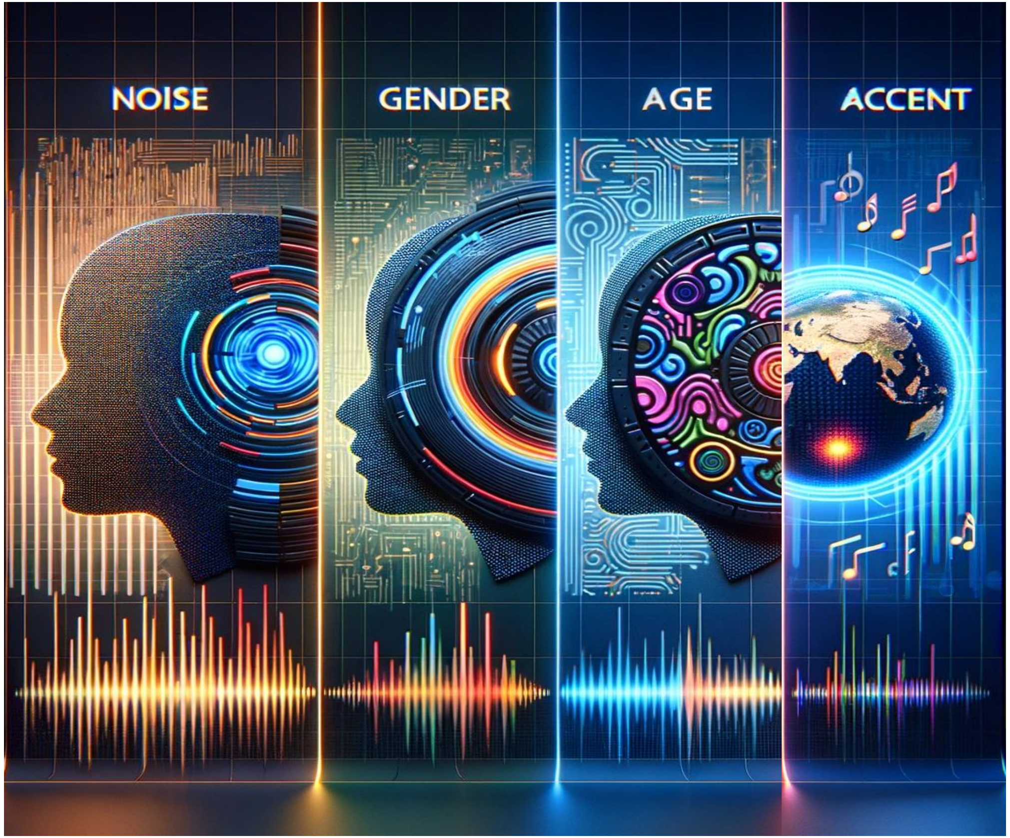
Group 15
Robust Speech Recognition Project
ELEC E5510

Members:
Xuan Binh (887799)
Tran An (487555)

Table of Contents

1. Introduction
2. LibriSpeech Dataset
3. Kaldi Project
4. General workflow
5. TDNN-F approach (baseline)
6. wav2vec approach
7. n-grams approach
8. Evaluation results
9. Discussions
10. Conclusions



Introduction to Robust Speech Recognition

- **Defining ASR**: Automatic Speech Recognition (ASR) technology translates spoken words into written text, facilitating human-machine communication.
- **Real-world Complexity**: ASR systems often grapple with variables like ambient noise, diverse dialects, and non-standard speech patterns, which can significantly impact accuracy.
- **Challenges and Solutions**: We'll explore common obstacles such as noisy environments, speaker variability, and speaker gender that ASR must overcome.
- **Benchmarking Robustness:** Our analysis includes testing the robustness of ASR systems by Word-Error-Rate to simulate real-life conditions.
- **Technological Advances**: We highlight the techniques like deep neural networks to enhance ASR robustness.
- **Project Objective:** The primary aim is to evaluate and improve ASR system performance, ensuring reliable recognition across different speakers and environments.

