Listen Anime Subtitle: An LAS-Based Automatic Subtitle Generator

Garrison Hess
Carnegie Mellon University
Pittsburgh, PA 15213
glhess@andrew.cmu.edu

Tomas Hrafn Johannesson
Carnegie Mellon University
Pittsburgh, PA 15213
tjohanne@andrew.cmu.edu

Nathan Riopelle
Carnegie Mellon University
Pittsburgh, PA 15213
nriopell@andrew.cmu.edu

Xiaoyu Sun
Carnegie Mellon University
Pittsburgh, PA 15213
seansun@andrew.cmu.edu

Abstract

In this work, we explore transfer learning and learning representations in end-to-end automatic speech recognition (ASR) models. We apply our methods to the *Kimi no Na wa* dataset, which entails transcribing movie dialogue for subtitles. Our core contributions include (1) transfer learning of a Listen-Attend-Spell (LAS) model from the Wall Street Journal dataset and (2) transfer learning of a large transformer model using the wav2vec 2.0 self-supervised learning representation for .WAV files. Specifically, we achieve an evaluation Levenshtein distance of 15.04 using transfer learning on spectrograms with our LAS model, and a distance of 6.93 using raw audio files with wav2vec 2.0 and a pretrained transformer model.

1 Introduction

Audiovisual media are incredibly pervasive in modern society and exist in many forms including movies, TV shows, and internet videos watched by people around the globe. YouTube alone reports that its users collectively watch over one billion hours of video content every day [3]. Yet despite the pervasiveness of video media, generating text subtitles for all of this content remains a daunting task. There are many benefits to providing subtitles for videos, including increasing audience accessibility and understanding. Traditional subtitle generation however has involved a laborious process of supplying a text script along with the video file and manually aligning speech utterances with text phrases. Despite the benefits of subtitles, a number of content creators cannot or choose not to create subtitles because of the time and budget costs of this manual process which greatly limits accessibility. Our project aimed to remedy this problem by automatically generating subtitles for films and other forms of visual media without the aid of transcripts.

The goal of this project was to generate subtitles for the animated film *Your Name* (Hepburn: *Kimi no Na wa [KNNW]*). Specifically, our task was to generate the dialogue text throughout the movie. Movie subtitle generation is a complex task that requires transforming audio input signals into accurate text. The traditional way to perform speech to text generation is using sequence-to-sequence methods. A commonly used name for this subfield is automatic speech recognition or ASR. The particular ASR model we chose to base our architecture off of is the Listen, Attend, Spell (LAS) attention-based model developed by Chan et al. [2] in 2015. On top of the baseline, we have also added major model enhancements to improve subtitle text accuracy. These enhancements include transfer learning for the weights of our LAS model on 80 hours of the WSJ dataset, applying SpecAugment [10] for audio pre-processing, and implementing the wav2vec 2.0 [1] model for unsupervised pretraining to

Preprint. Under review.

generate learned representations for speech data as a new input to our ASR model. We present both of these enhancements, along with other minor modifications to the LAS model in an exploratory study format directly comparing performance between each improvement and the previous architecture. By integrating all of these enhancements together and detailing each contribution, we hope to show not only the capabilities of our final combined model, but also the learning process we underwent along the way.

2 Literature Review

The core of our project involves Automatic Speech Recognition (ASR), which is a widely studied task. One such study, entitled DeepSpeech[6], focuses on end-to-end speech system is trained on large RNN with multiple GPUs for thousands of hours. Deep Speech is unique in that it trains using raw data that has not been processed for noise filtering or speaker adaptation. In addition, DeepSpeech leverages a bi-directional recurrent layer, including two hidden unit groups for forward and backward recurrence, respectively.

