

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
— ΙΔΡΥΘΕΝ ΤΟ 1837 —



M907 Speech Recognition and Speech Synthesis Technologies (Summer Semester 2023) **TTS-Assignment** Report

Kleopatra Karapanagiotou (It12200010)

Table of Contents

Introduction	3
Speech Synthesis Systems: Components	3
Steps to Train a Speech Synthesis System	5
Task Description	8
Methodology	8
Subjects	8
Evaluation Materials	8
Procedure	9
Producing the .wav files	9
Main Evaluation Experiment	9
Additional Evaluation Experiment.....	9
Results	10
Intelligibility.....	10
Naturalness	10
MOS -Results	10
Forced-choice task results.....	11
Discussion on best performing model(s)	11
Acoustic models: Deep Generative Models (Flow-based vs Autoregressive).....	12
Vocoders: Deep Generative models (GANs vs Diffusion)	12
Conclusions and further thoughts.....	12
References.....	14

Introduction

Speech synthesis, also known as text-to-speech (TTS) technology, is an advanced area of artificial intelligence that aims to convert words from a computer document (e.g., word processor document, webpage) into audible speech spoken through the computer speaker. It has diverse applications, such as virtual assistants, audiobooks, accessibility tools for the visually impaired, and more.

Speech Synthesis Systems: Components

In many TTS systems, one often finds a huge list of modules arranged in a pipeline. A typical system of this kind might include the following:

1. Text Processing

The text processing module is responsible for converting raw text input into a format suitable for speech synthesis. It involves various sub-processes:

- a. Sentence splitting of text input (input: a sequence of ascii characters of any length)
- b. Text Normalization: This step ensures that the input text follows consistent punctuation, capitalization, and abbreviations rules to enhance the naturalness of the synthesized speech. This step handles nonstandard words like (numbers, abbreviations, addresses)
- c. Text Tokenization: Text is divided into smaller linguistic units, such as words or phonemes, to facilitate linguistic analysis and synthesis.

2. Frontend Processing

The frontend processing component involves linguistic and acoustic analysis, responsible for capturing the linguistic and prosodic features necessary for generating human-like speech. Key sub-components include:

- a. Grapheme to Phoneme (G2P): This converts the graphemes of words into their corresponding phonetic representation of words, allowing the system to predict the correct pronunciation. Letter to sound rules (easy or hard depending on the language)
- b. Prosodic Prediction: Prosody includes aspects like syllabification, phrase/clause marking, intonation, pitch, and rhythm that convey emotions and contextual information. Prosody prediction models help generate expressive and contextually appropriate speech.
- c. Phonetic Encoding: The text is represented in phonetic form to facilitate synthesis with speech units such as diphones or triphones.

3. Unit selection module (in concatenative speech synthesis)

Selects the most appropriate segments from a large database of pre-recorded speech based on the input text and its linguistic context to synthesize the target sentence. Usually works at diphone level. It aims to find units that create smooth and natural-sounding speech with proper prosody and pronunciation.

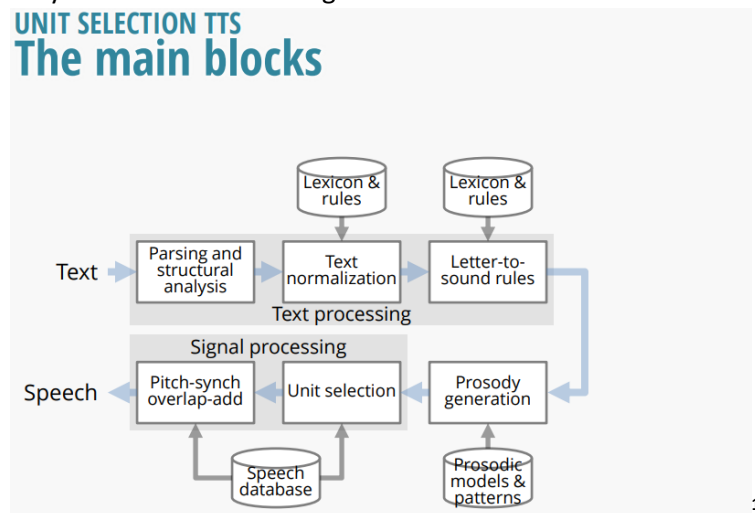
4. Acoustic Model

The acoustic model is a critical component of a Speech synthesis system. It is trained to map the linguistic features from the frontend processing to acoustic features that represent speech sounds. Common techniques include Hidden Markov Models (HMMs) and Deep Neural Networks (DNNs).