LibriSpeech ASR corpus

Identifier: SLR12

Summary: Large-scale (1000 hours) corpus of read English speech

Category: Speech

License: CC BY 4.0

Downloads (use a mirror closer to you):

dev-clean.tar.gz [337M] (development set, "clean" speech) Mirrors: [US] [EU] [CN]
dev-other.tar.gz [314M] (development set, "other", more challenging, speech) Mirrors: [US] [EU] [CN]
test-clean.tar.gz [346M] (test set, "clean" speech) Mirrors: [US] [EU] [CN]
test-other.tar.gz [328M] (test set, "other" speech) Mirrors: [US] [EU] [CN]

Reflects real-world complexities: clean and background noise, diverse speakers, and linguistic variations.

- Broad range of speakers: variations in accents, tones, and speaking styles.
- Metadata only reveals the gender of the speakers

Test-clean: 2620 clean speech utterances from 40 speakers (5.4 hours)

Test-other: 2939 utterances from the same speakers, but with more background noise and other distortions

Dataset Access: Available for download at OpenSLR.

Testing robustness in Librispeech

1. Robustness to noise/distortions: compare performance test-clean and test-other.
2. Robustness to speaker variations: compare performance on female and male speeches

subset	hours	per-spk minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Table 1. Data subsets in LibriSpeech [1]

build passing Gitpod Ready-to-Code Kaldi Speech Recognition Toolkit [2]

Official website: https://kaldi-asr.org Github: https://github.com/kaldi-asr/kaldi

- Kaldi is a state-of-the-art automatic speech recognition (ASR) C++ toolkit, containing almost any algorithm currently used in ASR systems.

- It also contains recipes for training our own 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.

General workflow of Kaldi

1. Transcript Preparation & Feature Extraction

- Obtain accurate transcripts of the speech data.

- Format transcripts following Kaldi's requirements

- Extract acoustic features from the audio using MFCC

2. Acoustic Model Training

- Train the initial acoustic models using target framework, such as HMM-DNN or TDNN-F

- Incorporate contextual information sets later stage for more complex models.

3. Audio Alignment & Contextual Modeling

- Force align audio with the initial acoustic models to optimize parameter estimation.

- Training monophone/ triphone models that consider the phoneme context, using phonetic decision trees

Refined Model Training

- Iterate the process of alignment and training

- Incorporating advanced techniques like LDA-MLLT or SAT to refine the triphone models.

Final Alignment & Model Optimization

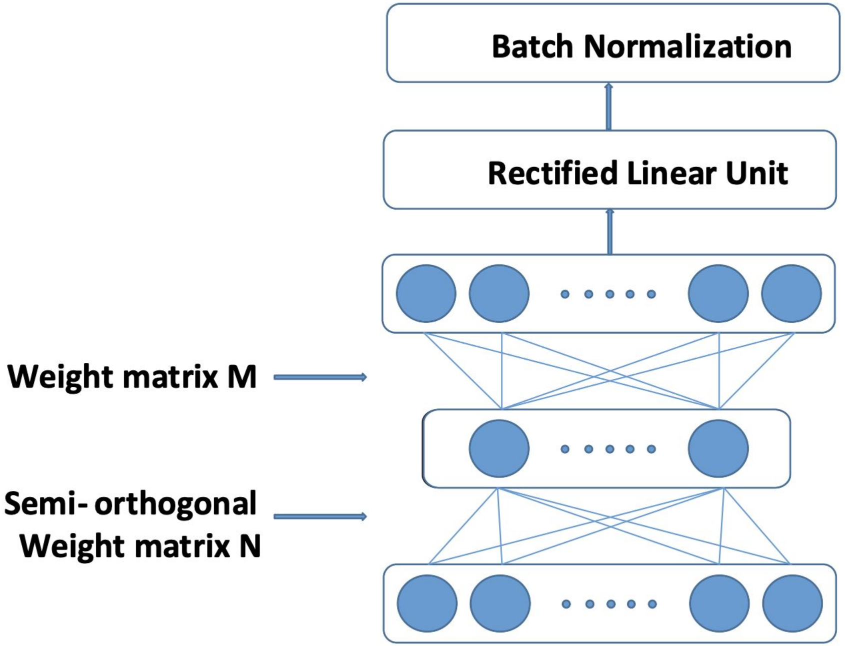
- Perform final alignments using SAT techniques like FMLLR to fine-tune the acoustic models.

