Wav2Vec2 Chinese Speech to English Text Exploration

Daniel Cheng, Echo Zhang, Simon Zheng, Yuqian Zhuang

1. Abstract

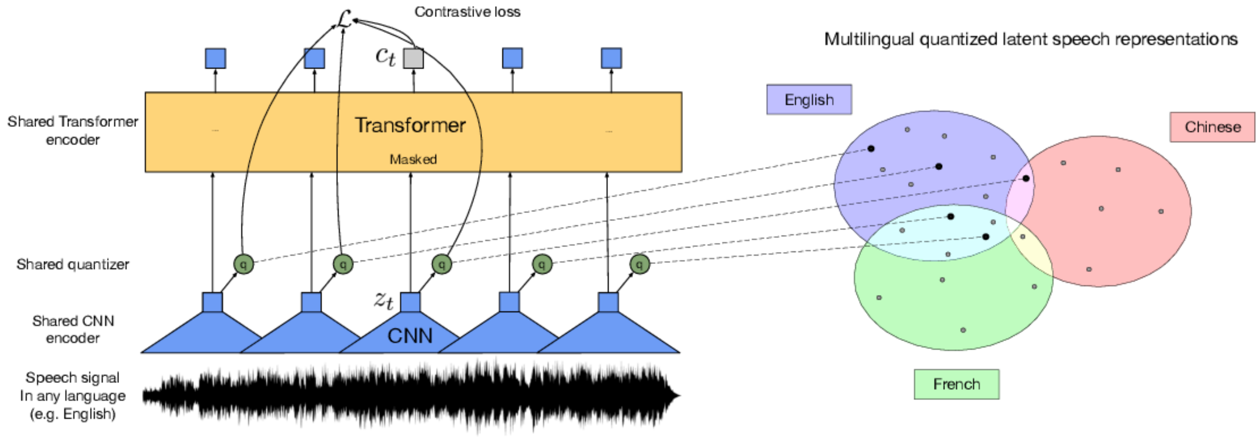
Recent models in Chinese Mandarin automatic speech recognition (ASR) and Mandarin-English translation have been increasingly efficient and accurate. The choice of models for either ASR and machine translation has expanded dramatically, with many different pre-trained models and different architectures. In addition, there has been more emphasis on using Wav2Vec for ASR, and neural networks for Chinese-English translation.

Initial ASR systems typically choose context-dependent states, or context-dependent phonemes as their modeling units (Zhou et al., 2018). However, different modeling units have been explored recently, such as sequence-to-sequence attention-based models. These models integrate a language model, pronunciation, and acoustic signals into a single neural network. For Mandarin Chinese, more recently literature is focused on seq-to-seq attention-based models, with the Transformer. We explore a specific Wav2Vec2 model in this paper, and see if we can replicate and improve the current BLEU/CER accuracy from various other papers.

2. Introduction

Although Chinese is considered a low-resource language, recent models from Facebook's Wav2vec2 have proven to work well on such languages. Wav2Vec2 is a self-supervised framework for speech learning, that utilizes raw-audio rather than transcribed audio. In addition, although neural networks typically require a large amount of annotated data, Wav2Vec2 is able to thrive with much less annotated data.

Wav2Vec2 mainly consists of a CNN encoder, and a Transformer (Fan et al., 2021). The CNN transfers raw waveforms to speech representations. Then, these representations are masked and converted into context representations, and fed into the Transformer. The astonishing part is that after pre-training, Conneau et al. applied this model to another ultra-low resource language, with only ten minutes of labeled data from Librispeech (2020). They were able to get a word error rate of 18.3%, which shows that phoneme-related information is preserved during the pre-training of the w2v-encoder, and the downstream task can benefit greatly (Fan et al., 2021).