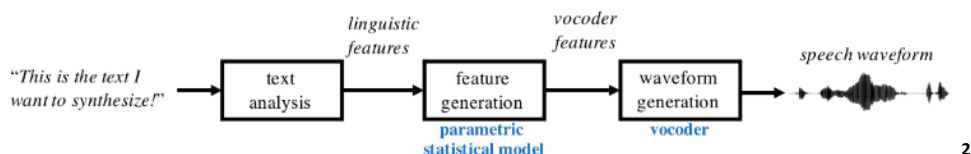
5. Waveform Generation

Waveform generation is the final step that produces the actual speech audio from the acoustic features obtained in the previous stages. Several methods are used for this purpose:

- a) **Concatenative Synthesis (Pitch-synch overlap-add):** This method stitches together short, pre-recorded speech segments (units like diphones or triphones) to form continuous speech. Although it can produce natural-sounding speech, it requires a vast database of high-quality recordings. An example of a TTS system generated through concatenative synthesis is the following:

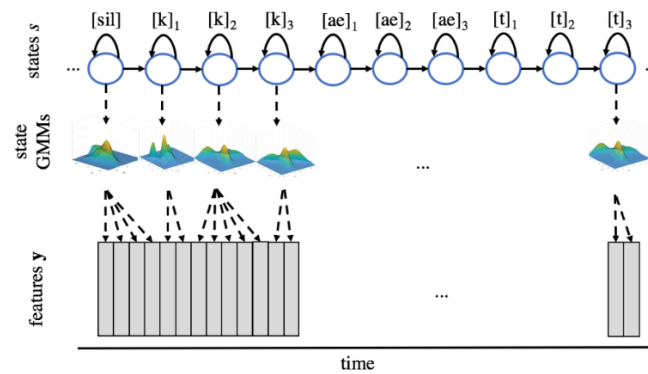


- b) **Statistical Parametric Synthesis (SPSS):** Parametric synthesis techniques use mathematical models to generate speech, where speech parameters are predicted by the acoustic model and converted into waveform signals. This approach requires less storage than concatenative synthesis. A standard approach for the probabilistic mapping has been to use an HMM-GMM as the statistical parametric model.



¹ Raptis, S. (n.d.). *Unit selection-based TtS* [Review of *Unit selection-based TtS*]. Retrieved September 5, 2023, from https://opencourses.uoc.gr/courses/pluginfile.php/16927/mod_resource/content/1/SPCC15_Raptis.pdf

² Räsänen, O. (2020). *SPSS_basic_pipeline* [Aalto University Wiki]. <https://wiki.aalto.fi/display/ITSP/Statistical+parametric+speech+synthesis>



State chain: $P(s_t | s_{t-1})$

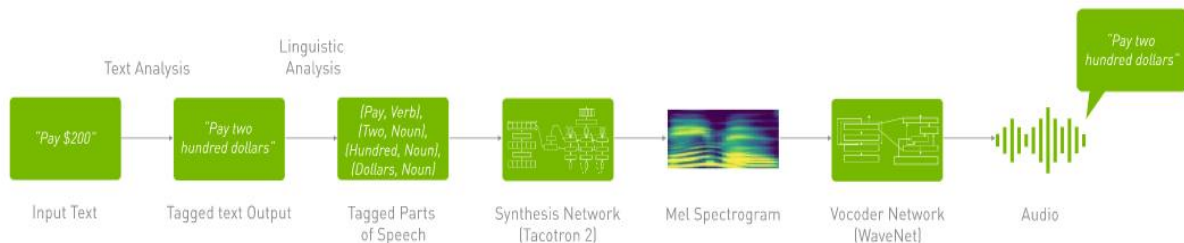
State emissions: $P(y_t | s_t) = \sum_{k=1}^K \phi_{k,s} N(y_t | \mu_{k,s}, \Sigma_{k,s})$

3

c) Parametric Neural networks (State-of-the-art)

Recent advancements have seen the rise of neural waveform generation techniques, such as WaveNet and WaveGlow, which use Deep Neural Networks to generate high-quality and natural-sounding speech directly from text input.