- This stage ensures the models are robust and can generalize well across different speakers and contexts.

Tested models

1. TDNN-F (Time Delay Neural Network with Factored parameters)
2. Wav2vec 2.0 (Self-supervised wave to vector representation)
   a. Wav2vec 2.0 base
   b. Wav2vec 2.0 base + 4-gram Language model (LM)

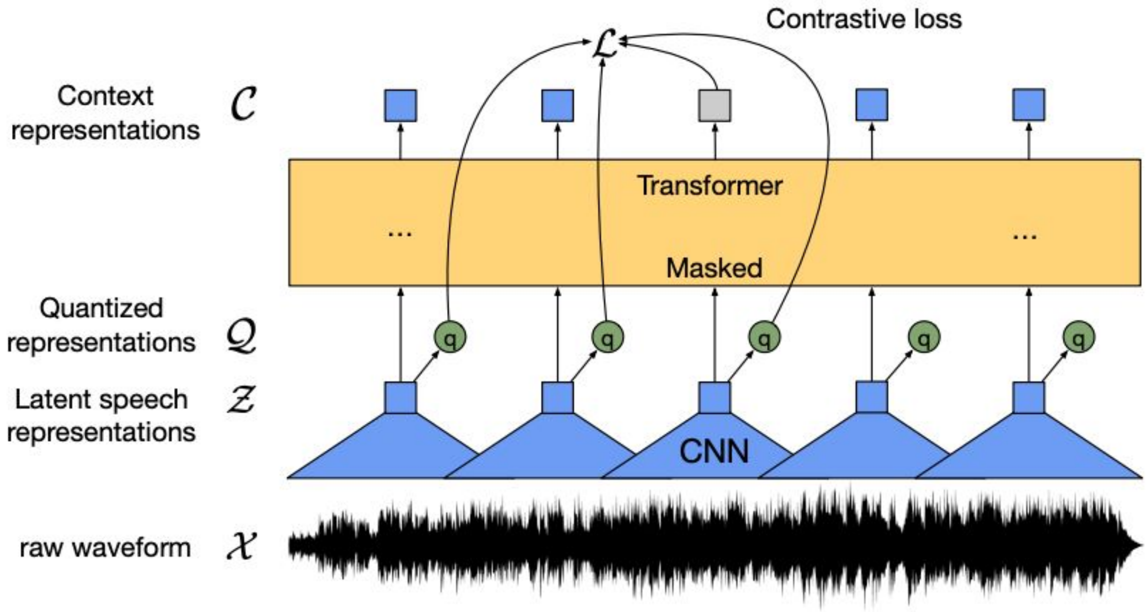
TDNN-F model



Factorized layer with semi-orthogonal constraint [3]

