Group Members: Willam Holden, Kavya Kathiravan, Alex Darwiche

Motivation

For accessibility, real-time subtitles and transcription, for intelligent assistants, and many other technologies we use daily, speech-to-text models have become an incredibly important tool. OpenAI's Whisper is one of these models. Whisper is a multilingual speech recognition model that achieves high accuracy. The computational needs, however, make it harder to deploy efficiently, especially on devices with limited resources. Reducing the memory and time needed for inference are critical in making Whisper more practical for real-world applications, like live captioning, voice assistants, and mobile transcription on embedded systems. Our project's main focus is on the optimization of Whisper to improve its speed and memory efficiency while maintaining accuracy.

Idea

Whisper works by converting audio, with speech in it, into text through a two-stage process: an AudioEncoder and TextDecoder. The AudioEncoder, based on transformer architecture, converts raw audio into a generated latent expression. The TextDecoder then translates this expression into text. Word Error Rate (WER) and Character Error Rate (CER), calculated using the JIWER package, are metrics by which the model's accuracy is typically measured. We will likely continue using these metrics while trying to reduce the model's computational requirements.

We will benchmark Whisper on various devices, starting with the Basemodel. Are two predicted methods of optimizations will be:

1. Reducing Inference Time: Using techniques learned in class - model quantization, parallelization, and memory management of Whisper.
2. Reducing Memory Usage: Using memory optimization techniques to allow the model to run on edge devices.

Our methods will include

- Running benchmarks before optimization to create a baseline for inference speed and memory usage.
- Implementing quantization (e.g., FP16 or INT8) to resources needed for inference.
- Utilizing parallelization and other hardware utilization techniques to improve run time.
- Utilizing pruning and compression to reduce the overall size of the model for smaller devices.
- Evaluating accuracy trade-offs to ensure that optimizations do not significantly degrade performance.

Expected Results

After integrating these optimizations, we expect to see an improvement in Whisper's performance by increasing its inference speed and ensuring faster transcription without significantly impacting the accuracy. We can also expect to reduce memory consumption to allow for efficient deployment on smaller devices with limited resources, such as mobile devices.