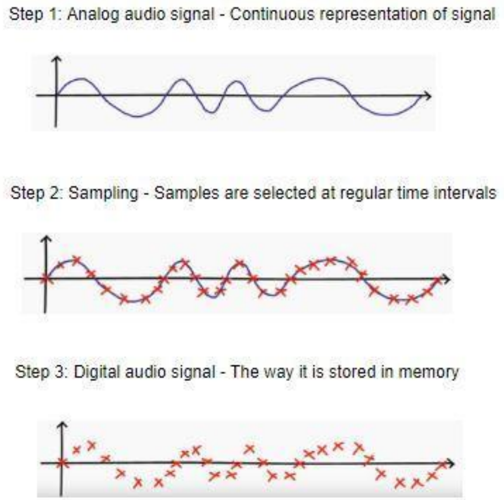
Brief History of STT

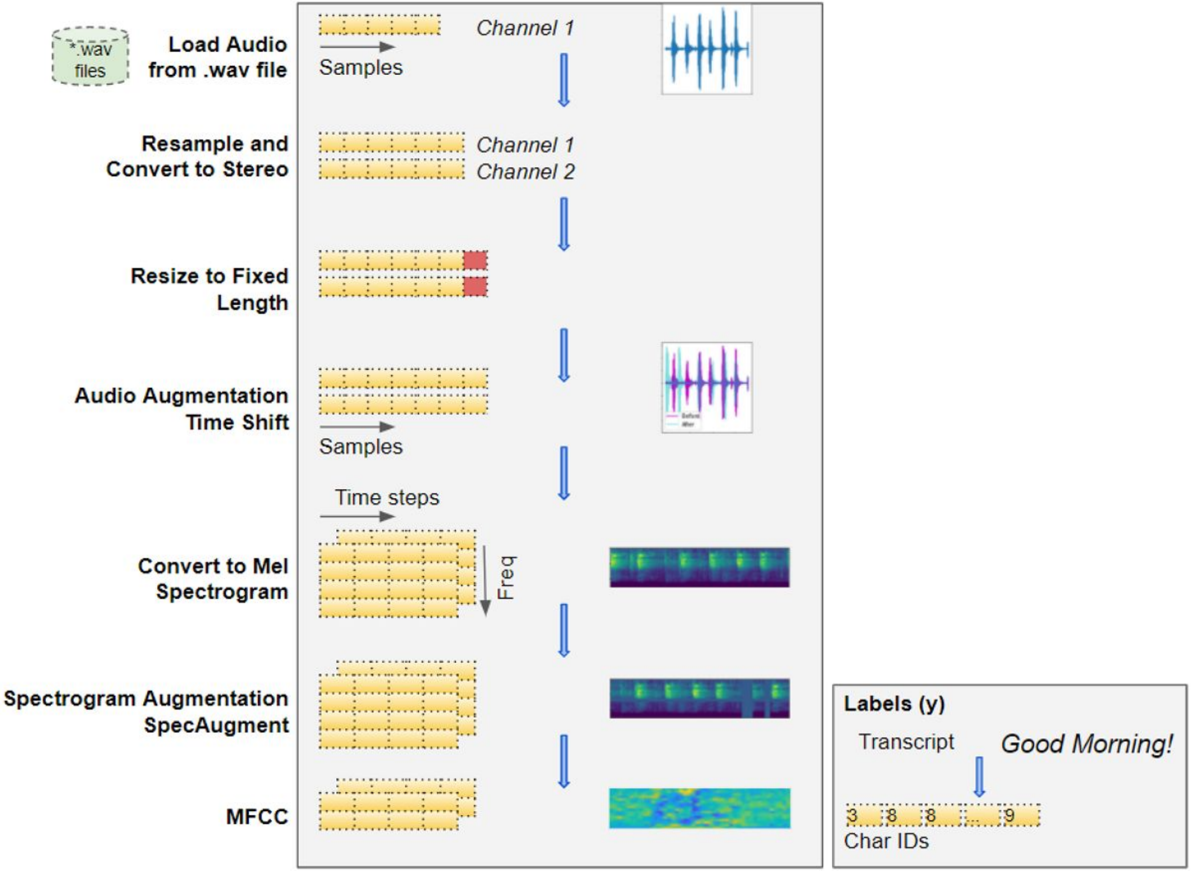
- Audrey was designed to recognize only digits.
- Just after 10 years, IBM introduced IBM Shoebox capable of recognizing 16 words including digits
- Harpy system was able to recognize 1011 words.
- The Hidden Markov Model In the 1980s
- In 2001 Google introduced the Voice Search
- In 2011 Apple launched Siri