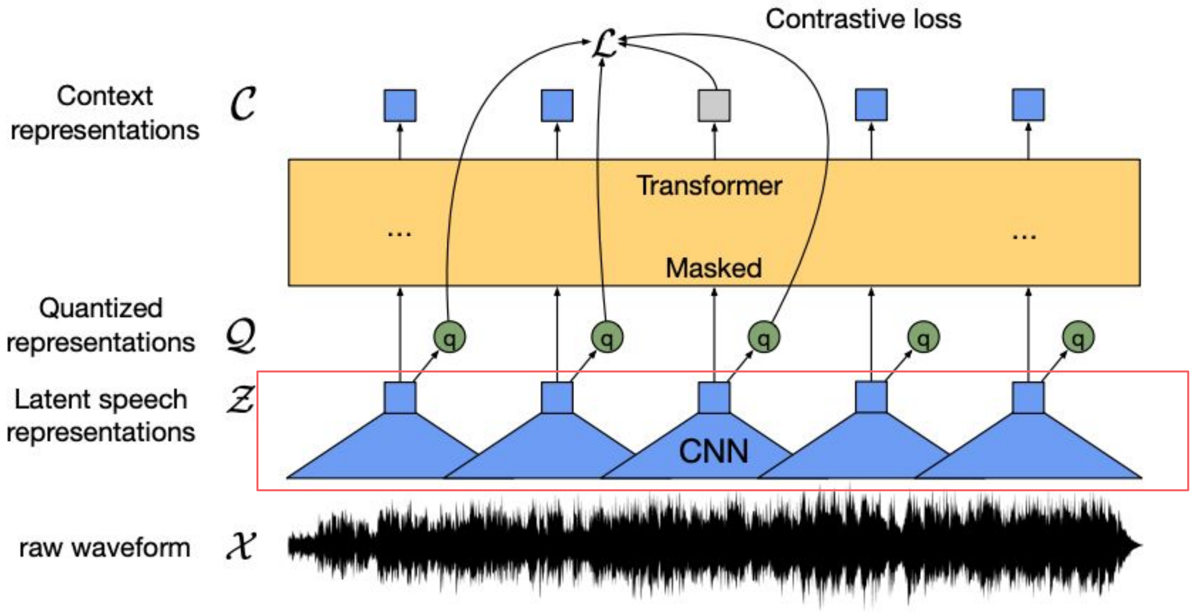
- Temporal Modeling of audio sequence
- Parameter Efficiency: The 'Factored' aspect of TDNN-F uses a low-rank matrix factorization approach
- Semi-Orthogonality: semi-orthogonal constraints on matrix factors
- Performance: TDNN-F models typically outperform standard TDNNs by achieving better accuracy with fewer parameters.
- Integration: TDNN-F is often used in Kaldi's chain models with LF-MMI training