- **Tacotron2 → Text to Mel:** A synthesis network (Tacotron 2) learns a context-based mapping from characters to Mel scale spectrograms (intermediate audio representations), which are the frequency at which vocal cords vibrate in voiced sounds.
- **WaveNet → Mel to Speech:** A vocoder network (WaveNet) inverts the Mel spectrogram back to speech, by learning a function that maps from the frequency domain to the time domain.



4

Steps to Train a Speech Synthesis System

Training a TTS involves several stages, requiring substantial data and computational resources. Here is an overview of the process:

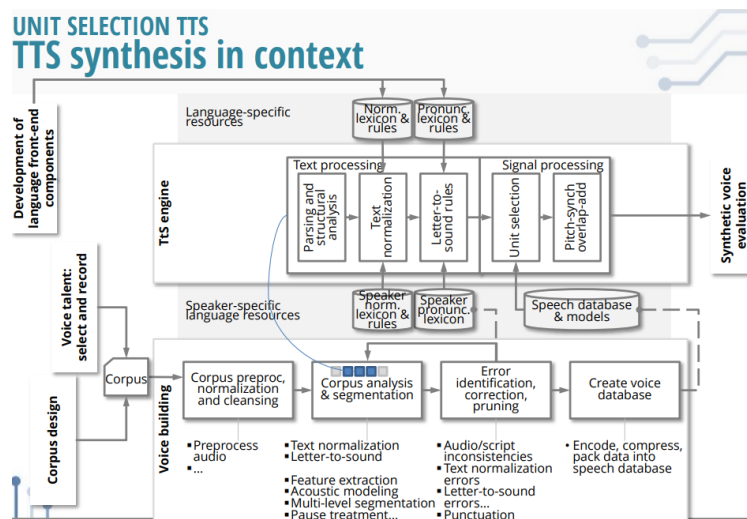
³ Räsänen, O. (2020). *Synthesis_HMM_GMM* [Aalto University Wiki].

<https://wiki.aalto.fi/display/ITSP/Statistical+parametric+speech+synthesis>

⁴ A vocoder network converts the time-aligned features into audio waveforms [NVIDIA A vocoder network converts the time-aligned features into audio waveforms]. Retrieved September 5, 2023, from <https://www.nvidia.com/en-us/glossary/text-to-speech/>

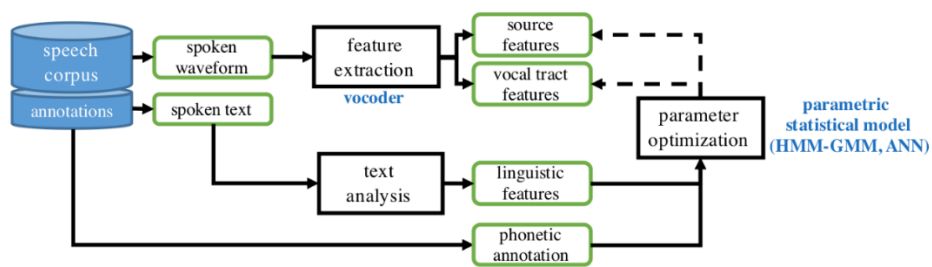
1. **Corpus Design:** A large and diverse corpus of text and corresponding speech recordings is collected. The corpus should cover various speakers, accents, and speaking styles to ensure the system's versatility.
2. **Corpus Preprocessing:** The collected speech data is segmented into smaller units (phonemes, diphones, or other speech units) and aligned with their corresponding text.
3. **Text and Acoustic Feature Extraction:** The preprocessed text data is passed through the frontend processing to extract linguistic and prosodic features. Simultaneously, the speech data is transformed into acoustic features, such as Mel Frequency Cepstral Coefficients (MFCCs) or mel-spectrograms.
4. **Acoustic Model Training:** Using the extracted acoustic features and corresponding linguistic features, the acoustic model (usually a deep neural network) is trained to map the linguistic input to acoustic output.
5. **Prosody Model Training:** If the system uses a separate model for prosody prediction, this model is trained to predict prosodic features based on the linguistic input.
6. **Waveform Generation Model Training:** If the system uses a neural waveform generation method, such as WaveNet, a separate model is trained to convert acoustic features into waveform signals.
7. **Joint Training (Optional):** In some systems, the acoustic model and waveform generation model can be trained jointly, optimizing the overall performance.
8. **Post-processing:** The synthesized speech may undergo additional processing to remove artifacts, normalize volume, and improve overall sound quality.
9. **Evaluation and Fine-tuning:** The trained system is evaluated using various techniques (glass-box vs. black-box evaluation, laboratory vs field, symbolic vs acoustic level, human vs. automated, judgement vs. functional testing, global vs. analytic assessment). If necessary, the system is also fine-tuned to improve its performance.

