Project 11: Improving Speech Recognition Performance using Synthetic Data

Ilias Arvanitakis
New York University
ilarv97@nyu.edu

Kristin Mullaney
New York University
kmm9492@nyu.edu

Alexandre Vives
New York University
av2926@nyu.edu

Michael Picheny
New York University
map22@nyu.edu

Bhuvana Ramabhadran
Google Research
bhuv@google.com

Brendan Lake
New York University
bl1611@nyu.edu

Wai Keen Vong
New York University
wv9@nyu.edu

Abstract

This project explores the question of whether synthetic speech can be used to improve the performance of a speech recognition model. The baseline speech recognition model in this project is trained on two hours of well-transcribed audio and achieves a word error rate of 35 percent. The goal of this project is to see if, in the absence of an abundance of high-quality training audio, synthesized speech can be used to lower the word error rate of the existing model.

1 Introduction

This capstone project is part of a larger project under NYU's Human and Machine Learning Lab that does research on speech development. Recordings were gathered through headcams worn by children, referred to collectively as the Saycam dataset. For this dataset to be adequately analyzed, the audio first needs to be transcribed manually. Of the total of roughly 300 hours of audio, 52 hours had been transcribed by a group of human transcribers. However, the computer-generated transcriptions were of variable quality due to the high variance in transcription quality across the multiple transcribers. In addition, in order to be useful for downstream processing, the transcriptions need to be finely

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

time-aligned against the audio, but to simplify the transcription task, this was only done very coarsely and for some recordings, effectively not at all. Therefore, we need a method to easily produce more, better, time-aligned transcriptions.

To address the problem above, mentor Michael Picheny produced a high-quality transcription of two hours of audio. In total, 50 hours of low-quality transcribed audio and two hours of high-quality transcribed audio are available. The first part of automating this process was done by the previous year's capstone. That project fine-tuned a pre-trained speech recognition system from Hugging Face using the high-quality transcriptions. On the test set, the word error rate was 50 percent before fine-tuning. Then the transformer-based pre-trained model was fine-tuned on the manually labeled data which brought down the word error rate to 35 percent. This project is picking up where that project left off. This time, trying to improve the word error rate by incorporating synthetic audio into the training data.

2 Related Work

This project is done in conjunction with Bhuvana Ramabhadran, who currently leads a speech recognition research team at Google. As an expert in the field of synthetic speech, several of Bhuvana's papers served as valuable resources for this project.

Her paper, *Injecting Text in Self-supervised Speech Pretraining*[1], proposes that contrastive learning techniques can be applied to linguistic lexical representations derived from synthesized speech, effectively learning from un-transcribed speech and unspoken text. This method results in a 15% reduction in the speech recognition model's word error rate, which can be decreased by an additional 6% with further calibrations.

In addition, *Tts4pretrain 2.0: Advancing the use of Text and Speech in ASR Pretraining with Consistency and Contrastive Losses*[2], proposes that introducing supervised speech with consistency-based regularization between real and synthesized speech earlier on in the training process allows for better learning of shared speech and text representations. This proposed pre-training method decreases the word error rate by two to 17 percent over previous approaches.

A third paper,*Improving Speech Recognition Using Consistent Predictions on Synthesized Speech* [3], demonstrates that promoting consistent predictions in response to real and synthesized speech enables significantly improved speech recognition performance. With the addition of a consistency loss term the improvement grows to 17 percent.

Finally, the paper *Improving Speech Recognition Using Consistent Predictions on Synthesized Speech*[4], demonstrates that improvements to speech recognition performance is achievable by augmenting training data with synthesized material. A major observation is that the value of synthesized speech in the training data is drastically less than that of the real speech.

3 Problem Definition and Algorithm

3.1 Task

As stated previously, the provided dataset consists of a total of 52 hours of transcribed audio of a parent speaking to their child. More specifically, 50 hours of the data's associated transcriptions were of poor quality - the text was not properly aligned with the audio, some words are incorrect, etc. The given task is to see if we can improve the performance of a baseline speech recognition system by generating synthetic data and adding it to the baseline models training data. By generating synthetic speech from the 50 hours of poor-quality transcriptions, the audio and text of this generated audio is ensured to be aligned properly for downstream processing.

3.2 Algorithm

This project can be divided into three sections, where each section requires a standalone model.