Wav2Vec 2.0: Pre-training

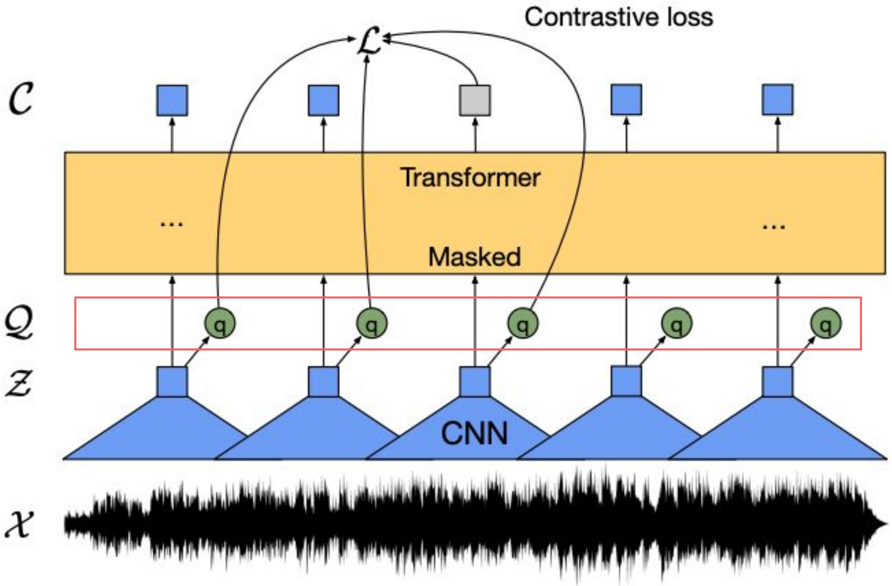


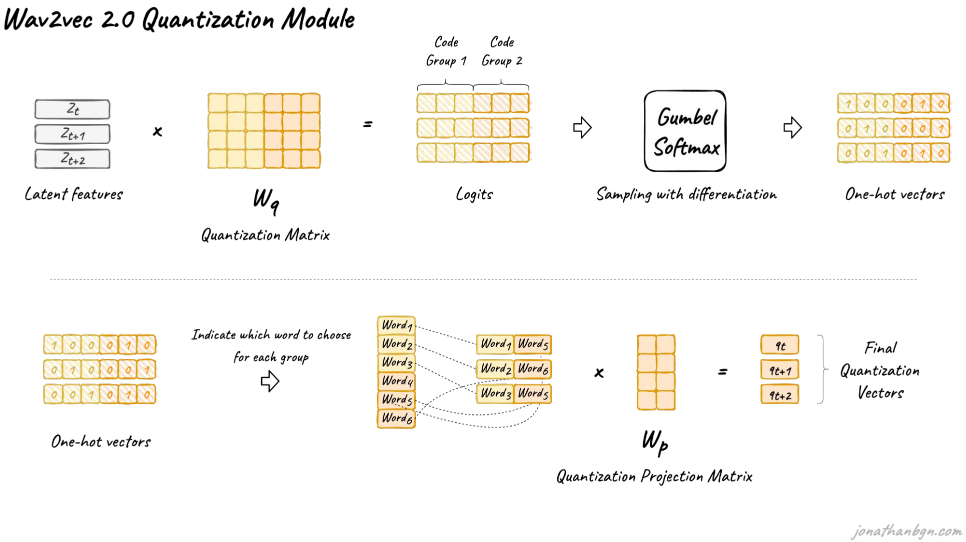
- Learning powerful representations from speech audio alone followed by fine-tuning on transcribed speech can outperform the best semi-supervised methods while being conceptually simpler [4]