The following images demonstrate the training in 3 different TTS approaches (Unit Selection, Statistical Parametric Speech Synthesis, Neural based (end-to-end) TTS)



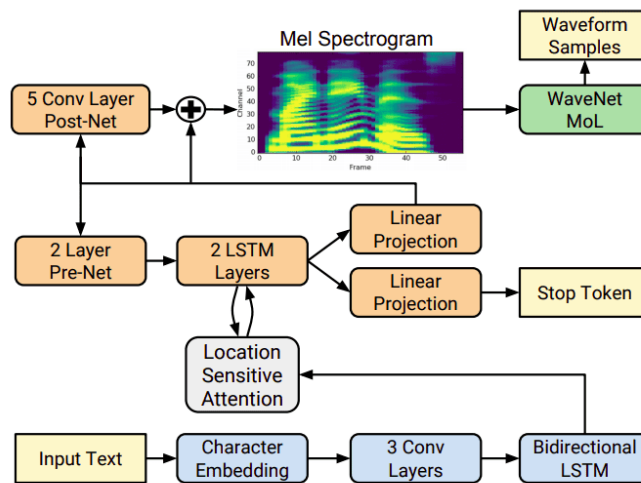
Unit selection training pipeline

⁵ Raptis, S. (n.d.). *Unit selection-based TtS* [Review of *Unit selection-based TtS*]. Retrieved September 5, 2023, from https://opencourses.uoc.gr/courses/pluginfile.php/16927/mod_resource/content/1/SPCC15_Raptis.pdf



6

A schematic view of SPSS system training



7

Block diagram of the Tacotron 2 system training architecture

⁶ Räsänen, O. (2020). *SPSS training pipeline* [Aalto University Wiki].
<https://wiki.aalto.fi/display/ITSP/Statistical+parametric+speech+synthesis>

⁷ *Tacotron 2 architecture* [NVIDIA]. Retrieved September 5, 2023, from
https://catalog.ngc.nvidia.com/orgs/nvidia/resources/tacotron_2_and_waverglow_for_pytorch

Task Description

Use coqui-TTS to synthesize 5 different sentences with 3 different TTS models. Evaluate the synthesis results based on these 3 different examined models in terms of Intelligibility and Naturalness on a scale from 1 to 5. Write a report including the Mean Opinion Scores of a team of (at least) 3 evaluators and further describe the best performing model in your experiments.

Methodology

The goal of this specific evaluation process was the comparison between 3 pretrained TTS models and the discussion on the best performing one for multidomain applications. The evaluation was carried out on the systems' final outputs and the aspects evaluated on the tests were intelligibility and naturalness. Both methods are very common in Speech Synthesis experiments. Speech intelligibility can be defined as how clearly a person speaks so that his or her speech is comprehensible to a listener. It measures the **word accuracy** in sentence transcription, and it mainly focuses on the system's performance. Measuring intelligibility involves calculating the **percentage of words understood** from a continuous speech sample (Kent et al., 1994; Kwiatkowski & Shriberg, 1992). For this purpose, a simple, listener task was designed, which assesses intelligibility on the sentence level ("Type-in the words you heard"-task). Naturalness on the other hand has a vaguer definition, but it is mainly connected to the listeners' perception of the degree to which speech meets the typical patterns in terms of intonation, voice quality, rate, rhythm, and intensity, with respect to the syntactic structure of the utterance. Naturalness was measured on a 5-point Lickert scale and evaluated based on the Mean Opinion Score per sentence and based on a forced choice task in which the listener got to choose 1 out of the 3 sentence versions, where each version represents a different TTS model.

