ELEC-E5510 – Speech Recognition

Group 15: Robust Speech Recognition

Members

An Tran, 487555

Nguyen Xuan Binh, 887999

Date: 15/12/2023

Abstract

Robust speech recognition remains a critical area in the field of automatic speech recognition (ASR), essential for ensuring accurate speech-to-text conversion across various environments and speaker conditions. This report examines the robustness of methodologies, such as time-delayed neural network (T-DNN) and Wav2vec 2.0, by analyzing their performance in characteristics such as background noise and speaker differences, particularly between male and female voices. The study aims to provide insights into the robustness of these models, highlighting the capabilities and limitations in handling real-world speech recognition challenges.

Table of contents

1. Introduction

Robust speech recognition is a critical component of automatic speech recognition (ASR) systems, characterized by the system's ability to accurately transcribe spoken language under a wide spectrum of challenging and diverse conditions. It encompasses the system's capacity to maintain high accuracy and reliability even in adverse or unexpected environments, where audio quality and speaking conditions may deviate significantly from ideal laboratory settings [1].

In our project, we focus on evaluating the robustness of pre-trained speech recognition models under varied conditions. The core challenge addressed here is the ability of these systems to maintain high accuracy despite encountering obstacles such as background noise, speaker variations, and varying environmental conditions. To assess the robustness, we compare the performance of several pre-trained models on a specific dataset, which is further divided into different categories such as noise distortions and speaker variations.

2. Literature study

2.1. Challenges in robust speech recognition

In automatic speech recognition, sound can be characterized by a wide range of factors, making the waveforms extremely diverse despite sharing the same transcription [2]. The first variation is related to noise and acoustics. This challenge pertains to the recognition of speech in the presence of various acoustic disturbances, such as background noise, echoes, reverberations, and fluctuations in speaking styles [2]. Another source of variations comes from diverse environmental conditions, such as outdoor settings, crowded places, or varying weather conditions [3]; or variations introduced by different recording devices and communication channels such as smartphones [2]

In addition, robust speech recognition must account for variations introduced by different speakers, including those with distinct accents, dialects, and speaking rates. Adaptation techniques are often employed to mitigate speaker-related variations [4].

Variations in vocabulary and language can also pose challenges for robust speech recognition. In multilingual and domain-specific applications, the recognition system must be robust to variations in languages, dialects, and specialized vocabularies [5]. Furthermore, robust ASR systems should account for words that do not exist within the system's standard vocabulary [2].

2.2. Methods in robust speech recognition

Early speech recognition systems, largely based on Hidden Markov Models (HMMs), struggled in non-ideal environments. The shift towards Deep Neural Networks (DNNs), has provided significant improvements in robustness [9]. Recent developments have seen the incorporation of advanced neural network architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which offer improved feature extraction and temporal dynamics modeling, respectively. Recently, end-to-end models like Connectionist Temporal Classification and attention-based models has simplified the speech recognition pipeline [10].

Regarding noise reduction and audio enhancement, traditional methods like spectral subtraction and Wiener filtering have been complemented by more advanced techniques. Deep denoising autoencoders, for example, have emerged as a powerful tool for extracting clean speech signals from noisy inputs [11]. There have been methods that process noise-robust features directly from raw audio such as Wav2Vec [12] and Whisper [7].

Addressing speaker variability is another critical aspect of robust ASR. Techniques such as speaker adaptation and normalization have been employed to mitigate differences in voice characteristics. The introduction of speaker embeddings, particularly those extracted using models like i-vectors, has significantly bolstered the ability to handle speaker variations [13].

Language variability remains a challenge, especially in multilingual contexts. Here, approaches like multi-lingual training and transfer learning have shown promising results, enabling models to adapt to different languages and dialects more effectively [14].

Overall, the methodologies in robust ASR have evolved significantly, moving from basic model-driven approaches to advanced, data-driven deep learning techniques. This evolution continues to improve the adaptability and accuracy of speech recognition systems in a variety of real-world applications.