Wav2Vec 2.0: Pre-training

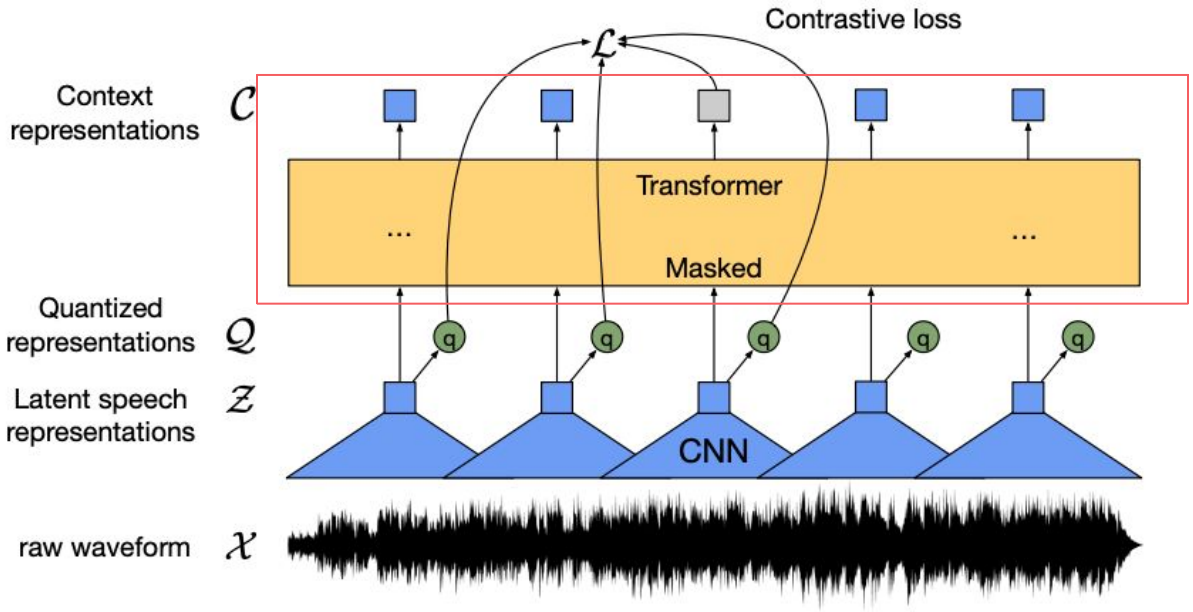


Wav2Vec 2.0: Pre-training

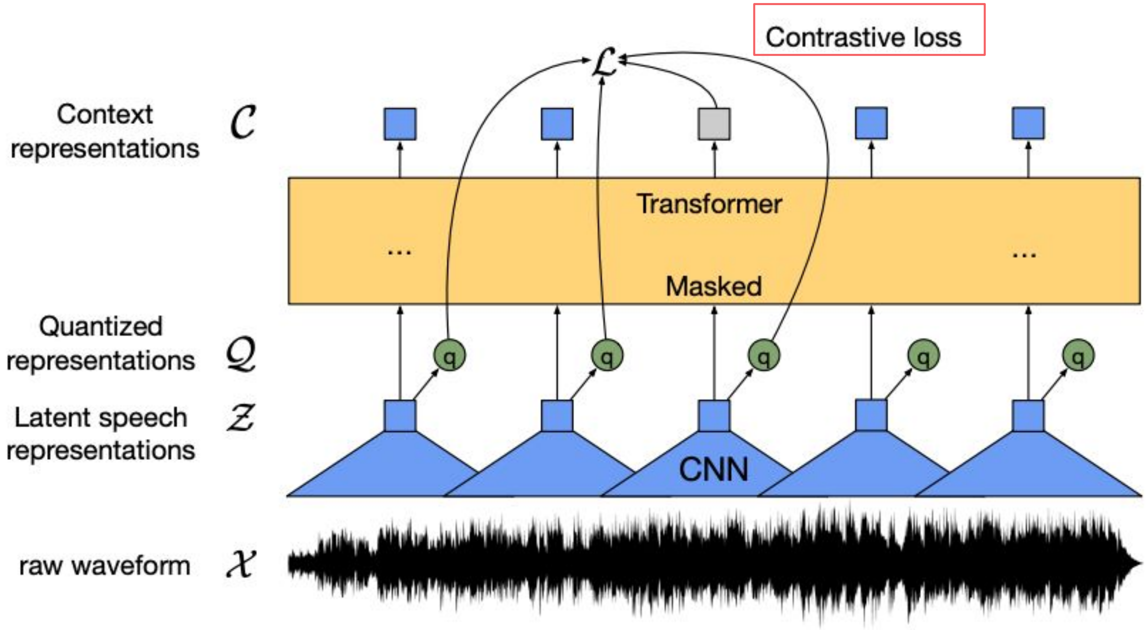




Wav2Vec 2.0: Pre-training



Wav2Vec 2.0: Pre-training



$$L = L_m + \alpha L_d$$

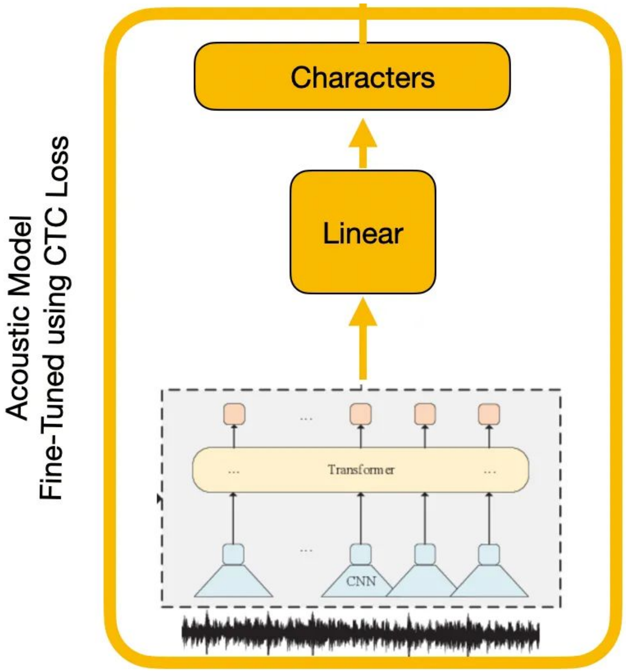
Contrastive loss:

$$L_m = -log \frac{exp(sim(c_t, q_t)/\kappa)}{\sum_{\tilde{q}\epsilon Q_t} exp(sim(c_t, \tilde{q})/\kappa)}$$

Diversity loss:

$$L_d = \frac{1}{GV} * (-H(\bar{p}_g)) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} log(\bar{p}_{g,v})$$