Subjects

The participants in our experiment setting were proficient (C2) and native speakers of English (US/UK) from various EU nationalities.

Evaluation Materials

We aimed for comparing 3 tts models in terms of multi-domain coverage. For this purpose, we used 5 different sentences from 5 different domains.

audiobooks: 1st Person-Direct speech, Declarative: expresses a statement.

News: 3rd person, Indirect speech, declarative: expresses a statement.

User manual: 2nd person, direct speech, imperative: expressing negation.

Weather forecast: 3rd person, indirect speech, declarative: expressing a statement.
















SUS: Semantically Unpredictable Sentence to avoid the ceiling effect⁸. This sentence is a typical example used in Linguistics to prove and point out the need for semantic analysis of an utterance along with the syntactic one and in our case, it is expressed in interrogative form

⁸ <https://www.scribbr.com/research-bias/ceiling-effect/>

Text	Source
"Now, I am become death, the destroyer of worlds"	Oppenheimer-movie (2023)
"With slow advance Ukraine aims for better shot at Russian targets "	nytimes.com
"Please do not remove the battery, while the phone is updating, as it may cause software issues and void the warrant"	Chat-GPT prompt
"Tomorrow the weather forecast predicts clear skies and high temperatures throughout the day with no chance of rain"	Chat-GPT prompt
"Why do colorless green ideas sleep furiously?"	Noam Chomsky (1957)

Procedure

Producing the .wav files⁹

Domain	audio-book	News headlines	User Manual	Weather forecast	SUS
Glow TTS + Multiband-MelGAN	 sentence_0.wav	 sentence_1.wav	 sentence_2.wav	 sentence_3.wav	 sentence_4.wav
Tacotron2 + Wavegrad	 sentence_0.wav	 sentence_1.wav	 sentence_2.wav	 sentence_3.wav	 sentence_4.wav
Capacitron-t2-c50 +HiFiGAN 2	 sentence_0.wav	 sentence_1.wav	 sentence_2.wav	 sentence_3.wav	 sentence_4.wav

Main Evaluation Experiment

Test design
<p>The tests were designed in such a way, so that all participants get to listen all the stimuli. To avoid priming and ordering effects, especially for the intelligibility part of the test, 3 separate Google forms for each of the 3 tts models (glow-tts, tacotron2, capacitron) were generated¹⁰.</p> <p>Sample size: 15 (5 participants for each model/Google form)</p> <p>Test duration: 3 min</p> <p>Number of stimuli per listener: 5</p>

Additional Evaluation Experiment

Test design
<p>A within-subjects test for ranking the 3 tts systems in naturalness (forced choice task: all listeners hear the same 3 TTS versions for each sentence, and they are asked to choose one of them in terms of naturalness)¹¹</p> <p>Sample size: 5</p> <p>Test duration: 5 min</p> <p>Number of stimuli per listener: 15</p>

⁹ For generating synthetic sentences, refer to the folder: **Code<tts.py**

¹⁰ Capacitron: <https://forms.gle/kLorKz5PGxno2TU36>

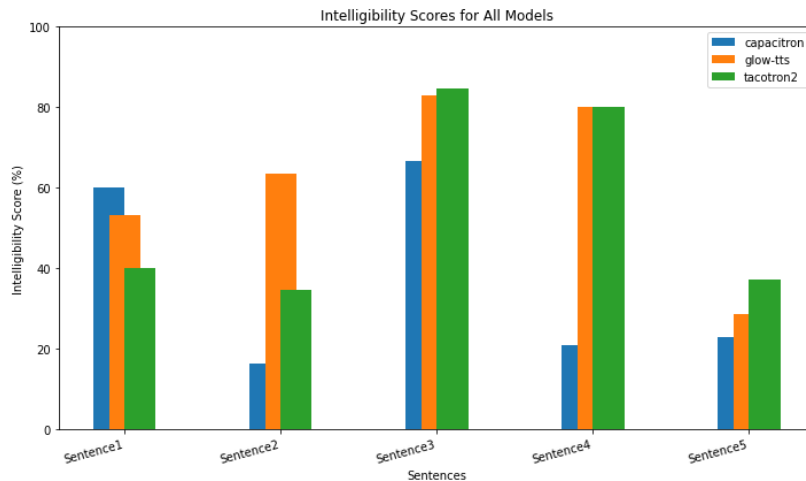
Glow-TTS: <https://forms.gle/uLtovxwoBwsGbPzP6>

Tacotron2: <https://forms.gle/fGJuHMqM8x2W7aBj9>

¹¹ TTS-Comparison: <https://forms.gle/GFBSxVtGjbGPbLUP6>

Results

Intelligibility



12

The results show inconsistency about the best performing model.