In the first section, audio of the speaker is fed as input into an out-of-the-box model that generates an embedding representation of the speaker's voice. An embedding is a numeric vector representation of a non-numeric object. In this case, the embedding represents an individual's voice. The perfect vocal embedding would numerically represent every aspect of how a person speaks. For example, the pauses they take between words, how they pronounce different sounds (or phonemes), and the pitch with which they speak. Due to some time and data resource constraints, the encoder that we are using to generate the vocal embedding in this project is an out-of-the-box system created by ESPNet.

In the second section, the vocal embedding of the speaker and text are fed into an out-of-the-box speech synthesizer to generate synthetic audio. This allows us to generate synthetic audio from the poorly transcribed text and to be sure that the audio matches the transcriptions perfectly. This section plays the role of substituting the perfectly transcribed synthetic audio for the poorly transcribed original audio. Once again, due to resource constraints, we are using out-of-the-box speech synthesizers by ESPNet for this section. ESPNet has managed to train 14 different speech synthesizers for out-of-the-box use. We will go further into our selection process in the methodology section below.

In the third and final section, we can use varying amounts of our newly generated audio, either alone or alongside real audio, to train the previous capstone project's speech recognition model. Our team trained many variations of this model, using various embeddings and quantities of synthesized speech. The goal was to observe how these factors (embeddings and quantities of synthesized speech) effect the speech recognition model's word error rate. While any observed results are of value, the ultimate goal is to find a variation that improved speech recognition performance the greatest.

4 Experimental Evaluation

4.1 Data

As previously stated, our data was in the form of headcam audio that was recorded by three children between the ages of 6 and 32 months old. This audio contained a large amount of background noise and was generally poor in quality. Of the 52 hours of transcribed audio, only two hours contained high-quality time stamps. The two hours of audio with high-quality transcriptions was recorded by a single child named Sam and almost solely contained speech from Sam's mother. The high-quality transcriptions consisted of 1000 lines of speech and was put in a training directory alongside transcription time-stamps and all associated audio. This would act as our training data from which to train our baseline speech recognition model. The final 50 hours of transcriptions would be sourced to synthesize varying amounts of speech. This synthetic speech was combined with the baseline training data to see what effect this will have on the word error rate.

4.2 Methodology

We hypothesized that adding synthetic speech to the speech recognition system would improve the word error rate. We also experimented with using different source audio for generating speaker embeddings. We hypothesized that using longer source audio with background noise removed would improve the quality of the embedding, and in-turn have a positive effect on the synthetic speech quality and the model's word error rate.

Our first step was to optimize the hyperparameters of the baseline model. We found the optimal hyperparameters for the baseline model to be a learning rate of 0.0001, 50 epochs, a weight decay of 0.01, a warmup value of 100, and a batch size of 32. To focus on the effect that varying embeddings would have on the word error rate, these hyperparameter settings were fixed for all future experiments.

Our next step was to select an adequate, out-of-the-box synthesizer. To do this, we experimented with 14 different synthesizers provided in the ESPNet toolkit. One technique we tried was to utilize phonetic pangrams, or sentences that contain every sound in the English language. For example, "That quick beige fox jumped in the air over each thin dog. Look out, I shout, for he's foiled you again, creating chaos." We synthesized five phonetic pangrams on each synthesizer and listened to the produced audio to return a quality rating of low, medium, or high. The synthesizers 11,12 and 13 were considered high-quality. While there was a large quality differential between the synthesizers deemed high-quality and the others, there seemed to be a marginal and arbitrary difference between the high-quality synthesizers. As a result, we chose to do all future experiments with the 13th synthesizer. Future work on this project might focus more on the effect of various synthesizers and techniques. However, to best focus on the effects of the sound used for generating the embedding, we will only use the 13th synthesizer for all future experiments.

The next step was to begin experimenting with different embeddings. The quality of the embedding was paramount to generating synthetic audio that resembled the real audio. Therefore, our team proceeded cautiously during the selection of the input audio used for its generation. Our team manually scanned the dataset looking for sentences that contained clearly pronounced words along with phonetic variability. After several scanning rounds, the resulting list of high-quality audio snippets were collected for embedding generation. We then randomly chose a single utterance (sound bite 428) to generate our first embedding from. This utterance was only a 6.7-second sound bite of Sam's mother speaking, and so we referred to this embedding as "Short" throughout the project. Our next embedding was a 47-second-long segment of high-quality speech from Sam's mother. We referred to the embedding generated from this audio as "Long". Finally, we lowered the background noise on the 47-second sound bite. This sound bite was used to generate an embedding that we refer to as "Long Enhanced". In total, we used three different audio files to generate three distinct embeddings for experimentation.

