TRAINING EFFICIENT WAV2VEC2.0 ASR ENGINE TRANSFORMER MODEL FROM SCRATCH WITH DIFFERENT HEADS

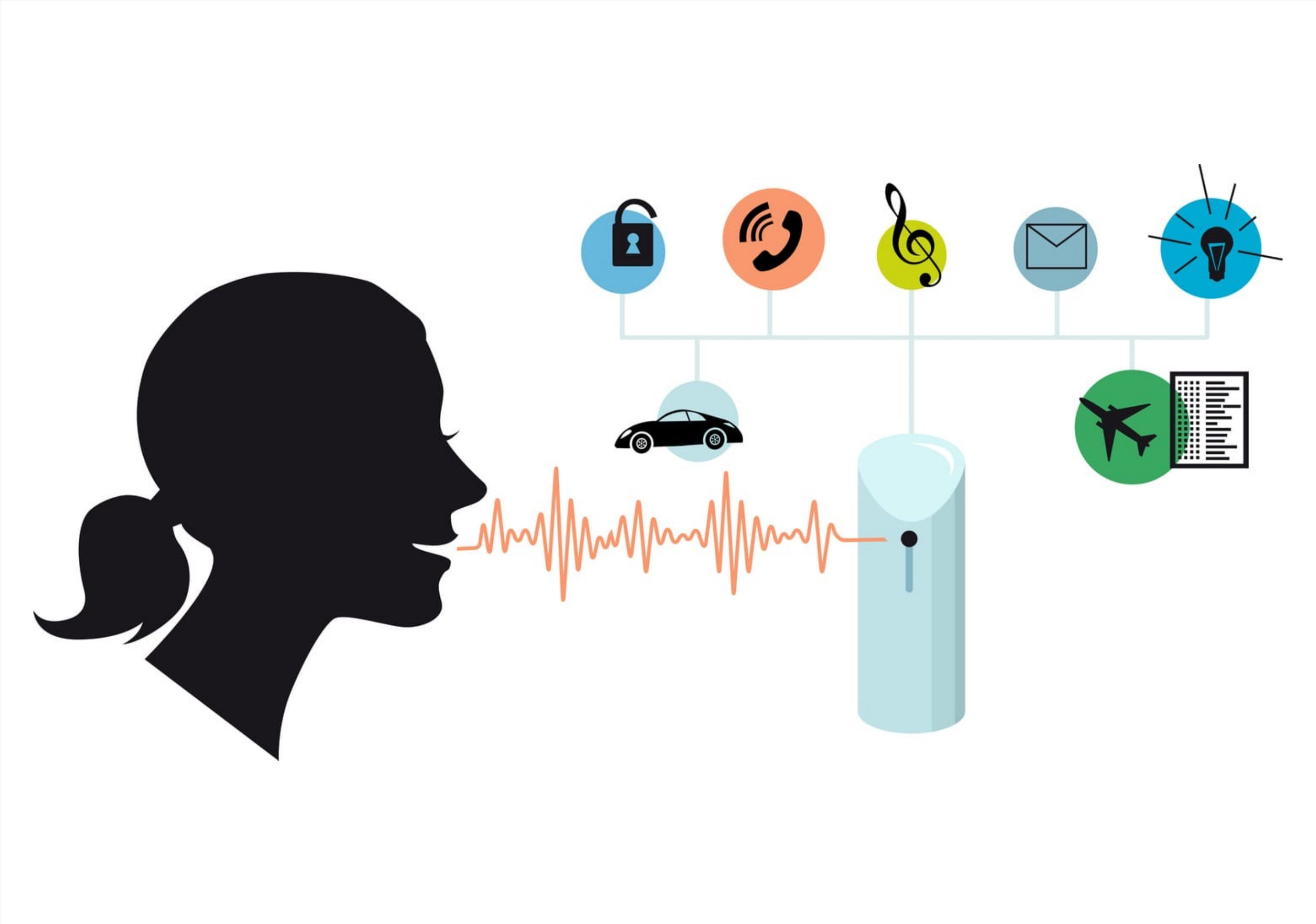
By: Tristin Johnson

DATS 6501: Data Science Capstone

March 21st, 2022

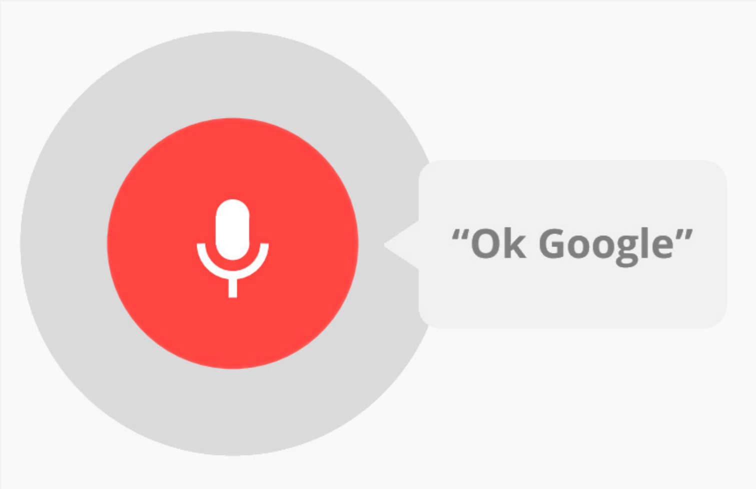
OVERVIEW

1. Introduction
2. Objective & Approach
3. Wav2Vec2.0
4. Datasets
5. Customized Training
6. Pre-Training
7. Fine-Tuning
8. Speech Classification
9. Limitations & Future Work
10. Conclusion



INTRODUCTION

- Automatic Speech Recognition (ASR) cover a wide variety of speech-related tasks
- ASR engines are extremely efficient, cheap and convenient across multiple different environment including:
  - Students in the classroom, business meeting, personal use, etc.
- ASR software also boosts productivity by providing the simplicity of having any speech automatically written down to a document
- Developing this type of engine can lead to multiple other in-depth research projects for any researchers out there trying to create ASR systems



OBJECTIVE & APPROACH

- **Project Objective:** Build and develop an Automatic Speech Recognition (ASR) Engine from scratch using state-of-the-art model architectures
- **Project Focus:** Pre-train an ASR engine that has the ability to transcribe any given audio file to its pertained text, fine-tune the model, then add a classification head to this model for classification speech-based datasets all while using custom-built methods and functions
- **Approach**:
  1. Gathering the data
  2. Speech & Data Preprocessing
  3. Develop custom-built pipeline
  4. Pre-train model from scratch (Wav2Vec2.0)
  5. Develop smaller versions of the pre-trained model
  6. Fine-tune and compare all models for speech recognition
  7. Apply and compare all models for speech classification

WAV2VEC2: INTRODUCTION

- "Wav2Vec 2.0 uses a self-supervised training approach for Automatic Speech Recognition, which is based on the idea of contrastive learning. Learning speech representation on a huge, raw (unlabeled) dataset reduces the amount of labeled data required for getting satisfying results." – Lukasz Sus
- Wav2Vec 2.0 is arguably the gold standard for ASR due to its self-supervised training, which is relatively new in the Deep Speech world.
- This way of training allows for pre-training a model on unlabeled data which is always more accessible.
- Then, this model can be fine-tuned on a particular dataset for a specific purpose.