For the audio-book line (sent1), it was an expected behavior to see the Capacitron model at the top, since it was trained on expressive data, and designed for such domain.

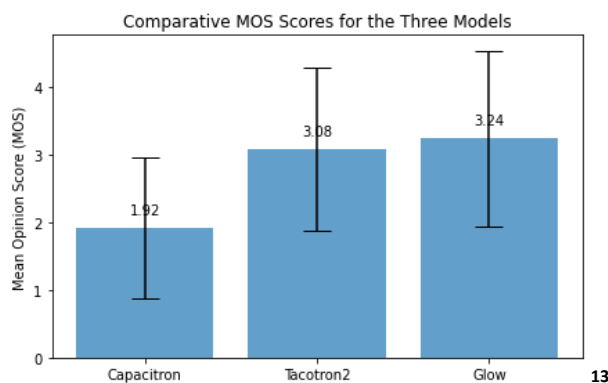
In the news domain (sent2), it seems that glow-TTS showed significantly better performance.

Whereas for the rest 3 domains (user manual: sent3, weather forecast: sent4, SUS: sent5) Tacotron2 model results were slightly more understandable compared to the glow-tts model and far model understandable compared to the Capacitron model.

It is worth noting that there was a statistically significant difference in the mean sentence length, which (as per listeners' feedback) affected them completing the intelligibility task correctly, even though there was no restriction on the number of times the participants could listen the stimuli. («Some sentences were too long, and I could not remember what was being said»). That being mentioned, we should take into consideration the small difference of glow tts and Capacitron2 and extend our experiments with more sentences of the same length to be able to reach more robust conclusions.

Naturalness

MOS -Results



13

¹² For the code refer to the folder path: **Code<intelligibility+plots.py**

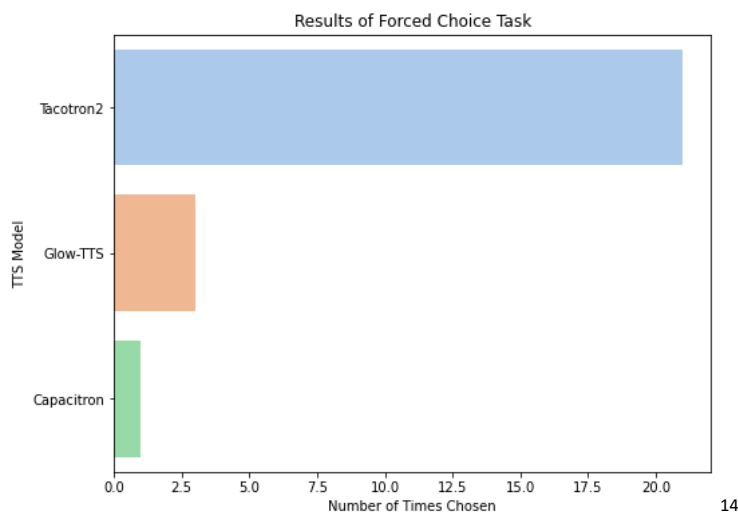
¹³ For the code refer to the folder path: **Code<MOS.py**

The results of the Naturalness task show **the Tacotron2 with WaveGrad vocoder** and the **Glow TTS with multiband MelGAN vocoder** again battling for the first place with the second one, winning with a real small difference. It appeared that both these 2 model and vocoder combinations gave the highest MOS scores but also the highest confidence intervals. This means that although on average, the raters found the output of Glow TTS and Tacotron2 to be of better quality in terms of naturalness, at the same time the wide confidence interval, that accompanies their scores, automatically cancels out our certainty about the true MOSs of our experiments. This should be expected, as we attempted to cover 5 different domains, with only one sentence instance from each of them. So, the variability in rater opinions along with the small sample size, were 2 factors that weakened our experiment setting, leaving us with uncertainties about the true model performances.

Forced-choice task results.