The next phase and final phase in the process was to begin experimenting with model variations. Our first class of experiments will involve experimenting with solely training the model with varying amounts of synthetic data, generated from all three embeddings. Our second class of experiments will involve integrating varying amounts of synthetic data from each of the three embeddings into the training data of the baseline model. It's important that we experiment with adding synthesized audio to a set of training data containing real audio, as the ultimate goal is to assist models that are training with real data, not to replace real data entirely.

4.3 Results

The first experiment involved using each of the three embeddings to synthesize the first 1000 transcriptions from the 50 hours of transcribed text. To prepare a model for training, we then have to make a directory containing all of the audio files that will be used for training, along with a train.tsv file that aligns the name of each audio file with its associated transcript. We then trained three models solely on the 1000 pieces of synthetic audio (with no real audio) to see how much signal the synthesized speech from each embedding seemed to carry. The results for this experiment can be observed in Table 1.

	Short	Long	Long Enhanced
1K Synthetic Only WER	83.8%	76.05%	66.92%

Table 1: Results are averaged across three runs. Transcripts randomly selected and unedited.

After observing that the synthesizer struggled with phrases that were very short, we tried trimming the transcriptions to only contain lines longer than 10 characters and running our initial synthetic-only experiments again. We noticed that this actually had a negative effect on the word error rate of the model, so all future experiments were done with unedited transcripts. Results from this experiment can be observed in Table 2.

	Short	Long	Long Enhanced
1K Synthetic Only WER	94.15%	87.15%	62.97%

Table 2: Results are averaged across three runs. Transcripts randomly selected and trimmed to remove phrases shorter than 10 characters.

As the final synthetic-only set of experiments, we used untrimmed transcripts to synthesize 1K, 2K, 5K, 10K, 15K, 20K, and 25K lines of synthetic speech for each of our three embeddings. We trained the speech recognition system on this audio for each of the three embeddings. Our results are in Figure 1.

After training the model on only synthesized data, we began adding various amounts of synthesized speech to baseline training data. We added the 1K, 2K, 5K, 10K, 15K, 20K and 25K lines of synthetic

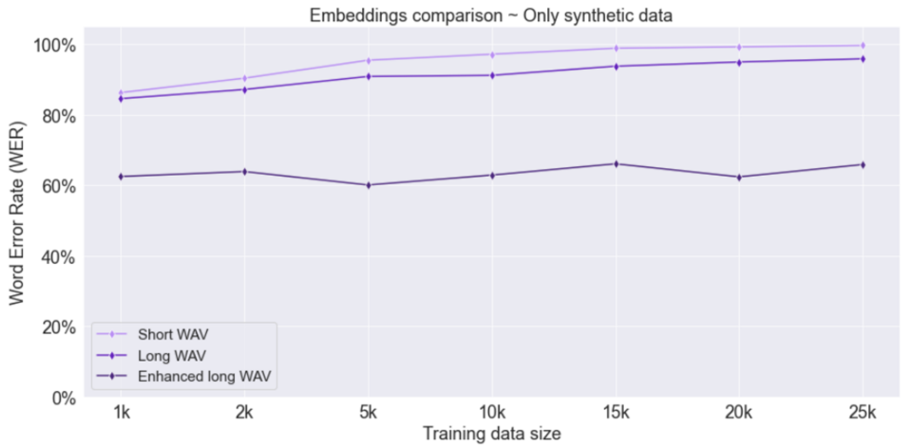

Figure 1: Results are averaged across two runs.

speech to training folders alongside the baseline audio. The results of our experiments are in Figure 2 and Table 3 below.