WAV2VEC2: ARCHITECTURE

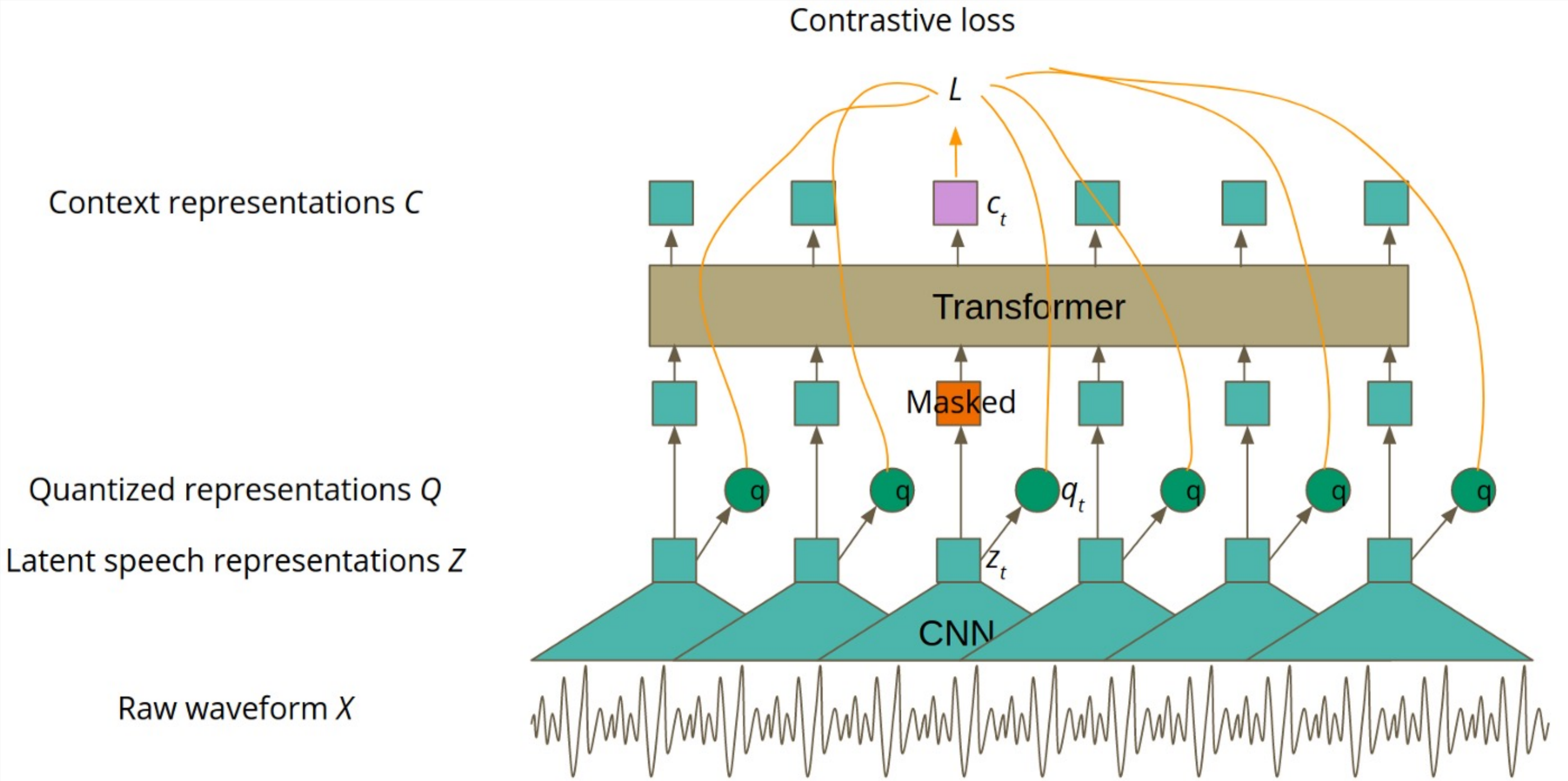


Figure 1: Wav2Vec2.0 architecture for self-supervised training (Image by Lukasz Sus)

WAV2VEC2: QUANTIZATION & MASKING

Quantization: The process of converting values from a continuous space into a finite set of discrete values

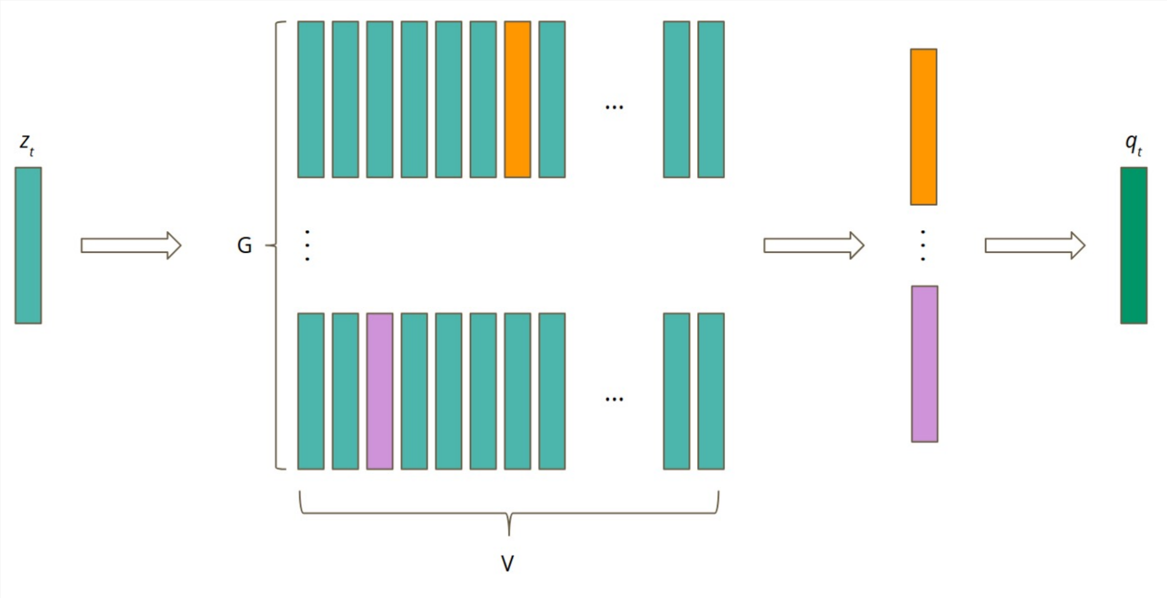


Figure 2:Wav2Vec2.0 quantized speech representation (Image by Lukasz Sus)

Masking:

- Take all time steps from space of latent speech rep. Z
- Sample without replacement proportion of vectors from previous step
- Chosen time steps are the starting indices
- For each index, consecutive M steps are masked