Key Features

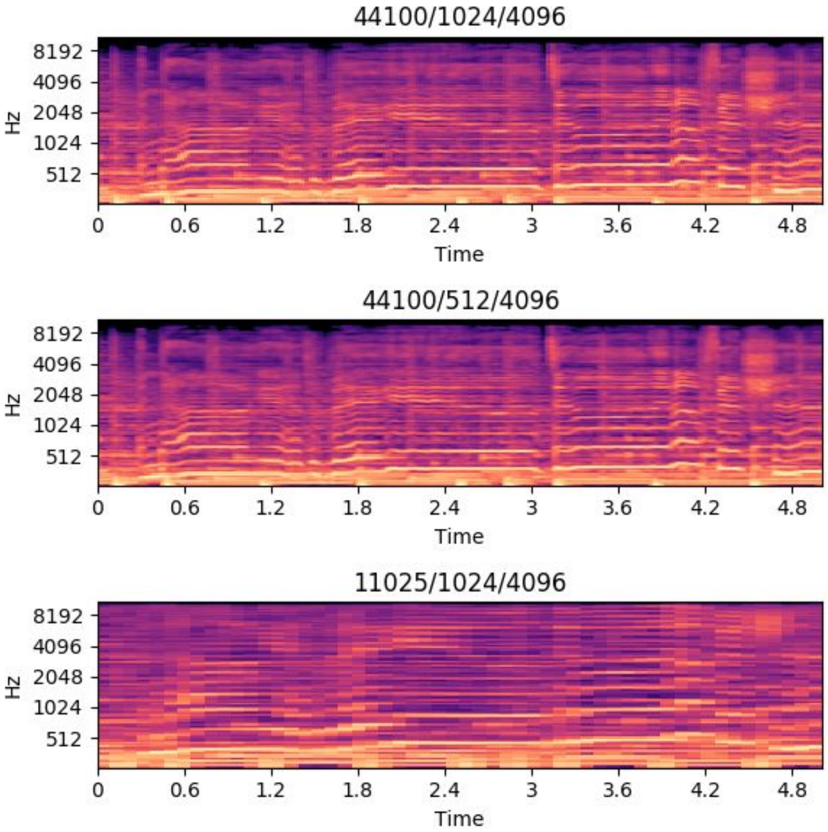
Features (X)	**Labels (y)**
	*Good Morning!*
Audio wave	Transcript





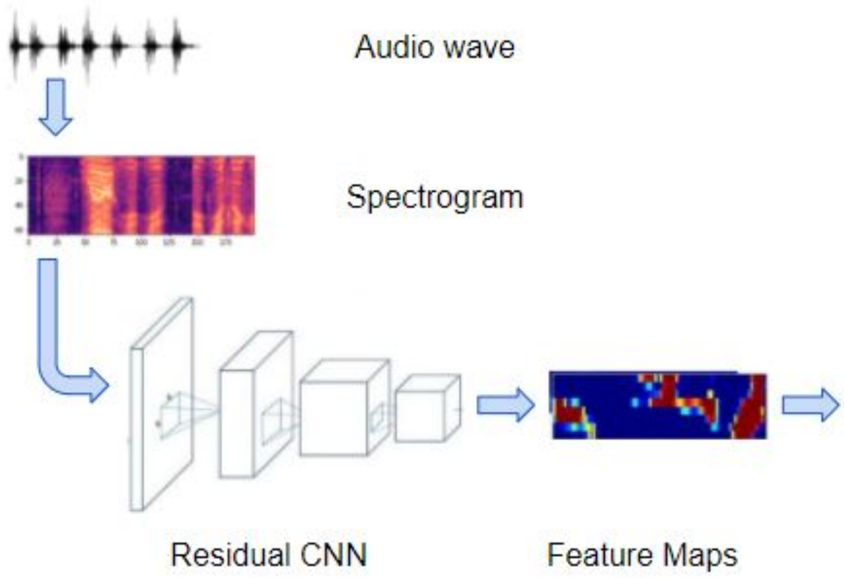
*.wav files
Load Audio from .wav file
Channel 1
Samples
Resample and Convert to Stereo
Channel 1
Channel 2
Resize to Fixed Length
Audio Augmentation Time Shift
Samples
Time steps
Freq
Convert to Mel Spectrogram
Spectrogram Augmentation SpecAugment
MFCC
Labels (y)
Transcript
Good Morning!
3 8 8 ... 9
Char IDs