Sequence labeling and prediction is yet another highly relevant area of research for our project. Sak et al.[12] exhibited the strength of Long-Short Term Memory architectures for such tasks, in contexts ranging from hand-writing recognition to phoneme classification. Recurrent Neural Networks (RNNs) were used to train acoustic models for vocabulary speech recognition. The LSTMs used often tend to converge relatively quickly, as well. Their project leverages Asynchronous Stochastic Gradient Descent to speed-up model training. Collectively, they used 1900 hours of speech to train the model. In essence, they showed that projected LSTM specification has an improved performance in comparison to the opposite Deep ANN for giant vocabulary

While traditional RNNs enabled numerous major breakthroughs in speech recognition, recent works have largely focused on transformers and CNN-based architectures. Different combinations of RNN, transformer, and CNN have enabled new state-of-the-art accuracies. Gulati et al. [4] proposed the convolution-augmented transformer for speech recognition, Conformer. Han et al. [5] proposed a novel CNN-RNN-transducer architecture, ContextNet. Both architectures have achieved state-of-the-art performance and enabled further improvements on ASR. Meanwhile, Schneider et al. [13] proposed wav2vec, an unsupervised pre-training on large audio dataset, that has led to improved performance on ASR. One year after the publication of wav2vec model, wav2vec 2.0 was proposed by Alexei et al. [1], which uses a transformer-based architecture and is pre-trained with unsupervised task on large audio dataset.

Audio pre-processing is important for ASR. Denoising is one important aspect, especially in the scope of generating dialogue subtitles from movies. Malek et al. [9] experimented and compared multi-condition training of the acoustic models and both fully-connected and convolutional denoising autoencoders (DAE). Liang et al. [8] proposed invariant-representation-learning. At training time, they sample a noisy counterpart for each training example at each training iteration and apply penalty to improve model robustness on learning correct examples. In addition, audio data augmentation is another important direction to pre-process audio data. Park et al. [10] proposed an augmentation method for audio input called SpecAugment. SpecAugment masks out the audio data along both time dimension and frame feature dimension. This introduces effective noise in the input data and thus improves ASR models' learning ability and generalization ability.

3 Dataset Description

The dataset for this project was the 2016 anime film *Kimi no Na wa*. For training data we used a raw .WAV file of the audio track from the movie and a text transcript with per-utterance timestamps as ground truth. Before providing it as input to our LAS model, we made a pre-processed version of the .WAV file that converted the audio into log mel spectrograms with an approximate frame size of 4.67 ms. The total run time of the film was about 107 minutes, or 6396010 ms, thus we ended up with 1370582 total mel spectrogram frames of data.

For our speech recognition task, we divided up the continuous audio frames into discrete utterances based on the timestamps provided in the transcript file. Thus for the batched input to our model, a batch size of N utterances would correspond to providing N variable length sequences of mel

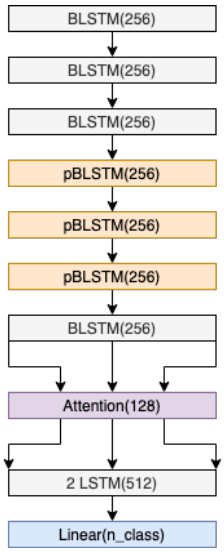

Figure 1: Baseline Architecture

spectogram frames and the ground truth text transcript for each utterance. In the raw transcript of the movie there were 1393 total utterances.

One of the challenges in using this dataset as opposed to more uniform and widely used speech datasets such as WSJ was that there was not a one-to-one mapping between the transcript and audible speech spoken in the movie. Many utterances in the transcript contained written cues for non-speech sounds such as laughing, crying, or gasping. In addition, some lines of the transcript provided a description for what was on screen when there was no audio at all, such as when a sign or note was shown in Japanese and an English translation was provided. In order to account for these mismatches between the data and labels, we implemented a regular expression matching function in our Dataset class to remove non-audible cues from transcript lines or ignore lines entirely if they had no spoken words. After account for ignored lines, there were 1368 usable utterances in the transcript. Beyond checking for non-spoken words, we also used the regex function to correct for other irregularities in the transcript such as unusual characters and punctuation. This step was critical as we planned to implement transfer learning from the WSJ dataset, thus it was necessary that the set of possible output characters be the same in each dataset.

4 Baseline Model and Proposed Enhancements

The task for this project has high similarity with the HW4P2 project as these two projects both focus on translating speech into English text. In HW4P2, our group member Garrison developed the winner model with only 8.42639 Levenshtein distance on the test set. Given that, we believe Garrison's winner model is a good baseline model for our project. The detailed structure of this baseline model is shown in Figure 1. BLSTM are bidirectional LSTM layers with hidden dimension of 256. pBLSTM layers are pyramidal BLSTM layers with hidden dimension of 256 [2]. It should be additionally noted that every LSTM layer in our architecture is followed with a dropout layer. We uniformly set the dropout probability to be 0.4 for all dropout layers. We used the Adam optimizer with weight decay of 0.000005. The learning rate was set to 0.0005 in the first epoch. We used a learning rate scheduler which halved the learning rate every 100 epochs. We used cross entropy loss with no reduction as a loss function.