3. ASR models

3.1. TDNN-F (Time Delay Neural Network with Factored parameters)

Time Delay Neural Networks (TDNNs) or one-dimensional Convolutional Neural Networks (1-d CNNs), are an efficient neural network architecture for speech recognition [17]

However, in TDNN-F, there is an effective way to train networks with parameter matrices represented as the product of two or more smaller matrices, with all but one of the factors constrained to be semi-orthogonal [17]. By giving a factorized TDNN (TDNN-F), and applying several other improvements such as skip connections and a dropout mask that is shared across time, the results from TDNN-F are often better than previous TDNN-LSTM and BLSTM results, while being much faster to decode [17]

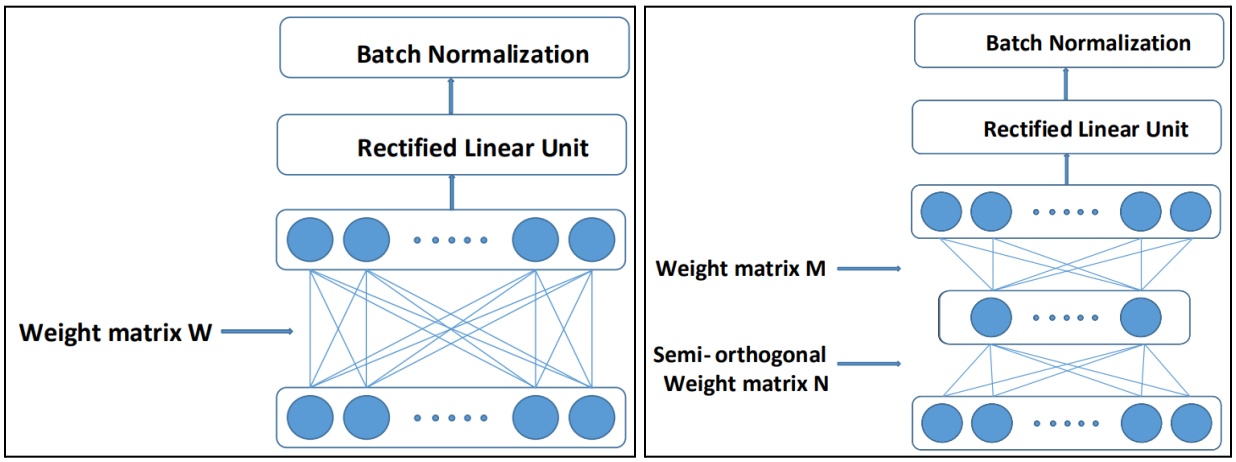


Figure 1: One standard feed-forward layer V.S Factorized layer with semi-orthogonal constraint

In the factorized layer with a semi-orthogonal constraint, initially there is batch normalization and a ReLU activation function. Then, the weight matrix M from the standard layer is factorized into two matrices. The factorization is done such that one of the matrices (N) is constrained to be semi-orthogonal. This means that N is close to an orthogonal matrix but not necessarily square, and when multiplied by its transpose (NN^T), it approximates the identity matrix [17]

The main purpose of factorization is to reduce the number of parameters in the model, potentially reducing overfitting and improving generalization. Additionally, the semi-orthogonal constraint can lead to more stable training because it encourages the weight matrix to have properties like an orthogonal matrix, which can prevent exploding or vanishing gradients.

Finally, even with a reduced number of parameters due to factorization, the model aims to maintain or even improve its performance compared to traditional feed-forward layers, likely due to more efficient learning of the important features in the data.

3.2. Wav2vec 2.0 and n-gram language model

Wav2Vec 2.0 [12] is an acoustic model designed by Facebook AI that leverages self-supervised learning to directly process raw audio waveforms. The model's architecture consists of a convolutional feature encoder followed by a Transformer-based context network, which together extract and contextualize latent speech representations. This approach allows Wav2Vec 2.0 to capture acoustic features with minimal reliance on labeled data, making it highly adaptable to diverse speech datasets.

Wav2Vec 2.0 + n-gram language model represents an integration of acoustic and linguistic modeling. The n-gram language model is added on top of the acoustic model and decodes a sequence of words based on their probabilities derived from large text corpora. This enhancement is particularly beneficial in providing contextual cues that help resolve ambiguities in the acoustic signal, leading to more accurate speech recognition outcomes. The combined model aims to combine the self-supervised learning capabilities of Wav2Vec 2.0 with the probabilistic linguistic predictions of the n-gram model to improve the robustness of ASR system.

4. Dataset

LibriSpeech is a widely used dataset in the field of robust speech recognition. It is a large-scale collection of English speeches derived from audiobooks. The dataset is designed to be diverse, containing recordings from a broad range of speakers with varied accents, ages, and genders, which impose challenges for robust speech recognition tasks [6]. Samples of Librispeech dataset are shown in the Appendix section.

In our project, we assess the robustness of pre-trained speech recognition models under two critical criteria: noise robustness and speaker's gender robustness.

