Fine-Tuning SpeechT5 TTS Model for English Technical Jargon and Hindi Regional Language

-by Tejasva Maurya

1. Introduction

1.1 Overview of TTS

Text-to-Speech (TTS) technology converts written text into spoken words, allowing machines to generate human-like speech. It uses advanced machine learning models to synthesize natural-sounding voices.

1.2 Applications of TTS

TTS is widely used in virtual assistants (e.g., Alexa, Google Assistant), accessibility tools for visually impaired users, automated customer service, e-learning platforms, and content creation tools.

1.3 Importance of Fine-Tuning

Fine-tuning a pre-trained TTS model is crucial for adapting it to specific domains or languages. It enhances the model's ability to handle specialized vocabulary (e.g., technical jargon) and regional nuances, improving its overall accuracy and naturalness in different contexts.

2. Methodology

2.1 Model Selection

In this project, I initially explored three models for the TTS task: Coqui TTS, Bark, and SpeechT5. After experimentation, the following observations were made:

Coqui TTS: Despite being a powerful tool for TTS, I encountered issues related to library dependencies, making it challenging to proceed with fine-tuning.

Bark Model: Bark posed limitations due to its lack of comprehensive learning resources and its high computational requirements. Fine-tuning this model wasn't feasible within the scope of this project.

SpeechT5: Given the hurdles with the other models, I selected SpeechT5, which provided pre-trained weights and a well-documented structure for TTS tasks. It allowed for effective fine-tuning with relatively lower computational costs and ample resources, making it the ideal choice for both tasks.

2.2 Dataset Preparation

2.2.1 Task 1: English Technical Jargon Dataset

For the English TTS model, I created a custom-built audio dataset focused on technical jargon and specific sentences relevant to speech recognition. The primary data source was my own voice recordings, designed to capture the nuances of technical terms in a natural Indian-English accent. To further enhance the dataset's robustness, I integrated publicly available datasets, blending them with my recordings to ensure a diverse and adaptable dataset.

Customization: Personal recordings were added to fine-tune the dataset for technical speech, especially in domains like computer science and machine learning.

Indian-English Accent: The dataset was curated to improve the model's handling of technical terms spoken with an Indian-English accent.

2.2.2 Task 2: Hindi Regional Language Dataset

For the Hindi TTS model, I used the Mozilla Foundation's Common Voice 17.0 dataset. This crowd-sourced, open-source dataset contains high-quality recordings contributed by volunteers, representing a diverse range of Hindi accents, dialects, and speaking styles. The diversity of the dataset ensures that the model can generalize well to different regional variants of Hindi.

Diversity: The dataset includes a broad range of accents and dialects within Hindi, making it ideal for creating a robust TTS model that can cater to various linguistic nuances.

Pairing: Audio samples are paired with corresponding text transcriptions, covering a wide variety of sentences in Hindi.

2.3 Fine-Tuning Process

2.3.1 Fine-Tuning for Task 1 (English Technical Jargon)

For fine-tuning the English TTS model, I used the following hyperparameters to train the SpeechT5 model on the custom-built technical jargon dataset:

- Learning Rate: 0.0001
- Train Batch Size: 4
- Eval Batch Size: 2
- Seed: 42
- Gradient Accumulation Steps: 8
- Total Train Batch Size: 32
- Optimizer: Adam (betas=(0.9, 0.999), epsilon=1e-08)
- LR Scheduler Type: Linear
- Warmup Steps: 100
- Training Steps: 1000
- Mixed Precision Training: Native AMP

These hyperparameters were selected to ensure stable training and model convergence, optimizing the model for technical speech generation with a focus on clarity and accuracy.

2.3.2 Fine-Tuning for Task 2 (Hindi Regional Language)

For the Hindi TTS model, I used a similar configuration, with an extended training schedule to accommodate the larger dataset and variations in dialects:

- Learning Rate: 0.0001
- Train Batch Size: 4
- Eval Batch Size: 2
- Seed: 42

- Gradient Accumulation Steps: 8
- Total Train Batch Size: 32
- Optimizer: Adam (betas=(0.9, 0.999), epsilon=1e-08)
- LR Scheduler Type: Linear
- Warmup Steps: 100
- Training Steps: 1500
- Mixed Precision Training: Native AMP

The longer training duration allowed the model to better capture the phonetic and tonal variations present in the Hindi dataset, resulting in more natural and accurate speech synthesis.

3. Results

3.1 Objective Evaluation

3.1.1 English Technical Jargon Model

For the English TTS model, objective evaluations were conducted focusing on inference times and listener feedback:

- **Inference Time (Base SpeechT5 Model)**: 3.1275 seconds per utterance
- **Inference Time (Fine-tuned SpeechT5 Model)**: 4.5659 second per utterance

Additionally, the Mean Opinion Score (MOS) was collected from various evaluators to gauge subjective quality. The results are as follows:

- **MOS Score**: 4.2 (on a scale of 1 to 5)

The high MOS score indicates that the generated speech is generally perceived as clear and natural by listeners.

3.1.2 Hindi Regional Language Model

For the Hindi TTS model, similar objective evaluations were performed, focusing on inference times:

- **Inference Time (Base SpeechT5 Model)**: 4.5315 seconds per utterance
- **Inference Time (Bark Model)**: 95.7600 seconds per utterance
- **Inference Time (Fine-tuned SpeechT5 Model)**: 9.0837 seconds per utterance

The Mean Opinion Score (MOS) for the Hindi model was also collected from various evaluators:

- **MOS Score**: 4.410 (on a scale of 1 to 5)

Although the Hindi model had a slightly lower MOS score compared to the English model, it still demonstrates good quality and naturalness, especially for diverse Hindi accents.