While having a baseline model directly from HW4P2 was a good starting point, we also wanted to explore how state-of-the-art (SOTA) models perform on generating subtitles for KNNW. Given the time limit and the scope of this final project, we selected wav2vec 2.0 pretrained model as our SOTA model to be explored for this task as it's well pretrained and has been successfully adapted for speech2text tasks [14]. For simplicity, we will use wav2vec to refer to wav2vec 2.0 model in the remaining parts of this report.

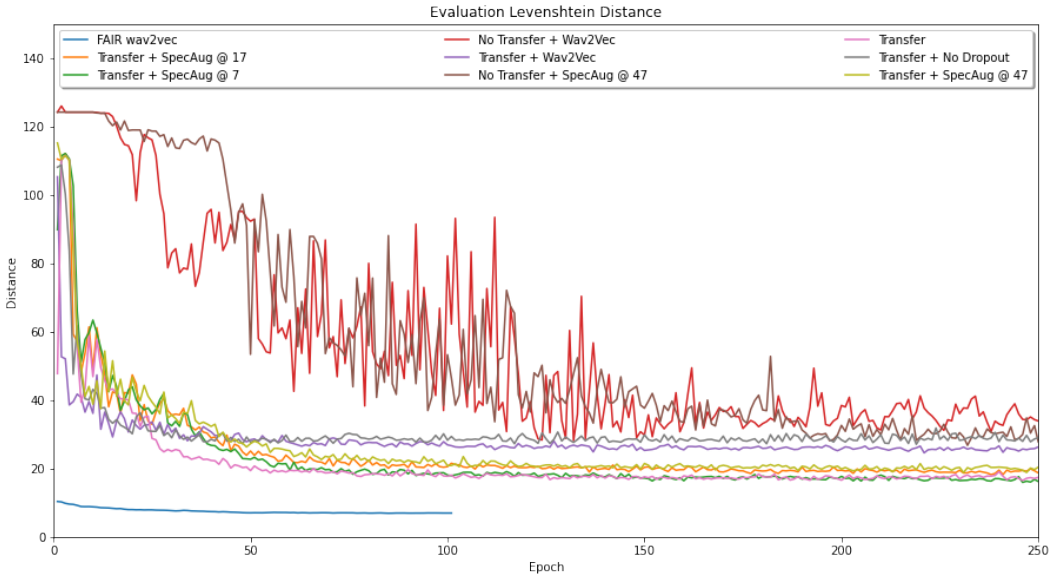

Figure 2: Validation Levenshtein Distance Curves

Model	Best Evaluation Levenshtein Distance
FAIR wav2vec	6.93
Transfer + SpecAug @ 7	15.04
Transfer	15.52
Transfer + SpecAug @ 17	17.59
Transfer + SpecAug @ 47	17.87
Transfer + Wav2Vec	24.59
No Transfer + SpecAug @ 47	26.79
Transfer + No Dropout	27.10
No Transfer + Wav2Vec	28.16

Table 1: Experiment Results Sorted by Best Evaluation Levenshtein Distance

Given achieving closer performance to wav2vec model is our goal, we propose two enhancements to our baseline model. Firstly, data augmentation is an interesting and important enhancement to explore. We propose applying SpecAugment as our data augmentation technique. While SpecAugment is conducted on featurized .WAV input, we realize wav2vec directly processed raw .WAV data and uses its internal feature extractor architecture to automatically extract features. So, as another direction to enhance input quality, we propose using a pretrained wav2vec feature extractor to featurize raw .WAV inputs. The output from feature extractor will then be passed into our baseline architecture.