Wav2Vec 2.0: Fine tuning

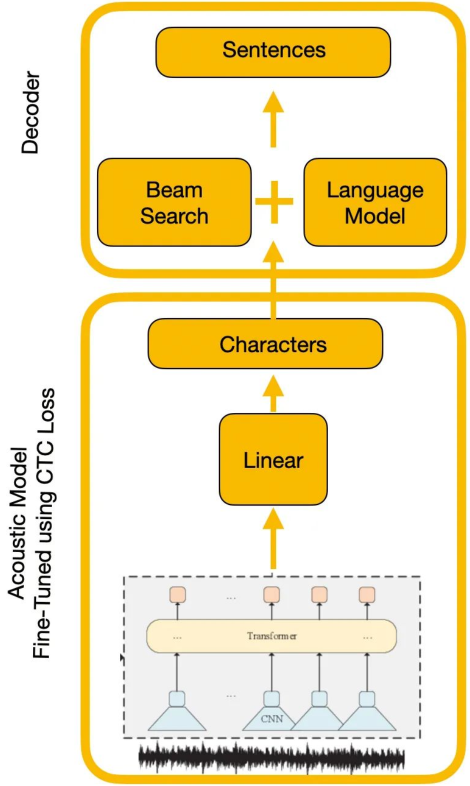


Classification to unique characters (from vocabulary)

Randomly initialized linear layer

Wav2Vec2 Contextual Representations

Wav2Vec 2.0 + n-gram LM



Beam Search + LM:
From characters to words to sentences

Classification to unique characters (from vocabulary)

Randomly initialized linear layer

Wav2Vec2 Contextual Representations

```
processor = Wav2Vec2Processor.from_pretrained(model_path)

decoder = build_ctcdecoder(
    labels=list(sorted_vocab_dict.keys()),
    kenlm_model_path="4gram.arpa",
)

processor_with_lm = Wav2Vec2ProcessorWithLM(
    feature_extractor=processor.feature_extractor,
    tokenizer=processor.tokenizer,
    decoder=decoder
)
```

Evaluation results

The TDNN-F model is tested on the LibriSpeech dataset as the baseline

WER	test-clean average	test-other average
TDNN-F 3 grams	5.28%	12.52%

WER	test-clean male	test-other male	test-clean female	test-other female
TDNN-F 3 grams	5.13%	13.84%	5.40%	11.16%