XLSR-Wav2Vec2 Model

Currently, the ASR from Google on our phones is great, but once it's translated into English, a lot of information is mixed. In hopes of exploring the complications of Chinese ASR, we recreated a pipeline by combining Wav2vec2's ASR ability, with openNMT's machine translation ability. Through this process, we gained insights into where the translation errors arise. We wanted to understand if it's an ASR problem, machine translation problem, or a language-related problem. In addition, we were interested in learning more about the Wav2Vec2 architecture from Facebook. The specific model we used was the "XLSR Wav2Vec2 Large 53 - Chinese" model, which is a model trained on 53 similar languages over 56k hours (Conneau et al., 2020).

3. Related Work

Previous models have achieved promising results from using various models on Chinese ASR. Our first paper examines a speech to text system with FairSeq on the CoVoST dataset. They achieved a BLEU score of approximately 6, when translating Chinese speech to English text (C. Wang et al., 2020). This is why we also used the CoVoST dataset to assess our pipeline, since this paper provided a BLEU score reference.

Next, Zhou et al. explore the complications of using seq-to-seq attention based models on Mandarin, with various datasets from HKUST (2018). Among the five modeling units they tested, character based models performed the best, with a CER of 26.64% on their datasets, which were an improvement compared to the CTC-attention based encoder-decoder network. Thus, for our model, we also implemented a character based model, since it achieved the highest performance. In addition, we assessed our ASR model on CER, which was what these researchers used.

The most recent paper published regarding Wav2Vec2 helped us the most, since it explored the exact XLSR-53 model that we chose. The researchers performed many different tests, and used the CommonVoice dataset as their training set. Next, they found that this specific model performed significantly better than other XLSR-10 models, noting a decrease in phoneme error rates when translating spoken Chinese to Chinese text. Thus, we used this dataset and this model, to see if we could achieve comparable results(Conneau et al., 2020).

For machine translation, a paper by Yao et al. explained the complications of using neural networks in translating Chinese to English text, with the best BLEU score of 33.52 (2015). In addition, this paper explores depth-gated recurrent neural networks, which connects memory cells to adjacent layers in the network. Overall, it explained that neural networks have lots of room for improvement, but neural networks are already performing much better than before.

With this recent research in mind, we learned how effective Wav2Vec2 was on Chinese, and we wanted to learn more. Through building our pipeline and using similar datasets from various papers, we are able to see if our results are reasonable. In addition, no researcher has explored the conversion from Chinese spoken language to English text on our datasets. So, the exploration of a zh-en pipeline is novel, and there was a lot to learn even though the ASR Wav2Vec2 model has many published articles on it. In addition, the ability to string together Wav2Vec2 and openNMT is extremely useful if it proves to be functional and accurate.

4. Datasets

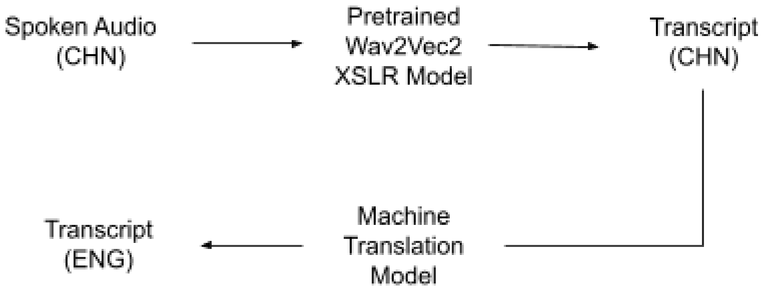
For this project, two datasets were used from CommonVoice and OPUS. The CommonVoice dataset was used for our ASR training, and the OPUS data was used for our Machine Learning training.

The .mp3 files in CommonVoice contained over 78 hours of speech files, which were from 3000+ unique Chinese speakers, from various regions in China. 38% of the participants of the participants were around 19 to 28 years old, which was the majority group. We separated the CommonVoice clips into training dataset and test dataset. The training dataset contained 18,451 clips, and the test dataset contained 8,067 clips. The domain of the content in CommonVoice data is very diverse, including topics like history, politics, and movies. In order to make the clips readable, we utilized Pyaudio packages to convert speech clips into vectors.