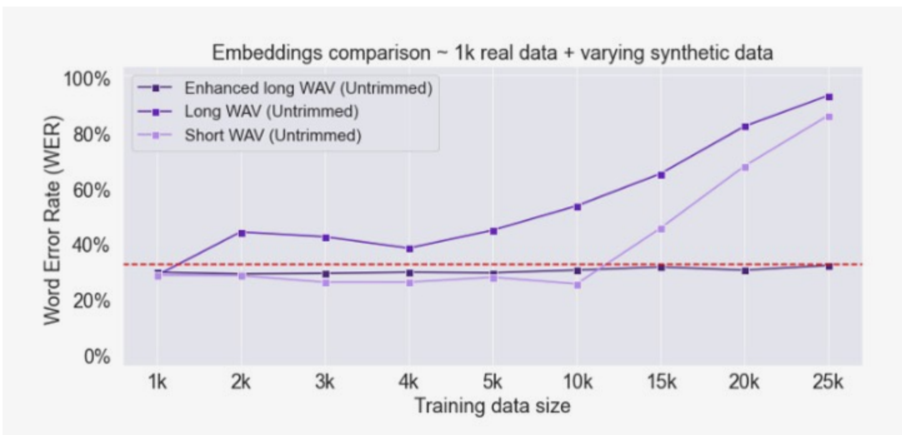

Figure 2: Results are averaged across two runs.

	1K	2K	3K	4K	5K	10K	15K	20K	25K
Short	33.0%	32.9%	31.7%	31.7%	32.6%	31.4%	42.8%	60.4%	80%
Long	32.9%	41.9%	40.8%	38.3%	42.3%	48.4%	57.9%	75.3%	89.3%
Long Enhanced	33.5%	33.2%	33.3%	33.5%	33.4%	33.9%	34.5%	33.9%	34.8%

Table 3: Results are averaged across two runs.

4.4 Discussion

The first interesting finding from our experiments was that the audio generated from the de-noised embedding (Long Enhanced) seemed to carry more signal than the audio generated from the Short or Long 'noisy' embeddings. This was observed across all of our experiments where we only trained the model on synthetic audio. Models trained on the Long Enhanced audio consistently outperformed models trained on the Short and Long audio by about 20 percent. This might be because the background noise in the real audio is very high, and generating an embedding from that audio without any adjustments to the background noise may lead to that noise becoming tangled up inside the representation of the speakers voice. It seems to be a case where noise is quite literally drowning out the signal.

On the other hand, if background noise is consistent enough, you can argue that there is true signal there. Our speech recognition model might benefit from being trained to expect audio blurred by a flurry of other sounds. This feeds into the next interesting observation, that the best performing model across all of our experiments was a model trained on our baseline real audio with 10,000 lines of synthetic data that had been generated from the Short embedding. This model reached a word error rate of 31.4 percent. This initially feels counter-intuitive because the Long Enhanced audio so clearly outperformed the Short and Long audio in the synthetic-only experiments. However, one possible explanation is that keeping background noise in the embedding allows the model to be more flexible and to better take in test data with a lot of background noise.

On the other hand, denoised embeddings might only be able to teach the model information that it has already acquired itself. The embedding might only be able to regurgitate back what it had taken in. Because it adds less randomness to the data, increasing the amount of synthetic data to the model seems to have no effect on the word error rate. However, that lack of randomness might make it less helpful to the model when small amounts of synthetic data are initially added to the baseline. Regardless of the amount of synthetic data applied alongside the baseline training audio, the word error rate stays consistently around 33 percent. This is a 1.7 percent improvement over our baseline, but still 1.6 percent higher than the best performing configuration.

Models trained on audio from both the Short and the Long 'noisy' embedding experienced an initial dip in word error rate with the addition of small amounts of synthetic data, and then an eventual spike when the amount of synthetic data surpassed about 10,000 lines. It's possible that the noise built into this audio might offer a degree of new information to the baseline model in the form of random added information, but that too much randomness begins to throw the model's ability to recognize patterns off.

It's possible that the optimal amount of synthetic audio is proportional to the amount of real training audio available. If that is the case, then it is interesting that a 10-to-1 synthetic-to-real audio ratio would be optimal. This might have something to do with the conclusions reached in *Improving Speech Recognition Using Consistent Predictions on Synthesized Speech*[4]. In that study, it was observed that models do not consider synthetic and real audio to be equal, and in fact, the model naturally favors the information stored in the real audio. This could explain why so much synthetic audio is needed to make a marginal gain in speech recognition performance.

5 Conclusions

Our first conclusion is that synthetic speech can potentially be beneficial to a speech recognition model. Supported by prior research, there is already evidence that speaker independent systems can be improved using synthetic speech. In this project, it is shown that this also applies to speaker dependent systems. In this case the system is dependent the voice of the mother. Thus, synthesizing speech that resembles her voice had an impact on the final WER. Further research can be done on speaker dependent systems with multiple target voices. We could expand to other datasets where there are more than one target voices. A major question is how the model would perform if we added the father of Sam in our dataset. In this case we would have to generate two embeddings and train the model on sound that would combine both parents.