5 Preliminary Results and Discussion

In order to investigate the effect of each of our proposed enhancements upon the baseline LAS model, we conducted a series of experiments to analyze performance changes. In each experiment we used the previous best model as a control and compared its results directly against an experimental model which incorporated one model enhancement. In total, we ran three experiments focused around enhancing with wav2vec, transfer learning, and SpecAugment. In additional to these, we also ran some minor tests to explore configurations not covered by the main experiments. In each experiment we cover the differences in the model, a comparative plot of performances, and a brief explanation for why we believe we observed the results that we did. Figure 2 summarizes evaluation Levenshtein distances for all configurations we experimented. We also report the best evaluation Levenshtein distances in Table 1 where rows are sorted in ascending order based on best evaluation Levenshtein

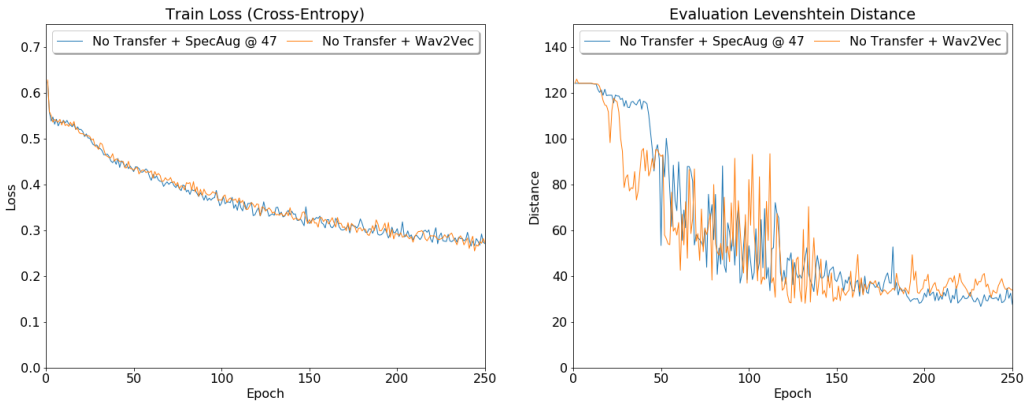

Figure 3: Experiment 1 performance comparison for cross-entropy loss (left) and evaluation Levenshtein distance (right) for the baseline LAS model with and without the wav2vec front end.

distance. In addition, we provide some examples of our generated subtitles from our best LAS transfer learning model and best wav2vec model in Appendix A.

5.1 Experiment 1: wav2vec

As discussed in section 4, we experimented with a wav2vec pretrained model to explore the SOTA model performance on our task. wav2vec's validation Levenshtein distance during fine tuning is shown as the lowest curve on Figure 2, and its best validaiton Levenshtein distance is recorded in Table 1. From Figure 2 and Table 1, it becomes clear that wav2vec outperforms all other configurations we experimented. This shows the transformer architecture is powerful in learning and generalizing speech2text task. This also shows the effectiveness of large-scale training. However, as aforementioned, beating wav2vec performance is not our goal in this project. We only experiment with wav2vec pretrained model to gain a sense of how SOTA model perform on our task.

The highly performant wav2vec model inspired us to combine our basline model with wav2vec components, more specifically, the front-end feature extractor in the wav2vec model. So, the first experiment we ran was investigating the addition of a wav2vec front-end for generating learned representations of raw .WAV audio data to provide as inputs to our model in lieu of the traditional mel spectrogram frames. In this experiment we ran a simple direct comparison between our baseline LAS model using the mel spectrogram input and the same model with the featurized wav2vec vectors substituted as input. The performance results from this experiment are shown in Figure 3.

As can be seen from the plot, wav2vec did not result in a significant evaluation performance improvement, thus it was left out from our final model. We think the reason behind this is the word distribution mismatch between WSJ dataset, which was the dataset that wav2vec model was pretrained on, and our KNNW dataset. This is intuitive as real-life speech data from WSJ dataset can be largely different from the speech data in an anime movie. In addition, we also noticed major disparities in the frequency composition of the two datasets in their respective mel spectrogram formats. We believe being so dissimilar in the frequency domain also contributed to the failure of wav2vec as features extracted from one spectrogram type would not be as applicable to the other.