The data we used from OPUS is the Chinese TedTalk text from 2013, which contained 119,999 sentences in the training dataset and 34,580 sentences in the training dataset. The raw data was separated into two different files, Chinese and English, and their indexes in the files were perfectly aligned, which helped us save time on preprocessing.

5. Methods

Our full model pipeline includes two parts: Chinese spoken texts recognition and Machine translation model that translate Chinese transcript to English Transcript



For speech recognition, we used the pre-trained Wav2Vec2 XSLR model and followed the tutorial to fine tune the model using zh-CN dataset from Common Voice. The hyperparameters were set mostly as provided in the tutorial, except we changed \`per_device_train_batch_size=2\` and \`gradient_accumulation_steps=16\` for smoothing the data processing process.

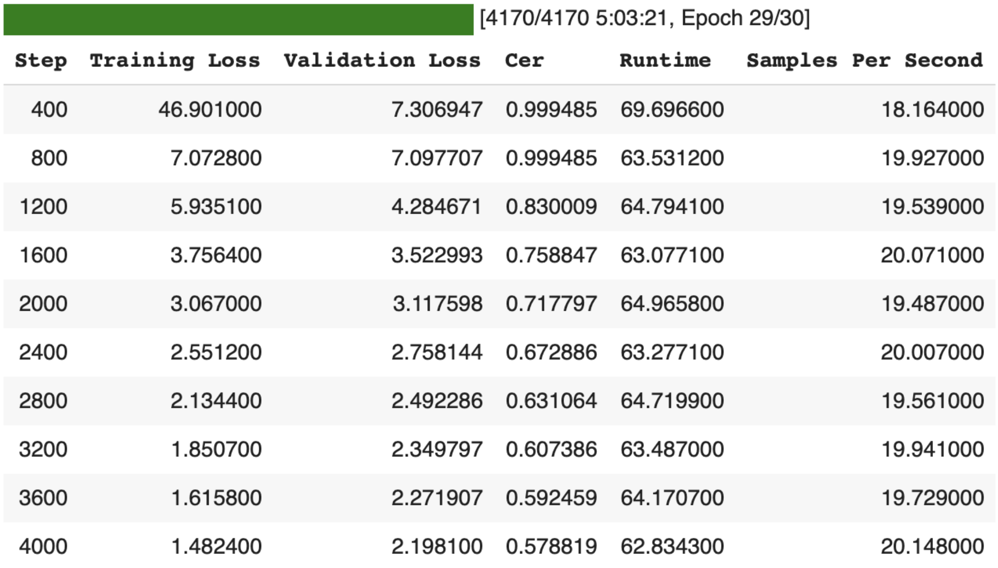
For the machine translation process, in order to make our model train on sufficient amounts of data, we used aligned text files for Chinese and English in the TedTalk OPUS dataset. Before applying any neural network model on our dataset, we needed to tokenize our sentences first. We used Spacy tokenizer for English, and Jieba tokenizer for Chinese. While tokenizing our sentences, we separated the sentences into 70% train and 30% dev files. We then used OpenNMT for the machine translation architecture, and it trained quite well. The architecture mainly consisted of a RNN encoder and RNN decoder, global_attention: mlp, optimization: adam, and dropout rate of 0.2. To confirm that it was working correctly, we calculated the BLEU score on the dev set, achieving a BLEU score of 11.63.

6. Experiments

For the ASR model, we initially used a character level model, with the purpose of replicating the work done previously in either recent literature, or HuggingFace: fine-tuning the model with Chinese datasets. The baseline on HuggingFace was 20.9%, measured using CER (character error rate). However, we realized that the vocab size for Chinese is very large (4500+ characters in our dataset) compared to other languages like English (26 characters). Combined with the different length of speech audios, this made the input of our model unexpectedly large and resulted in a "CUDA Out of Memory" error (shortage of GPU memory). Therefore, we tried to limit the length of each train clip and adjust the hyperparameters: \`per_device_train_batch_size\` and \`gradient_accumulation_steps\`. We finally trained the model on about 25% of the original training data (around 4500 clips), which contains clips under 4 seconds only with a batch size of 2 and gradient accumulation step of 16.