	Sentence 1	Sentence 2	Sentence 3	Sentence 4	Sentence 5
Glow	-	33%	40%	-	33%
Tacotron2	100%	67%	60%	67%	67%
Capacitron	-	-	-	33%	-

These findings indicate a preference for «Tacotron2» in the majority of the sentences, suggesting that it may be the most favorable version in this forced-choice scenario. This is verified by the following concentrative table:



However, the varying preferences for specific sentences highlight the importance of considering the nuances of each version's performance in future evaluations.

Discussion on best performing model(s)

It was obvious from our experiments that the 2 spectrogram models, Tacotron2 and Glow tts were competing for the first place in both naturalness and intelligibility tasks. However, as per literature, Glow-TTS differs from Tacotron 2 in several ways, since the former was initially designed to overcome some limitations of the latter, so it is worth pointing out their differences.

Acoustic models: Deep Generative Models (Flow-based vs Autoregressive)

Glow-TTS is a flow-based generative model for parallel TTS that does not require any external aligner, whereas Tacotron 2 is an autoregressive TTS model. Autoregressive models generate speech by predicting one acoustic feature at a time, conditioned on the previously generated features and the input text. This makes them slow and computationally expensive, especially when generating long utterances. Generative flows, on the other hand, generate speech by applying invertible transformations to a latent representation of speech, which is conditioned on the input text. This allows them to generate speech in parallel, which makes them much faster than autoregressive models. Additionally, autoregressive models can sometimes produce serious attention errors when the input text includes repeated words. Whereas, generative flows can be trained to maximize the likelihood of the data, which makes them easy to optimize. Finally, generative flows can be used to synthesize diverse and controllable speech, which is difficult to achieve with autoregressive models.

Vocoders: Deep Generative models (GANs vs Diffusion)

Both TTS models in our experiments used a type of a generative model for the vocoder part, these were their default vocoders in coqui-tts. Glow TTS used the Multi-band MelGAN and Tacotron2 used the WaveGrad vocoder. Both vocoders are designed to generate high-quality speech waveforms from intermediate acoustic representations, such as mel-spectrograms. But they do have differences worth pointing out.

1. In terms of architecture, Multiband MelGAN relies on a generative adversarial network (GAN) with a multi-resolution generator, which generates different frequency bands separately and combines them for the final waveform. WaveGrad, on the other hand, is an autoregressive model that refines audio waveforms sample by sample.
2. Their training approaches also differ, with Multiband MelGAN using a GAN-based objective involving a generator and discriminator, while WaveGrad employs an autoregressive training process, predicting one audio sample at a time.
3. In the context of parallelism and complexity, Multiband MelGAN offers faster inference due to its parallelizable generation process, while WaveGrad, being inherently autoregressive, can be slower during inference and computationally more demanding.
4. Regarding audio fidelity, WaveGrad excels, producing audio with very high fidelity and minimal artifacts, matching one of the best autoregressive models (Kalchbrenner et al., 2018) in terms of subjective naturalness.¹⁵ This was proven by the forced choice task results in the Tacotron2 model with WaveGrad vocoder. In contrast, Multiband MelGAN aims for high-quality audio but may have slightly lower fidelity in some cases.

Conclusions and further thoughts

The above two TTS model and vocoder combinations gave us comparable results on the two subjective measures for evaluation (naturalness/intelligibility). It appeared, though, that including many parameters in the experiment setting (e.g., multi-domain coverage and multiple sentence-length coverage) with a single sample sentence for each domain was not a smart idea for a robust comparison of the 3 TTS models. Furthermore, our findings indicate that for the design of a TTS system for multidomain applications, we possibly need to consider more parameters in the architecture of our chosen model. To overcome the limitations of the above experiment setting, the following steps are suggested for future experimentation.