5.2 Experiment 2: Transfer Learning

Our second experiment was centered around transfer learning. Specifically, we were interested in the idea that performing transfer learning on our baseline model trained on 80 hours of the WSJ dataset could lead to significant improvement in performance on KNNW. As implemented, this experiment was another direct comparison between our baseline LAS model and the same model pretrained on 80 hours of WSJ data where the model weights from the pretraining were used as the initialization weights at the start of the KNNW training task. The performance results from this experiment are shown in Figure 4.

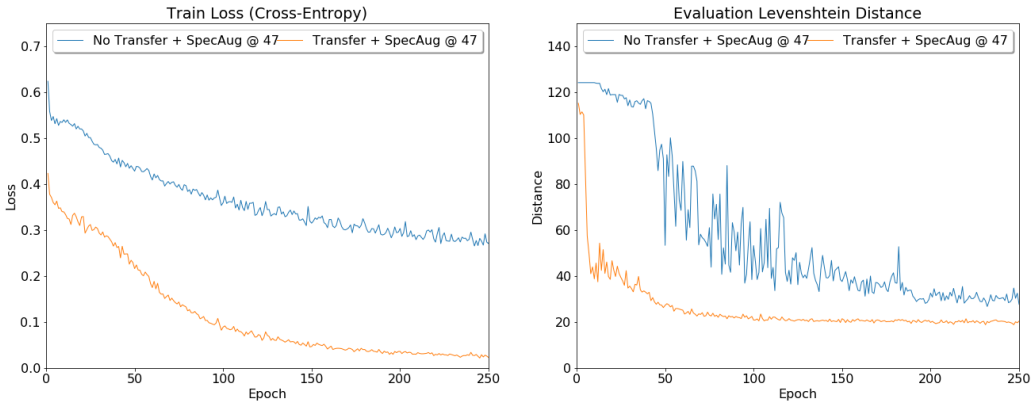

Figure 4: Experiment 2 performance comparison for cross-entropy loss (left) and evaluation Levenshtein distance (right) for the baseline LAS model with and without transfer learning on WSJ for model weights.

The results show using transfer learning resulted in significantly better evaluation performance with a 8.92 reduction in Levenshtein distance. This is an interesting observation as feature extractor pretrained on WSJ dataset didn't help improve performance but an entirely pretrained model on WSJ dataset successfully transfer learned the KNNW dataset. The intuition behind this observation is that while raw audio input distribution can vary from dataset to dataset, all speech share internal similarities, which is the foundation of the success of transfer learning in our experiments.

5.3 Experiment 3: SpecAugment

For our final experiment, we examined the possible benefits of using methods from the SpecAugment paper for audio pre-processing. The two processing operations mentioned in the paper that we implemented in this experiment were time masking and frequency masking. In each one of these operations, a processing function is setup with a max value for the number of indices in our data vector we would like to mask (set to zero). When the function is run, the method will randomly sample a starting index for the mask and sample a uniform distribution from zero to the set max and use this sampled number as the length of mask.

Knowing from our previous experience with the baseline model that SpecAugment does provide an advantage, our primary goal in this experiment was to compare different max mask sizes and determine which achieved the best results. We compared compared four different max mask sizes (0, 7, 17, 47) where the same size was used for both the time and frequency masking functions. The performance results from this experiment are shown in Figure 5.

After running all experiments, we determined the best model to use no wav2vec, transfer learning from WSJ, and SpecAugment with a max mask of 7. This model achieved a Levenshtein distance of 15.04 and is shown as the top performing model in Table 1.

5.4 Additional Tests

Additionally, during our model architecture search, we experimented with several additional configurations that do not perfectly follow these three experimental designs we proposed but also show interesting results. All the results for these additional experiments can be found in Figure 2 and Table 1. Recall we experimented with the wav2vec feature extractor without transfer learning at the beginning, but it did not improve performance possibly because of a mismatch in the word distribution and frequency makeup between datasets. Following on that, we experimented with a transfer learning + wav2vec configuration where we use model pretrained on WSJ dataset and add wav2vec feature extractor on the top of the model. We expected this to fix the distribution mismatch issue but the evaluation Levenshtein distance curve shows only comparable performance with other