For the machine learning part, we used RNN encoder/decoder as our model. We initially tried different hyperparameters on the model. For example, we have tried different dropout rates and different train/test splits. We ended up choosing a lower dropout rate, because it seemed to achieve the lower perplexity, and likely meant that our model would generalize better and not overfit. We wanted to lower the amount of overfitting, because we knew that the ASR would be trained on a different dataset, which is a different domain. In addition, we tried a 90/10 split, and decided to use a 70/30 split, to also prevent overfitting. Lastly, we chose Adam as the optimizer for our model because other github projects had also used Adam on Chinese translation.

7. Results



[4170/4170 5:03:21, Epoch 29/30]

Step	Training Loss	Validation Loss	Cer	Runtime	Samples Per Second
400	46.901000	7.306947	0.999485	69.696600	18.164000
800	7.072800	7.097707	0.999485	63.531200	19.927000
1200	5.935100	4.284671	0.830009	64.794100	19.539000
1600	3.756400	3.522993	0.758847	63.077100	20.071000
2000	3.067000	3.117598	0.717797	64.965800	19.487000
2400	2.551200	2.758144	0.672886	63.277100	20.007000
2800	2.134400	2.492286	0.631064	64.719900	19.561000
3200	1.850700	2.349797	0.607386	63.487000	19.941000
3600	1.615800	2.271907	0.592459	64.170700	19.729000
4000	1.482400	2.198100	0.578819	62.834300	20.148000

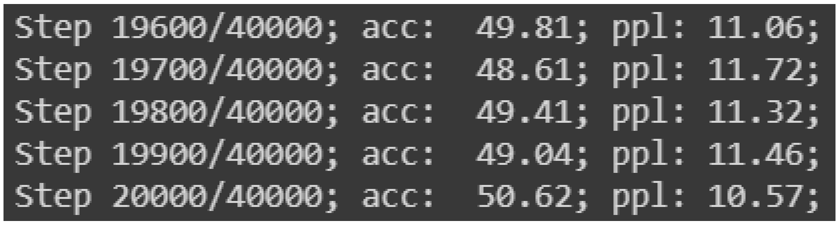
Results from ASR model training in a Google Colab Notebook

With previous settings, we were able to train the model up to 4000 steps with a final train loss of 1.4824 and a CER of 57.88%. This result was more or less expected, as we only trained on 25% of the training data, which is too little for the model to generalize well, especially with many homonyms in the Chinese language. The sample prediction looks reasonable (provided below). However, this could not be fed entirely into the machine translation portion with such CER as it would be a garbage in, garbage out situation.

Prediction	Reference
二十一一年去士	二十一年去世

(Homonyms in the example: “世” and “士”)

Using the OPUS TedTalk dataset in the machine translation process, we achieved a 11.63 BLEU score with approximately 10.5 perplexity and 50.62% accuracy on the dev set. We had no baseline for this specific corpus, but an 11.63 BLEU score on a low resource language was comparable to other similar machine translation research.



```
Step 19600/40000; acc:  49.81; ppl: 11.06;
Step 19700/40000; acc:  48.61; ppl: 11.72;
Step 19800/40000; acc:  49.41; ppl: 11.32;
Step 19900/40000; acc:  49.04; ppl: 11.46;
Step 20000/40000; acc:  50.62; ppl: 10.57;
```

Results from OpenNMT machine translation training in a Google Colab Notebook

8. Conclusion

For ASR, our model is able to predict Chinese texts based on the speech with a CER of 57.88%. Through analyzing the predictions, we realized that one of the reasons that our high CER was a result of homonyms in Chinese. This is actually associated with a challenge that we faced during training: Chinese has too many characters and thus our vocabulary size is too large to train on. This solution definitely hurts the performance of our model, but we had to compromise to complete the training process. This computing shortage also made the whole process more difficult. During the preprocessing and training, we kept running into many `CUDA out of memory` errors and RAM crashes from time to time. If we’re given more time, we would try the Pinyin (romanization of Chinese characters) approach, which converts characters to Pinyin first for training, then converts them back for translation. In this way, we can limit the vocab size down to 31(26 letters+5 tones represented by numbers from 1 to 5) without reducing the amount of data and avoiding the problem caused by homonyms. This would likely achieve a higher CER, and also allow us to use more than 25% of the dataset.