4.1. Robustness to noise assessment

To assess the model's performance in noise conditions, we utilized two distinct subsets of the LibriSpeech dataset: the 'test-clean' and 'test-other'. The 'test-clean' subset consists of high-quality, clean audio recordings that serve as a benchmark for the model's performance

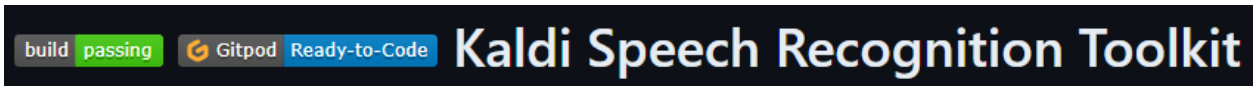
under ideal conditions. In contrast, the 'test-other' subset comprises recordings with a higher presence of background noise and acoustic distortions, presenting a more challenging scenario for our model. By comparing performance across these two datasets, we can effectively measure how noise impacts the model's accuracy.

4.2. Robustness to speaker's gender assessment

We segmented the LibriSpeech 'test-clean' and 'test-other' datasets based on gender, creating male and female subsets. This allows us to analyze the model's accuracy in recognizing speech from different genders and identifies any potential biases or weaknesses in the model's ability to process male versus female speech patterns.

5. Software and experiments

5.1 The KALDI SOFTWARE TOOLKIT



Kaldi is a state-of-the-art automatic speech recognition (ASR) C++ toolkit, containing various algorithms currently used in ASR systems [15]. It also contains recipes for training the acoustic models on commonly used speech corpus such as LibriSpeech, Wall Street Journal (WSJ), Chime, TIMIT, and more. These recipes can also serve as a template for training acoustic models on our own speech data.

Acoustic models are necessary not only for ASR, but also for forced alignment, a technique used to align phonetic transcriptions with the corresponding speech audio, forcing the alignment of the audio with the text at the phoneme level. Kaldi provides tremendous flexibility and power in training our own acoustic models and forced alignment system. [15]

Kaldi's official website: https://kaldi-asr.org

Kaldi's official project repository: https://github.com/kaldi-asr/kaldi

5.2 Workflow of training and testing TDNN-F model in Kaldi

In the recipe file run.sh of LibriSpeech, there are 20 steps. However, there are only 10 distinct stages for training the model from scratch until model validation. The stages are as follows

- Stage 1 downloads and formats data. This stage downloads the LibriSpeech data and converts the downloaded data into Kaldi's standard format for data directories
- Stage 2 prepares the phonetic dictionary and language model data. It constructs constant ARPA format language models for 3-gram and 4-gram models.
- Stage 3 extracts feature and computes Mel Frequency Cepstral Coefficients (MFCCs), which are a feature representation of the audio data. Then, it calculates Cepstral Mean and Variance Normalization (CMVN) statistics from the MFCC.
- Stage 4 trains a monophone model and aligns the data using the monophone model, then trains a triphone model using delta and delta-delta features

- Stage 5 force aligns data with the previous triphone model and trains a new model using Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT). Forced alignment (FA) produces the bidirectional mapping between the given text and speech sequences [18]. FA is widely adopted in speech processing tasks because the bidirectional mapping information helps recognize fine-grained patterns of speakers [18]
- Stage 6 uses Speaker Adaptive Training (SAT) by using the LDA+MLLT model to train a more robust model that accounts for speaker variability.
- Stage 7 computes pronunciation and silence probabilities from training data and uses these to refine the language and phonetic models.
- Stage 8 extracts i-vector. In the feature extraction phase, low-dimensional vectors are extracted using Total Variability (TV) algorithm [19]. It relies on the task-independent redundant information contained in underlying acoustic features to represent the difference between Gaussian mean supervectors of each speech segment [19]. When used with TDNNs, i-vectors enhance the model's ability to recognize speech patterns by incorporating speaker-specific information, which is necessary in environments with multiple speakers or varying acoustic conditions.
- Stage 9 performs decoding on the LibriSpeech test dataset.
- Stage 10 computes the WER score from the decoded sentences.

Steps 4-7 are not necessary to run when we use a pretrained TDNN-F model coupled with the small trigram language model and an i-vector extractor. To use the pretrained model, we extracts MFCC and CMVN features, convert it to i-vectors, then used the acoustic and language model to decode and finally compute the WER score.