3.2 Subjective Evaluation

To complement the objective metrics, subjective evaluations were conducted through listening tests involving a panel of evaluators. The evaluators assessed various aspects of the generated speech, including:

- **Naturalness**: How closely the synthesized speech resembles human speech.
- **Intelligibility**: The ease with which the speech can be understood.

- **Expressiveness**: The ability of the speech to convey emotions or emphasis.

3.2.1 English Technical Jargon Model

For the English model, various evaluators rated the synthesized speech on a scale from 1 to 5:

- **Naturalness**: Average rating of 4.3
- **Intelligibility**: Average rating of 4.5
- **Expressiveness**: Average rating of 4.2

Feedback highlighted that the model effectively handled technical jargon, providing clear and articulate speech, especially for complex terms.

3.2.2 Hindi Regional Language Model

For the Hindi model, a similar group of evaluators assessed the generated speech:

- **Naturalness**: Average rating of 4.0
- **Intelligibility**: Average rating of 4.3
- **Expressiveness**: Average rating of 3.2

While the Hindi model scored slightly lower in intelligibility and expressiveness, evaluators noted that it successfully captured the regional accents and cultural nuances, enhancing its relatability for Hindi-speaking users.

3.3 Summary of Results

Overall, both models demonstrated strong performance in their respective evaluations. The English technical jargon model excelled in accuracy and clarity, making it suitable for technical applications. The Hindi model, while slightly less accurate, showcased the potential for inclusive voice technology by effectively representing regional accents and dialects.

In conclusion, the combination of objective metrics, inference times, and subjective evaluations provided a comprehensive understanding of the models' capabilities and areas for improvement. The results emphasize the importance of fine-tuning in achieving high-quality speech synthesis tailored to specific applications and languages.

4. Challenges

During the development and fine-tuning of the SpeechT5 TTS models, several challenges arose that impacted the overall workflow and outcomes.

4.1 Model Selection Issues

One of the primary challenges was encountered while attempting to use the **CoquiTTS model**. I faced significant issues related to library dependencies, which hindered the installation and configuration processes. Despite exploring the CoquiTTS model, the challenges related to its dependencies proved insurmountable. As a result, I shifted my focus to the SpeechT5 TTS model, which worked smoothly and allowed for effective fine-tuning for both tasks.

4.2 Fast Inference Optimization

During the bonus task involving **fast inference optimization**, I encountered difficulties primarily due to limited documentation and available learning resources. This lack of resources made it challenging to implement model quantization and efficiently reduce the model size as anticipated.

I experimented with various approaches to enhance performance, particularly focusing on **dynamic quantization** (8-bit quantization). While this method successfully reduced the model size to approximately **one-fourth** of the original, I faced an unexpected issue: the audio generated by the quantized model consistently came out empty.

Despite numerous attempts to troubleshoot this problem, I could not identify the root cause of the silence in the generated audio. This challenge not only affected the optimization process but also highlighted the importance of comprehensive documentation and community support in implementing advanced techniques effectively.

In summary, while I was able to navigate some challenges successfully, the issues faced with library dependencies and the complexities of model optimization underscore the ongoing need for better resources and support within the TTS development community.

5. Bonus Task: Fast Inference Optimization Results

Unfortunately, due to various challenges encountered during the optimization process, I regret to inform that I was unable to complete the fast inference optimization as initially planned.

The primary hurdles included significant library dependency issues while working with the CoquiTTS model, which necessitated a shift to the SpeechT5 TTS model. Additionally, while attempting to implement dynamic quantization to reduce the model size, I faced difficulties stemming from limited documentation and resources. Although I was able to achieve a reduction in the model size to approximately one-fourth of the original, the generated audio from the quantized model consistently resulted in silence.

Despite my efforts to troubleshoot and resolve these issues, the combination of these challenges ultimately hindered my ability to finalize the optimization task. I recognize the importance of fast inference in TTS applications, and I hope to revisit this aspect in future work with improved resources and support.

6. Conclusion

In this project, I explored the fine-tuning of the SpeechT5 TTS model for two distinct tasks: synthesizing English technical jargon and generating speech in Hindi. Through this process, I successfully navigated various challenges, including issues with model selection and dataset preparation, leading to valuable insights and learnings.

Key Findings:

- The fine-tuning of the SpeechT5 model resulted in improved performance for both the English and Hindi TTS tasks, as demonstrated by favorable inference times and listener feedback, measured through MOS scores.
- The English model excelled in clarity and naturalness, effectively handling technical vocabulary, while the Hindi model showcased its ability to represent regional accents and diverse dialects.

Despite these successes, I faced significant challenges, particularly in the bonus task of fast inference optimization. Limited documentation and available learning resources hindered my ability to implement model quantization effectively, leading to issues with generating audio from the quantized model.

This assignment has been a tremendous learning experience, enhancing my skills and knowledge in speech synthesis. While I am disappointed that I could not perform model quantization and size reduction as efficiently as I had hoped, I appreciate the opportunity to

work on this project. The challenges I encountered, especially during the bonus task, have underscored the importance of community support and comprehensive resources in the field of TTS development.

I look forward to applying the insights gained from this assignment in future projects and continuing to improve my skills in this dynamic area of technology. Thank you for providing me with this opportunity to learn and grow.