This way, we would have a complete model and be able to assess the entire pipeline on the CoVoST dataset, which was our initial goal. However, due to the poor CER in the ASR model, we were not confident enough to feed the ASR text into our machine translation model. In addition, the machine translation model BLEU score was much lower than typical high-resource languages, which would have created another bottleneck. Overall, Chinese ASR was much more difficult than we expected, even with a newly released pretrained Wav2Vec2 model. Ultimately, we learned a lot from preprocessing, loading, and training an ASR model, and we hope to learn more about Chinese ASR in the future, with more computing and storage power.

References

Baevski, Alexei, et al. *Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. , 2 Oct. 2020.

colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/Fine_Tune_XLSR_Wav2Vec2_on_Turkish_ASR_with_%F0%9F%A4%97_Transformers.ipynb. Accessed 20 Apr. 2021.

Conneau, Alexis, et al. *UNSUPERVISED CROSS-LINGUAL REPRESENTATION LEARNING for SPEECH RECOGNITION*. , 15 Dec. 2020.

Ctl/Wav2vec2-Large-Xlsr-Cantonese · Hugging Face." Huggingface.co, huggingface.co/ctl/wav2vec2-large-xlsr-cantonese. Accessed 20 Apr. 2021. "Fine-Tune XLSR-Wav2Vec2 on Turkish ASR with Transformers.ipynb." Colab.research.google.com, 11 Dec. 2020

Fan, Zhiyun, et al. *EXPLORING WAV2VEC 2.0 on SPEAKER VERIFICATION and LANGUAGE IDENTIFICATION*. , 14 Jan. 2021.

Jia, Bingjing, et al. "Enhanced Character Embedding for Chinese Named Entity Recognition." *Measurement and Control*, vol. 53, no. 9-10, 21 Sept. 2020, pp. 1669–1681, 10.1177/0020294020952456. Accessed 20 Apr. 2021.

Wang, Changhan, et al. *FAIRSEQ S2T: Fast Speech-To-Text Modeling with FAIRSEQ*. , 2020.

Wang, Hongfei, et al. *Chinese Grammatical Correction Using BERT-Based Pre-Trained Model*. , 2020.

Wikipedia Contributors. "Speech Recognition." Wikipedia, Wikimedia Foundation, 29 July 2

Yao, Kaisheng & Cohn, Trevor & Vylomova, Katerina & Duh, Kevin & Dyer, Chris. (2015). Depth-Gated Recurrent Neural Networks.

Zhou, Shiyu, et al. *A Comparison of Modeling Units in Sequence-To-Sequence Speech Recognition with the Transformer on Mandarin Chinese*. , 18 May 2018.

"Huggingface/Datasets." GitHub, 20 Apr. 2021, github.com/huggingface/datasets. Accessed 20 Apr. 2021.

"Librosa — Librosa 0.8.0 Documentation." Librosa.org, librosa.org/doc/latest/index.html.

"README.md · Ydshieh/Wav2vec2-Large-Xlsr-53-Chinese-Zh-Cn-Gpt at Main."Huggingface.co, huggingface.co/ydshieh/wav2vec2-large-xlsr-53-chinese-zh-cn-gpt/blob/main/README.md. Accessed 20 Apr. 2021.

“Wav2vec 2.0: Learning the Structure of Speech from Raw Audio.” *Ai.facebook.com*, ai.facebook.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/. Accessed 20 Apr. 2021.