¹⁵ For reference, you can listen to some audio samples in Soundcloud:

<https://soundcloud.com/sanjaesc-395770686/sets/tacotron2-tts-wavegrad-vocoder>

1. Rerun the experiments with sentences of the same length, to ensure the intelligibility task is completed with no bias in favor of shorter sentences.
2. Rerun the experiments with sentence of one specific domain and experiment with more possible utterances and domain-specific semiotic classes such as addresses, dates, numbers.
3. Also, the random selection of TsS models was a factor that may have probably blurred the performance results on the synthesized sentences. It would be worth investigating TtS models for the domain and the purpose they were developed for. E.g by limiting the experimentation in the audiobook domain, we could experiment with models of the Blizzard Challenge (Blizzard 2013) to see how they differentiate the results on the synthesized sentences.
4. Apart from spectrogram models, we can try out E2E models that contain no vocoders in their architecture (like VITS)
5. Experiment Combining Tacotron2 with attention methods, like DDC (Double Decoder Consistency)¹⁶ and DCA (Dynamic Convolutional Attention)¹⁷.
6. Try out objective forms of evaluation for synthetic speech, like: Mel Cepstral Distortion or Root Mean Square Error of F0.

¹⁶ <https://coqui.ai/blog/tts/solving-attention-problems-of-tts-models-with-double-decoder-consistency>

¹⁷ <https://arxiv.org/pdf/1910.10288.pdf>

References

- Audio samples from “Effective Use of Variational Embedding Capacity in Expressive End-to-End Speech Synthesis.” (n.d.). Google.github.io. Retrieved September 7, 2023, from <https://google.github.io/tacotron/publications/capacitron/index.html>
- English female TTS Model capacitron t2 c50 Encoding Trained on blizzard2013 Dataset at 24000Hz - speech-synthesis - text-to-speech - AI Models Hub. (n.d.). AI Models Hub. Retrieved September 7, 2023, from <https://aimodels.org/models/text-to-speech/speech-synthesis/English-female-TTS-Model-capacitron-t2-c50-Encoding-Trained-on-blizzard2013-Dataset-at-24000Hz>
- Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search. (n.d.). Slideslive.com. Retrieved September 7, 2023, from <https://slideslive.com/38938507/glowtts-a-generative-flow-for-texttospeech-via-monotonic-alignment-search>
- J. Robert Oppenheimer: “I am become Death, the destroyer of worlds.” - YouTube. (2011, August 6). Ww.youtube.com. https://www.youtube.com/watch?v=lb13ynu3lac&ab_channel=PlenilunePictures
- Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., Oord, A. van den, Dieleman, S., & Kavukcuoglu, K. (2018). Efficient Neural Audio Synthesis. *ArXiv:1802.08435 [Cs, Eess]*. <https://arxiv.org/abs/1802.08435>
- King, S. (n.d.). *Evaluating Speech Synthesis*. Retrieved September 7, 2023, from https://speech.zone/media/images/Simon_King_Crete2016_3_speech_synthesis_evaluation_for_publication.pdf
- Kwiatkowski, J., & Shriberg, L. D. (1992). Intelligibility Assessment in Developmental Phonological Disorders. *Journal of Speech, Language, and Hearing Research*, 35(5), 1095–1104. <https://doi.org/10.1044/jshr.3505.1095>
- PyTorch. (2019). Pytorch.org. https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/
- Taubert, S. (n.d.). *mean-opinion-score: Library for calculating the mean opinion score and 95% confidence interval of the standard deviation of text-to-speech ratings according to Ribeiro et al. (2011)*. PyPI. Retrieved September 7, 2023, from <https://pypi.org/project/mean-opinion-score/>
- Raptis, S. (n.d.). *Unit selection-based TtS*. Retrieved September 7, 2023, from https://opencourses.uoc.gr/courses/pluginfile.php/16927/mod_resource/content/1/SPCC_15_Raptis.pdf
- *Statistical parametric speech synthesis - Introduction to Speech Processing - Aalto University Wiki*. (n.d.). Wiki.aalto.fi. Retrieved September 7, 2023, from <https://wiki.aalto.fi/display/ITSP/Statistical+parametric+speech+synthesis>
- *Tacotron2 TTS + WaveGrad Vocoder*. (n.d.). SoundCloud. Retrieved September 7, 2023, from <https://soundcloud.com/sanjaesc-395770686/sets/tacotron2-tts-wavegrad-vocoder>
- Tae, J. (2022, April 11). *Glow-TTS*. Jake Tae. <https://jaketae.github.io/study/glowtts/>
- *What is Text to Speech?* (n.d.). NVIDIA Data Science Glossary. Retrieved July 20, 2023, from <https://www.nvidia.com/en-us/glossary/text-to-speech/>