Visualization of the pipeline for using the pretrained model

data preparation => MFCC features => i vector extraction => decoding => WER scoring

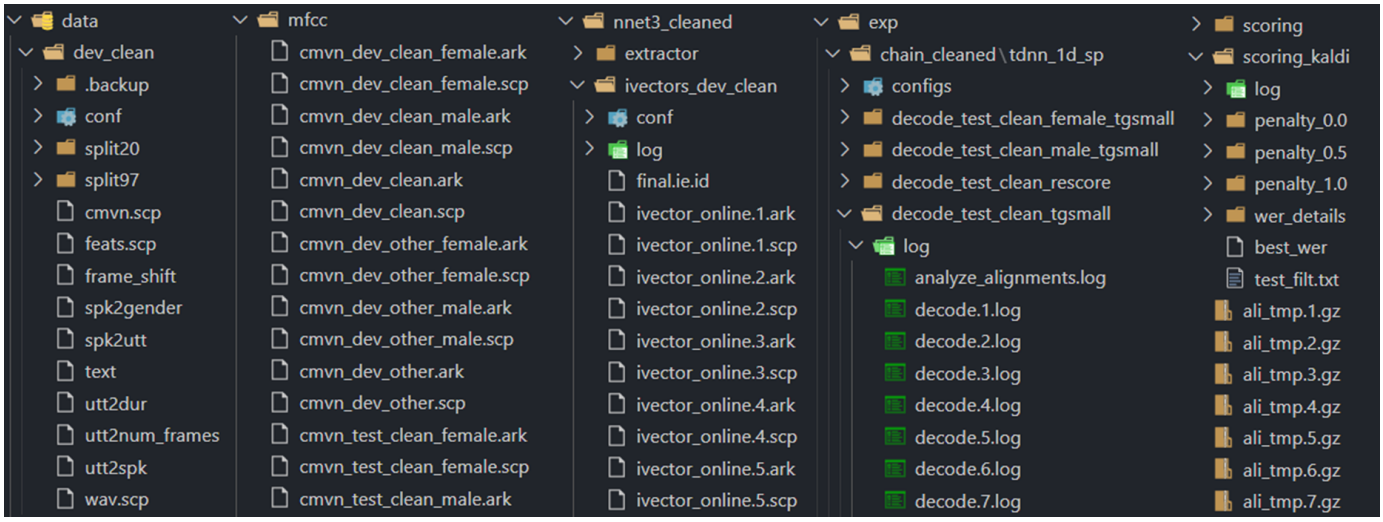


Figure 2: Pipeline for running and testing the pretrained TNDD-F acoustic model

5.3. Workflow of testing Wav2Vec 2.0

Wav2Vec 2.0 is designed to work directly with raw audio waveforms. Therefore, in our experiment, we directly fed the raw waveform audio files into the model. The dataset has been imported into the PyTorch Dataset structure, with audio files being processed using the torchaudio library to convert them into PyTorch tensors, thereby more suitable for GPU processing operations.

We then processed the audio data through a pre-trained Wav2Vec 2.0 on Librispeech to generate latent speech representations. Subsequently, these representations were fed into the n-gram language model to produce the final transcriptions. The pre-trained model is available in Hugging Face under the name `facebook/wav2vec2-base-960h`.

The link to the pretrained model: https://huggingface.co/facebook/wav2vec2-base-960h

For the language model component, we utilized the n-gram model provided with the LibriSpeech dataset. This statistical model is trained on the transcripts from the same corpus. The n-gram model is designed to predict the probability of word sequences using beam search.

6. Results

To assess the performance of the models, Word Error Rate (WER) is used. It compares the model-generated transcription against a reference or ground truth transcription by counting the number of substitutions, insertions, and deletions that the model makes to arrive at the correct sequence of words [16].

WER is defined by the formula: $WER = \frac{N}{S+D+I}$, where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the number of words in the reference [16].

Table 1 shows the WER for each model on LibriSpeech test-clean and test-other datasets. The Wav2Vec 2.0 + 4-gram model leads in performance, with significantly lower WERs in both the test-clean (2.60%) and test-other (6.47%) datasets compared to the other two models.

WER	test-clean average	test-other average
TDNN-F tgsmall	5.28%	12.52%
Wav2vec 2	3.39%	8.57%
Wav2vec 2 + 4-gram	2.60%	6.47%

Table 1: Word Error Rate on test-clean and test-other datasets

Table 2 breaks down the evaluation results on male and female speech in both the test-clean and test-other datasets. It can be seen from these results that the Wav2Vec 2.0 + 4-gram model not only outperforms across gender splits but also shows in recognizing female speech, with a WER of 2.85% on test-clean female and 5.64% on test-other female datasets. For male voices,

Wav2Vec 2.0 reported a WER of 3.25% on test-clean and 9.58% on test-other. The 4-gram integration improved these figures to 2.34% and 7.26%, respectively.