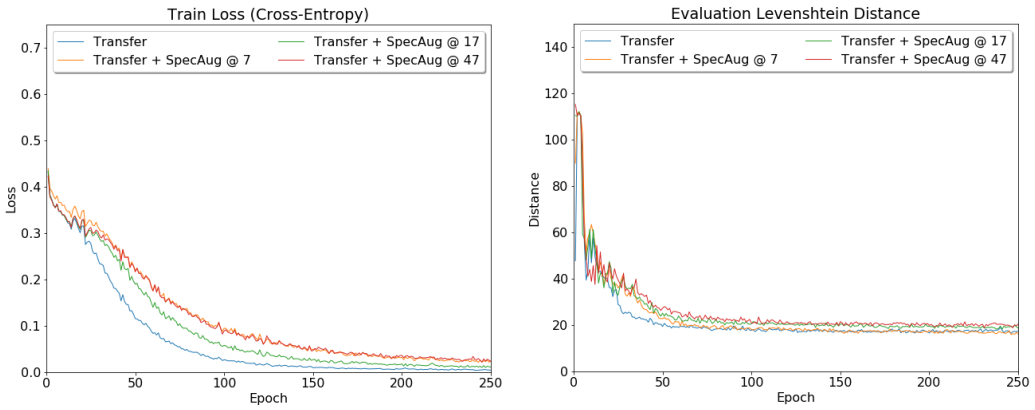

Figure 5: Experiment 3 performance comparison for cross-entropy loss (left) and evaluation Levenshtein distance (right) for the the baseline model plus transfer learning with four different maximum SpecAugment mask sizes. The best evaluation performance was found with a SpecAugment max mask size of 7 (indices).

transfer learning configurations without wav2vec which indicates the wav2vec feature extractor does not really help.

Another additional experiment we tried is a transfer learning + no dropout configuration. This is built on the success of transfer learning in our experiments and we wanted to test the effect of using dropout, which is a common consideration during architecture search. But, again, the result was only comparable with other transfer learning configurations. So we believe adding more dropout does not play signification role in this task.

6 Future Directions

As mentioned in the previous Results section, we noticed major disparities in the numeral scale and preprocessed masks of the two datasets used in this project: KNNW and the WSJ dataset. We believe these disparities were a major factor in preventing us from realizing the benefits of wav2vec. Thus, as a future direction, we propose obtaining the raw audio for each of these datasets and performing our own preprocessing to convert each into log mel spectrogram format. By using the same preprocessing function on both datasets, we hypothesize that the degree of similarity between them would be much higher and enable more successful feature extraction of KNNW data using wav2vec trained on WSJ.

To solve the issue of different word distributions between the two datasets, we believe it would also be prudent to use a larger dataset of anime movies for training. *Kimi no Na wa* has a runtime of only 107 minutes, which after the 80/20 training/eval split, only 85.6 of were used for training. Compared to the hundreds of hours of WSJ data wav2vec was pretrained on, the difference in data size is large. With a larger collection of speech from similar anime movies incorporated into the training dataset, we believe that the disparity would be less large. To improve wav2vec performance specifically, wav2vec could also be pretrained either in-part or exclusively on anime data.

While we already had some intuition from homework 4 that SpecAugment would improve performance, we believe the reason that transfer learning worked while wav2vec did not was that we only used the feature extractor part of wav2vec, which was originally pretrained on WSJ data, thus it could not generalize well to anime data. The differences between the spectral makeup of the two datasets as explained in the previous paragraph explain why generalization was not possible. In transfer learning meanwhile, we were able to train an entire model on WSJ and thereby learn deeper representations of speech data. We believe that had we also used wav2vec on the baseline model using for transfer learning, we could have seen much better results on the Kimi no na wa dataset.

Outside of limitations to our current project, there are also some new directions that we did not have time to explore, but we believe would be worth pursuing in the future. The first of these is automatic speaker identification. When we were processing the text transcript of the movie, we went through a

great amount of effort to remove some of the speaker identifiers that were provided in the transcript as they were non-audio text. This gave us the idea that it would interesting to develop a system that could classify the identity of the current speaker, at least for the main characters such as Mitsuha and Taki. Although there are already many well established systems for identifying speakers from only the audio channel [7], we believe this would also be an interesting opportunity to integrate the visual data from the movie and use a multi-modal model. The Look, Listen, and Learn system developed by Ren et al. [11], could provide a good baseline model for solving this problem.