Library librosa for transformation from audio file to spectogram

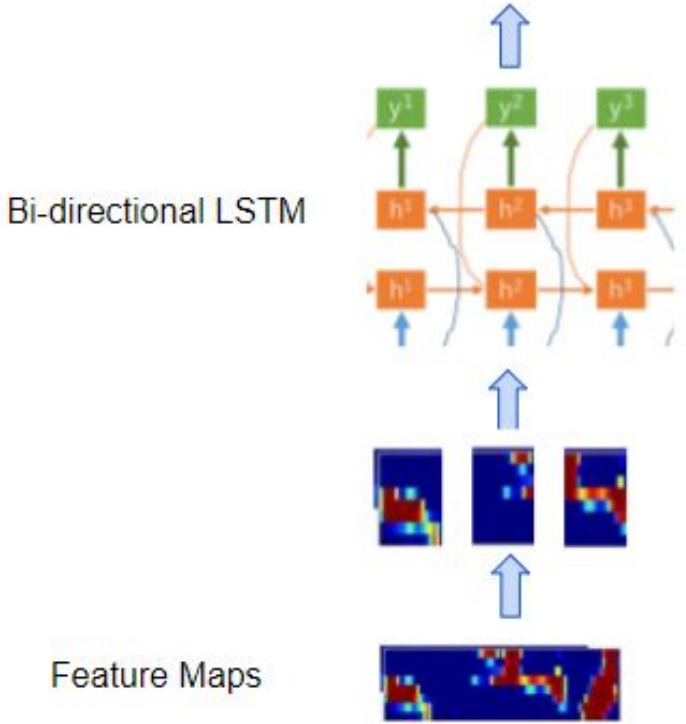


Audio to feature map transformation

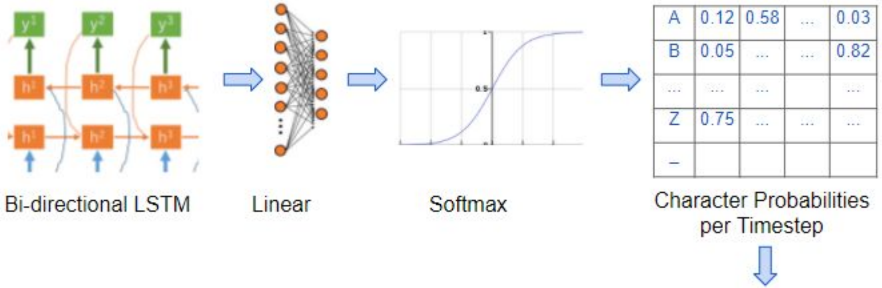
Regular convolutional network consisting of a few CNN layers that process the input spectrogram images and output feature maps of those images.

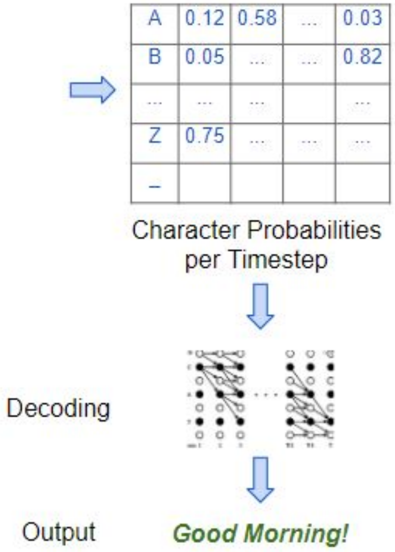


Regular recurrent network consisting of a few Bidirectional LSTM layers that process the feature maps as a series of distinct timesteps or ‘frames’ that correspond to our desired sequence of output characters.



A linear layer with softmax that uses the LSTM outputs to produce character probabilities for each timestep of the output.





Mapping the timesteps to individual characters in target transcript.

Hidden Markov Models

- Arranges phonemes in the right order by using statistical probabilities. The structure is expressed in three layers
- In the first layer, the model has to check the acoustic level and the probability that the phoneme it has detected is the correct one.
- In the second layer, the model checks phonemes that are next to each other and the probability that they should be next to each other.
- In the third layer, the model checks the word level. That is, whether words next to each other make sense. It does this by checking the probability that they should be next to each other.

References

- https://librosa.org/librosa_gallery/auto_examples/plot_presets.html#sphx-glr-auto-examples-plot-presets-py
- https://github.com/caiomiyashiro/music_and_science/blob/master/Chord%20Recognition/presentation_pydata_hidden_markov_models_for_chord_recognition.ipynb
- https://www.analyticsvidhya.com/blog/2019/07/learn-build-first-speech-to-text-model-python/
- https://towardsdatascience.com/audio-deep-learning-made-simple-automatic-speech-recognition-asr-how-it-works-716cfce4c706
- https://pytorch.org/hub/snakers4_silero-models_stt/
- https://towardsdatascience.com/text-to-speech-foundational-knowledge-part-2-4db2a3657335