WER	test-clean male	test-other male	test-clean female	test-other female
TDNN-F tgsmall	5.13%	13.84%	5.40%	11.16%
Wav2vec2	3.25%	9.58%	3,52%	7.58%
Wav2vec2 + 4-gram	2.34%	7.26%	2,85%	5.64%

Table 2: Word Error Rate on test-clean and test-other datasets, split by gender

The Table 3 demonstrates various examples of the predictions made by the TDNN-F, Wav2Vec 2.0, and Wav2Vec 2.0 4-gram on the test-clean dataset.

	Example 1	Example 2	Example 3
TDNN-F	after early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels (WER : 0.0)	does the head of a parrot with a little flower in his beak from a picture of carpaccio's one of his series of the life of saint george (WER: 0.07)	hedge offence (WER: 0.67)
Wav2vec2	after early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels (WER : 0.0)	it is the head of a parrot with a little flower in his beak from a picture of carpacios one of his series of the life of saint george (WER : 0.03)	hedge offence (WER: 0.67)
Wav2vec2 + 4-gram	after early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels (WER : 0.0)	it is the head of a parrot with a little flower in his beak from a picture of carpaccios one of his series of the life of saint george (WER : 0.03)	hedge offence (WER: 0.67)
Ground truth	after early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels	it is the head of a parrot with a little flower in his beak from a picture of carpaccio's one of his series of the life of saint george	hedge a fence

Table 3: Models' predictions on test-clean dataset

Table 3 shows that all models incorrectly transcribed the third example sentence, "hedge a fence," as "hedge offence." This misinterpretation could be attributed to the phonetic resemblance between "offence" and "a fence," combined with the accent of the speaker. In the second example, the two Wav2Vec 2.0 variations exhibit a minor error with the word "carpaccio's," possibly due to insufficient contextual information. While the TDNN-F model correctly identifies "carpaccio's," it falsely transcribes "it is" as "does the", altering the declarative sentence into a question. The first example illustrates a scenario where all models achieve perfect alignment with the ground truth,

Table 4 provides a comparison of sentence outputs from the TDNN-F, Wav2Vec 2.0, and Wav2Vec 2.0 integrated with a 4-gram language model on the test-other dataset, which contains more challenging acoustic scenarios.

Table 4: Model' predictions on test-other dataset

	Example 1	Example 2	Example 3
TDNN-F	now do i here six old fool its legs rattling behind one another (WER: 0.23)	what would you do papa how would you set about it (WER: 0.0)	why at once and half the size i lay (WER: 0.63)
Wav2vec2	now do i hear six old folts legs rattling behind one another (WER: 0.08)	what would you do papa how would you say to bout it (WER: 0.27)	wild ones and a half size elay (WER: 0.63)
Wav2vec2 + 4-gram	now do i hear six old fools legs rattling behind one another (WER: 0.0)	what would you do papa how would you say about it (WER: 0.09)	wild ones and a half size a la (WER: 0.63)
Ground truth	now do i hear six old fools legs rattling behind one another	what would you do papa how would you set about it	wild ones ain't alf the size i lay

In the first example, the Wav2Vec 2.0 models come close to the ground truth, with the 4-gram variant achieving perfect transcription with a WER score of 0.0 The TDNN-F, on the other hand, produces a transcription that diverges significantly from the grouth_truth, suggesting difficulties in capturing the correct sequence of words and possibly grammar.

For the second example, TDNN-F delivers an accurate transcription, indicating its capability to handle certain sentences well. The Wav2Vec 2.0 + 4-gram model nearly matches the ground truth, albeit with a small error. The standard Wav2Vec 2.0 model, however, exhibits a larger deviation from the ground truth, suggesting that the additional context provided by the 4-gram model can be beneficial.

In the third example, all models demonstrate a high WER of 0.63, indicating a universal challenge with this particular sentence. This could be attributed to the difficulties posed by the phonetic composition of the ground truth sentence, which affects the models' ability to discern the correct words, as the language models do not recognize the

7. Discussion and conclusion

The results indicate that all models performed better on the test-clean dataset compared to the test-other, which is expected as test-other contains noisy conditions. In addition, gender-specific results show that the models had varying levels of success across male and female voices, with the Wav2Vec 2.0 combined 4-gram model consistently showing the best performance. Generally, all models perform equally well on both genders in a the clean test dataset. However, in a noisy test other dataset, the three models seem to perform better on the female dataset than the male dataset. This is because females have higher voice pitch, which often have more energy in the higher frequency bands. These higher frequencies can provide more distinctive phonetic information, which can be easier for acoustic models to capture and distinguish.

Additionally, the variations in model prediction on the more complex test-other dataset confirm that the combination of acoustic and language models is more robust across speaker variability compared to only acoustic models.