Once a multi-modal model has been introduced there are many other avenues that could be explored. One of these, which was an original stretch goal for this project, is to use visual data to not just identify speakers, but also aid in the speech recognition pipeline. Another example of this could be using vision to read Japanese signs and writing on the screen in order to fill in gaps in the transcript that audio data could not provide. Although these vision-based methods are exciting, one aspect we are curious and cautious of is how well CNN models trained live-action video could generalize to animated video such as *Kimi no Na wa*.

7 Conclusion

In this study, we applied various contemporary ASR models and methods to generate subtitles on a very small dataset using transfer learning. We presented an ablation study showing which methods were most effective, and analyze these results in a learning representation context. We found that transfer learning from the WSJ data to the KNNW data was far more effective than training from scratch. However, the significant difference in spectrogram representations from WSJ (40-dimensional) to KNNW (129-dimensional) hampered the transfer learning process. Accordingly, we suggested that far greater performance can be achieved by conducting transfer learning on representations generated by applying wav2vec 2.0's feature extractor to the raw audio files. This process avoids discrepancies in spectrogram generation and reflects the end-to-end aspect of our ASR models. Ultimately, we found that transfer learning with end-to-end ASR models is highly sensitive to different learning representations, and this issue can be averted by using .WAV audio files directly in true end-to-end fashion.

References

[1] Alexei Baevski, Henry Zhou, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. 06 2020.

[2] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*, 2016. URL `http://williamchan.ca/papers/wchan-icassp-2016.pdf`.

[3] Cristos Goodrow. You know what's cool? a billion hours. URL `https://blog.youtube/news-and-events/you-know-whats-cool-billion-hours/`.

[4] Anmol Gulati, Chung-Cheng Chiu, James Qin, Jiahui Yu, Niki Parmar, Ruoming Pang, Shibo Wang, Wei Han, Yonghui Wu, Yu Zhang, and Zhengdong Zhang, editors. *Conformer: Convolution-augmented Transformer for Speech Recognition*, 2020.

[5] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context. *arXiv e-prints*, art. arXiv:2005.03191, May 2020.

[6] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014. URL `http://arxiv.org/abs/1412.5567`. cite arxiv:1412.5567.

[7] Honglak Lee, Yan Largman, Pham Peter, and Andrew Y. Ng.

[8] D. Liang, Z. Huang, and Z. C. Lipton. Learning noise-invariant representations for robust speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 56–63, 2018. doi: 10.1109/SLT.2018.8639575.

[9] J. Malek, J. Zdansky, and P. Cerva. Robust automatic recognition of speech with background music. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5210–5214, 2017. doi: 10.1109/ICASSP.2017.7953150.

[10] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. Specaugment: A simple augmentation method for automatic speech recognition. In *INTERSPEECH*, 2019.

[11] Jimmy Ren, Tongtao Hu, Yu-wing Tai, Chuan Wang, Li Xu, Wenxiu Sun, and Qiong Yan.

[12] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. *arXiv e-prints*, art. arXiv:1402.1128, February 2014.

[13] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469, 2019. doi: 10.21437/Interspeech.2019-1873. URL `http://dx.doi.org/10.21437/Interspeech.2019-1873`.

[14] Yu Zhang, James Qin, Daniel Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc Le, and Yonghui Wu. Pushing the limits of semi-supervised learning for automatic speech recognition, 10 2020.

A Generated Subtitle Samples

Subtitle estimate	Ground truth label
all those dreams of the miyamizu people at	all those dreams that the miyamizu people had
you were wrote youve gotta be	you were wrong you gotta be
yeah ats itomori teach me bark	yeah thats itomori takes me back
well mitsuha	half of mitsuha
whach you just given a rest algothes ago monsels	would you just give it a rest with all this occult nonsense

Generated Subtitle Samples from Best LAS Model

Subtitle estimate	Ground truth label
could she call musake	kuchikamizake
ya of course they do	yeah of course they do
i'd never do thatandin front of everybody embarrassing right	ugh i'd never do that and in front of everybody embarrassing right
who cares at a few of the kits from mew school saw that	who cares if a few of the kids from your school saw that
oh and then you could call it shrine mane and secket you'll make bot lo	oh and then you can call it shrine maiden's sake you'll make buttloads

Predicted Subtitle Samples from Best wav2vec 2.0 Model