Secondly, it seems that denoising the audio used to generate the embedding makes a significant difference for the synthesizers. Preprocessing the audio, yields a better result, because the synthesizer can better focus on the voice of the target. After manually listening to the audio generated by this embedding, we noticed that the synthesized speech would better resemble the voice of the mother. As a next step we would use professional denoising software that would help us create a better-quality audio as an input to generate the embedding and synthesize higher quality speech.

With synthetic audio that utilized an embedding containing background noise, there was a degree of model improvement with small to moderate amounts of synthetic data. As the amount of synthetic data got larger, the word error rate of the speech recognition model increased greatly. It can be concluded, that with a greater amount of lower quality synthetic data, the effects of the original data tend to be overshadowed. On the contrary, the synthetic audio that utilized the embedding without the background noise, had a consistent WER without a lot of variability.

For future work, we should experiment more with a broader variety of speech synthesis techniques in hopes of producing more realistic synthetic speech. What we used in this project is format synthesis which uses a mathematical model to produce synthesized speech. Other techniques include concatenative synthesis, that involves stringing together smaller pieces of recorded speech and produce more complex utterances. There is also the parametric synthesis technique which uses parameters like the pitch, rate, and tone to create synthesized speech. It is also possible to combine these three methods.

There are ample future experiments to expand upon from this project, but one that is of primary interest is the potential effect of resynthesizing new speech for the model between epochs. Instead of using the same dataset for every epoch, we would resynthesize new data for every epoch. We would expect this to further reduce the WER, since the model would have a wider variety of sounds and words to learn during every iteration.

6 Lessons Learned

Our team has learned several important lessons over the course of our project. One of the most important lessons we learned is the importance of identifying whether tasks are sequential or parallel when dividing work among team members as it can help ensure that the work is completed efficiently and without any unnecessary delays.

Another key lesson we learned is the importance of taking into account the potential for an HPC cluster to become overwhelmed at certain times. Due to its public access, an HPC cluster can sometimes be subject to high levels of usage, which can affect the performance of individual tasks. In order to avoid this, it is important to monitor the cluster's usage and adjust our work accordingly.

Finally, we learned the value of consulting with an expert in the field before getting started on a project. This can help us develop a more refined initial vision for the project, which can in turn lead to better results and a more efficient overall process. By taking the time to meet with an expert, we can gain valuable insights and perspectives that can help us to avoid common pitfalls and achieve our goals more effectively.

7 References

[1] Chen, Z., Zhang, Y., Rosenberg, A., Ramabhadran, B., Wang, G., Moreno, P. (2021). Injecting text in self-supervised speech pretraining. 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). https://doi.org/10.1109/asru51503.2021.9688018

[2] Chen, Z., Zhang, Y., Rosenberg, A., Ramabhadran, B., Moreno, P., Wang, G. (2022). Tts4pretrain 2.0: Advancing the use of text and speech in asr pretraining with consistency and contrastive losses. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp43922.2022.9746475

[3] Wang, G., Rosenberg, A., Chen, Z., Zhang, Y., Ramabhadran, B., Wu, Y., Moreno, P. (2020). Improving speech recognition using consistent predictions on synthesized speech. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp40776.2020.9053831

[4] Rosenberg, A., Zhang, Y., Ramabhadran, B., Jia, Y., Moreno, P., Wu, Y., Wu, Z. (2019). Speech recognition with augmented synthesized speech. 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). https://doi.org/10.1109/asru46091.2019.9003990

8 Student Contributions

Kristin Mullaney: Helped troubleshoot the initial synthesizer with guidance from Michael Picheny. Synthesized mass amounts of audio and prepared training directories to test various model configurations. Ultimately, ran and tested many different model configurations.

Ilias Arvanitakis: In the initial part of the project experimented with the hyperparameters of the model and familiarized with the speech synthesizer. Contributed to the experiments by running models for various train sizes. Focused on writing the initial part of the report and the conclusions.

Alexandre Vives: Contributed to the hyperparameter tuning of the baseline model. Manually identified the best short embedding, created the visualizations of the results and contributed to the report by describing the modeling process.