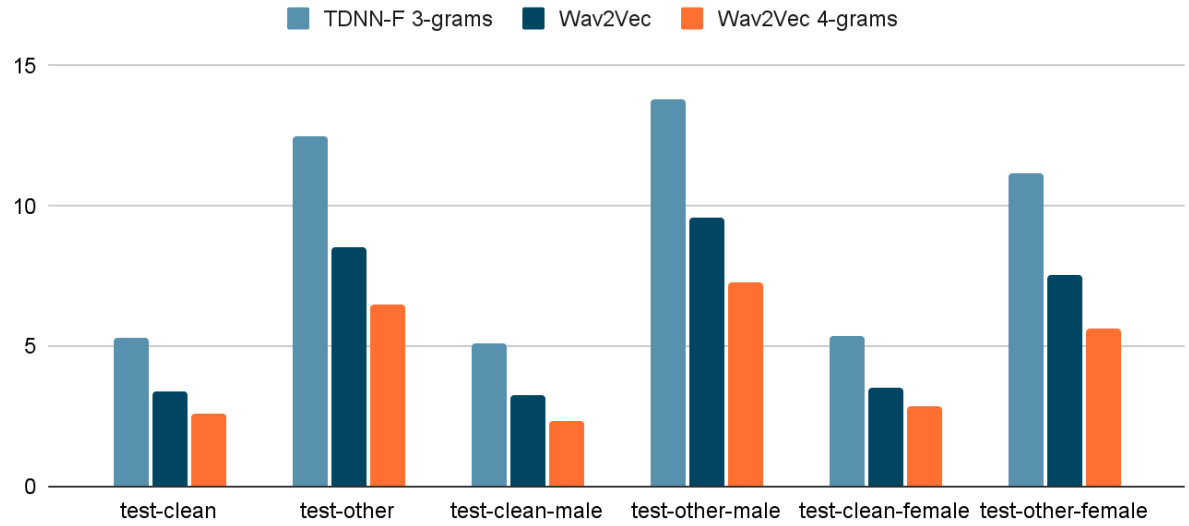


Figure 3: Word error rate comparison of the three models tested on LibriSpeech dataset

In conclusion, the combination of Wav2Vec 2.0 with a 4-gram language model not only enhances overall speech recognition accuracy but also improves the robustness against background noise and across different speaker attributes. While the acoustic model effectively decodes patterns in human speech, the language model is crucial in refining the output and aligning model prediction with linguistic patterns.

In future steps, we could proceed to use our models to test robustness of the pretrained models on other datasets, such as Wall Street Journal (WSJ) and Chime 6. The WSJ corpus contains North American speech newspaper articles, and it is primarily designed for Large Vocabulary Continuous Speech Recognition (LVCSR). On the other hand, Chime 6 uses the recordings from dinner party scenarios, where multiple speakers are conversing in a real home environment, leading to challenging settings in multi-Speaker ASR even to most robust models.

8. References

[1] Gales, M. J., & Young, S. (2007). The Journal of the Acoustical Society of America, 120(5), 3059-3075.

[2] Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education.

[3] Gao, J., Malpass, D., & Bridle, J. S. (2015). IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(4), 692-703.

[4] Huang, X., Acero, A., Hon, H. W., & Reddy, R. (2001). Speech and Audio Signal Processing. John Wiley & Sons.

[5] Hermansky, H. (2000). In Proceedings of the ISCA Workshop on Automatic Speech Recognition: Challenges for the Next Millennium.

[6] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). LibriSpeech: An ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5206-5210. doi:10.1109/ICASSP.2015.7178964

[7] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. ArXiv, abs/2212.04356.

[9] Hinton, G.E., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A.W., Vanhoucke, V., Nguyen, P., Sainath, T.N., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition. IEEE Signal Processing Magazine, 29, 82.

[10] Graves, A., Mohamed, A., & Hinton, G.E. (2013). Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 6645-6649.

[11] Xu, Y., Du, J., Dai, L., & Lee, C. (2015). A Regression Approach to Speech Enhancement Based on Deep Neural Networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23, 7-19.

[12] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. Advances in Neural Information Processing Systems.

[13] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-Vectors: Robust DNN Embeddings for Speaker Recognition. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5329-5333.

[14] Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Conference on Empirical Methods in Natural Language Processing.

[15] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N.K., Hannemann, M., Motlícek, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., & Veselý, K. (2011). The Kaldi Speech Recognition Toolkit.

[16] Morris, A.C., Maier, V., & Green, P.D. (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. Interspeech.