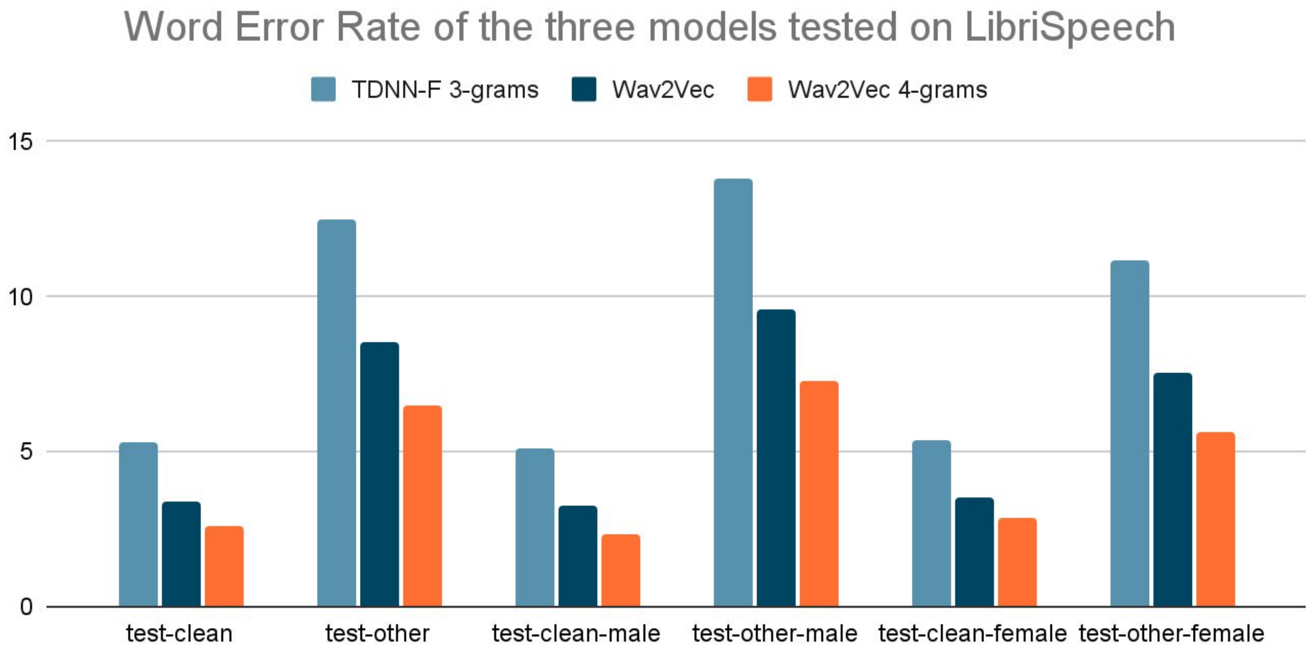
Evaluation results

wav2vec and n-gram models are tested on the LibriSpeech dataset

WER	test-clean average	test-other average
Wav2vec 2	3.386%	8.568%
Wav2vec 2 + 4-gram	2.601%	6.473%

WER	test-clean male	test-other male	test-clean female	test-other female
Wav2vec2	3.247%	9.582%	3.516%	7.579%
Wav2vec2 4gram	2.337%	7.264%	2.850%	5.642%

Discussions



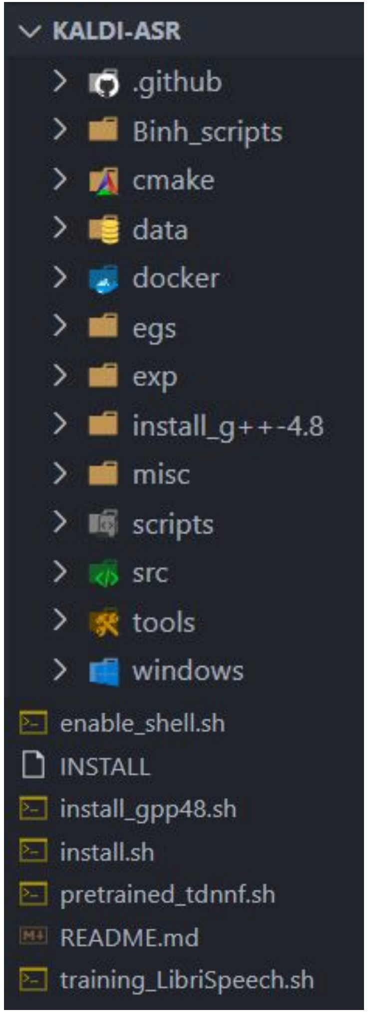
Model performance from best to worst

1. Wav2Vec 4-grams
2. Wav2Vec (no LM)
3. TDNN-F

WER on female dataset is better than WER on the male dataset

Conclusions

- Wav2vec2 + 4-gram model performs best results in all of the sub-datasets compared to Wav2vec2 and TDNN-F models
- Better performance in female speeches than male speeches.
- Worse testing performance on noisy environment than clean speech on LibriSpeech
- Kaldi is a powerful tool for researchers in ASR that supports all ASR stages and diverse models, albeit a steep learning curve
- We managed to install Kaldi project on the Windows Linux Subsystem, which is a lengthy process due to many dependencies
- Basic models are covered in Kaldi like HMM-GMM (Speaker Adaptive Training) to advanced models like NNLM toolkit



References

[1] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5206-5210). IEEE.

[2] Povey, D., Ghoshal, A., Boulianne, G., et al. (2011). The Kaldi Speech Recognition Toolkit. In IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society

[3] 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

[4] Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. ArXiv, abs/2006.11477.

Question and Answers? (Please don’t be hard 😢)