[17] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., Khudanpur, S. (2018) Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. Proc. Interspeech 2018, 3743-3747, doi: 10.21437/Interspeech.2018-1417

[18] Li, J., Meng, Y., Wu, Z., Meng, H., Tian, Q., Wang, Y., & Wang, Y. (2022). NeuFA: Neural network based end-to-end forced alignment with bidirectional attention mechanism. In 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-6). IEEE.

[19] Wang, Wei & Song, Wenjie & Chen, Chen & Zhang, Zhaoxin & Xin, Yi. (2019). I-Vector Features and Deep Neural Network Modeling for Language Recognition. Procedia Computer Science. 147. 36-43. 10.1016/j.procs.2019.01.181.

Division of labor

Two group members work on two different ASR models

Tran An: Literature review, Wav2Vec 2.0 and Wav2Vec 2.0 with n-gram language model (LM)

Xuan Binh: Kaldi software toolkit installation, writing shell scripts, TDNN-F with tgsmall LM

Acknowledgements

We acknowledge the contributors of the Kaldi software, a remarkable toolkit that has been indispensable in our speech recognition project. Its robust and versatile framework greatly facilitated our work and enhanced the quality of the WER results.

Additionally, we express appreciation to Guangpu Huang, the teaching assistant for this course. His support and expertise helps us understand the complexity of ASR concepts.

Appendix

Sample of Librispeech dataset

For each sample in the Librispeech dataset, it contains the audio file ending in .flac format.

For each respective speaker and chapter, there is a transcription of that audio contained in {speaker_id}_{chapter_id}.trans.txt, such as 374-180298.trans.txt

Here are four samples in the Librispeech Dataset

audio file	text	speaker_id	chapter_id	id
374-180298-0000.flac	CHAPTER SIXTEEN I MIGHT HAVE TOLD YOU OF THE BEGINNING OF THIS LIAISON IN A FEW LINES BUT I WANTED YOU TO SEE EVERY STEP BY WHICH WE CAME I TO AGREE TO WHATEVER MARGUERITE WISHED	374	180,298	374-180298-0000
374-180298-0001.flac	MARGUERITE TO BE UNABLE TO LIVE APART FROM ME IT WAS THE DAY AFTER THE EVENING WHEN SHE CAME TO SEE ME THAT I SENT HER MANON LESCAUT FROM THAT TIME SEEING THAT I COULD NOT CHANGE MY MISTRESS'S LIFE I CHANGED MY OWN	374	180,298	374-180298-0001
374-180298-0002.flac	I WISHED ABOVE ALL NOT TO LEAVE MYSELF TIME TO THINK OVER THE POSITION I HAD ACCEPTED FOR IN SPITE OF MYSELF IT WAS A GREAT DISTRESS TO ME THUS MY LIFE GENERALLY SO CALM	374	180,298	374-180298-0002
374-180298-0003.flac	ASSUMED ALL AT ONCE AN APPEARANCE OF NOISE AND DISORDER NEVER BELIEVE HOWEVER DISINTERESTED THE LOVE OF A KEPT WOMAN MAY BE THAT IT WILL COST ONE NOTHING	374	180,298	374-180298-0003

Code implementations

1) Kaldi Project and TDNN-F

The recommended installation script of Kaldi itself does not suffice and does not run right away on WSL. We have written a documentations on how to probably install Kaldi's dependencies on the Windows Subsystem for Linux (WSL). Additionally, the gpp version 4.8 is deprecated from the official installation command, so we need to compile it from source links.

Regarding replicating the results, readers can examine the folder librispeech_binh in egs folder. Open the s5, which contains the recipe folder for the LibriSpeech dataset composed by various. Since the provided recipe train the model from scratch and also tests on a different datatset, so we need to write new scripts in a pipeline to obtain decoding sentences and WER results

The Kaldi ASR project is forked from the source project with custom scripts implemented by Binh. The repository is hosted at this link

https://github.com/SpringNuance/kaldi-ASR

Instructions for installation on WSL: Kaldi_ASR/installation_instructions.txt

All installation shell scripts should be run at the root directory Kaldi_ASR/.

Instructions for result reproduction: Kaldi_ASR/egs/librispeech_binh/s5/pipeline_instructions.txt

All pipeline shell scripts should be run at the directory Kaldi_ASR/egs/librispeech_binh/s5

If readers follow correctly, they should be able to produce WER results like the figures below

```
springnuance@DESKTOP-JGURQ37:~/kaldi-ASR/egs/librispeech_binh/s5$ ./run_stage6_WER_tgsmall.sh
Current working directory: /home/springnuance/kaldi-ASR/egs/librispeech_binh/s5
utils/mkgraph.sh: exp/chain_cleaned/tdnn_1d_sp/graph_tgsmall/HCLG.fst is up to date.
steps/scoring/score_kaldi_wer.sh --cmd utils/parallel/run.pl --mem 2G data/test_clean exp/chain_cleaned/tdnn_1d_sp/graph_tgsmall
 exp/chain_cleaned/tdnn_1d_sp/decode_test_clean_tgsmall
steps/scoring/score_kaldi_wer.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 5.28 [ 2775 / 52576, 328 ins, 270 del, 2177 sub ] exp/chain_cleaned/tdnn_1d_sp/decode_test_clean_tgsmall/wer_12_0.0
steps/scoring/score_kaldi_wer.sh --cmd utils/parallel/run.pl --mem 2G data/test_other exp/chain_cleaned/tdnn_1d_sp/graph_tgsmall
 exp/chain_cleaned/tdnn_1d_sp/decode_test_other_tgsmall
steps/scoring/score_kaldi_wer.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 12.52 [ 6551 / 52343, 633 ins, 786 del, 5132 sub ] exp/chain_cleaned/tdnn_1d_sp/decode_test_other_tgsmall/wer_13_0.0
Done scoring.
```

```
springnuance@DESKTOP-JGURQ37:~/kaldi-ASR/egs/librispeech_binh/s5$ ./run_stage6_WER_tgsmall_male.sh
Current working directory: /home/springnuance/kaldi-ASR/egs/librispeech_binh/s5
utils/mkgraph.sh: exp/chain_cleaned/tdnn_1d_sp/graph_tgsmall/HCLG.fst is up to date.
steps/scoring/score_kaldi_wer.sh --cmd utils/parallel/run.pl --mem 2G data/test_clean_male exp/chain_cleaned/tdnn_1d_sp/graph_
tgsmall exp/chain_cleaned/tdnn_1d_sp/decode_test_clean_male_tgsmall
steps/scoring/score_kaldi_wer.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 5.13 [ 1316 / 25664, 157 ins, 121 del, 1038 sub ] exp/chain_cleaned/tdnn_1d_sp/decode_test_clean_male_tgsmall/wer_12_0.0
steps/scoring/score_kaldi_wer.sh --cmd utils/parallel/run.pl --mem 2G data/test_other_male exp/chain_cleaned/tdnn_1d_sp/graph_
tgsmall exp/chain_cleaned/tdnn_1d_sp/decode_test_other_male_tgsmall
steps/scoring/score_kaldi_wer.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 13.84 [ 3578 / 25846, 384 ins, 404 del, 2790 sub ] exp/chain_cleaned/tdnn_1d_sp/decode_test_other_male_tgsmall/wer_13_0.0
Done scoring.
```

```
springnuance@DESKTOP-JGURQ37:~/kaldi-ASR/egs/librispeech_binh/s5$ ./run_stage6_WER_tgsmall_female.sh
Current working directory: /home/springnuance/kaldi-ASR/egs/librispeech_binh/s5
utils/mkgraph.sh: exp/chain_cleaned/tdnn_1d_sp/graph_tgsmall/HCLG.fst is up to date.
steps/scoring/score_kaldi_wer.sh --cmd utils/parallel/run.pl --mem 2G data/test_clean_female exp/chain_cleaned/tdnn_1d_sp/graph_
tgsmall exp/chain_cleaned/tdnn_1d_sp/decode_test_clean_female_tgsmall
steps/scoring/score_kaldi_wer.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 5.40 [ 1453 / 26912, 178 ins, 140 del, 1135 sub ] exp/chain_cleaned/tdnn_1d_sp/decode_test_clean_female_tgsmall/wer_11_0.0
steps/scoring/score_kaldi_wer.sh --cmd utils/parallel/run.pl --mem 2G data/test_other_female exp/chain_cleaned/tdnn_1d_sp/graph_
tgsmall exp/chain_cleaned/tdnn_1d_sp/decode_test_other_female_tgsmall
steps/scoring/score_kaldi_wer.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 11.16 [ 2958 / 26497, 272 ins, 322 del, 2364 sub ] exp/chain_cleaned/tdnn_1d_sp/decode_test_other_female_tgsmall/wer_11_0.0
Done scoring.
```

2) Wav2Vec 2.0 model

Code for Wav2Vec 2.0 and Wav2Vec 2.0 + 4-gram language model can be found here:

https://github.com/an-tran528/wav2vec2-librispeech

In addition, the predictions of Wav2Vec 2.0 and Wav2Vec 2.0 + 4-gram with WER scores computed can also be found in the same repository.