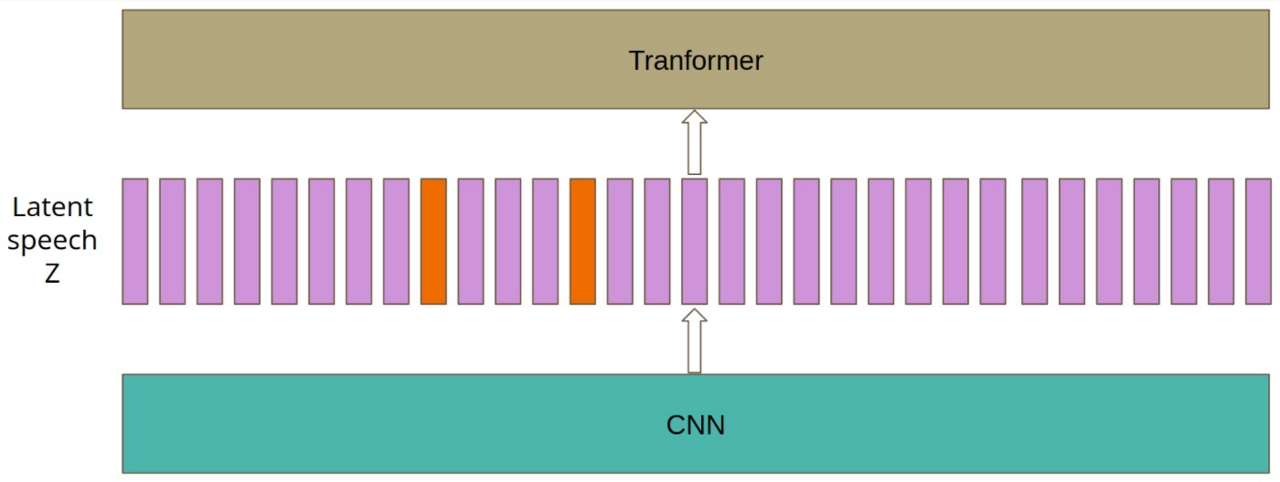


Figure 3:Wav2Vec2.0 masking speech representation (Image by Lukasz Sus)

DATASETS

- **Pre-Training ASR → LibriSpeech**
  - LibriSpeech is a corpus of approximately 1000 hours of 16kHz English speech (57 GB), prepared by Vassil Panayotov and Daniel Povey
  - The data is derived from English read audiobooks from the LibriVox project
- **Fine-Tuning ASR→ TI-MIT**
  - TI-MIT is an acoustic-phonetic speech corpus including 630 speakers of 8 different American English dialects, each reading 10 phonetically rich sentences
  - TI-MIT is known for including the phonemes of each speech recording
- **Speech Classification → RAVDESS**
  - The Ryerson Audio-Visual Database of Emotional Speech and Songs corpus includes 24 different speakers, 12 male and 12 female, vocalizing lexically matched statements in a neutral North American accent
  - Emotional classes: Neutral, Calm, Happy, Sad, Angry, Fearful, Surprise, Disgust

TRAINING: HUGGINGFACE TRAINER

1. Develop script to create CSV with metadata about speech files
2. **Load audio into HuggingFace Custom Dataset**
3. Clean dataset
4. Create custom vocabulary from dataset
    1. Wav2Vec2 Tokenizer, Feature Extractor, Processor
5. Prepare the data
    1. Librosa to load audio, get input values, encode translated text, implement custom data collator, WER for metrics
6. Load Wav2Vec2 Model
    1. Configuration for pre-training
    2. Pre-trained model for fine-tuning and classification
7. **HuggingFace Trainer and TrainingArgs**
    1. Pre-training, Fine-tuning, Speech Classification

TRAINING: PYTORCH

1. Develop script to create CSV with metadata about speech files
2. **Load audio into PyTorch Custom Dataset and DataLoader**
3. Clean dataset
4. Create custom vocabulary from dataset
    1. Wav2Vec2 Tokenizer, Feature Extractor, Processor
5. Prepare the data
    1. Librosa to load audio, get input values, encode translated text, implement custom data collator, WER for metrics
6. Load Wav2Vec2 Model
    1. Configuration for pre-training
    2. Pre-trained model for fine-tuning and classification
7. **PyTorch training framework**
    1. Pre-training, Fine-tuning, Speech Classification

WHAT HAPPENED TO PRE-TRAINING?

- Wav2Vec2.0 was originally pre-trained using 128 GPU's
- Took over 120 hours to complete
- Each batch was around 2.7 hours of audio

CUSTOMIZING & FINE-TUNING WAV2VEC2.0

- Original Wav2Vec2.0 model:
  - Pre-trained on ~960 hours of LibriSpeech Data
  - Total training parameters: ~95,000,000
- Medium-Sized Wav2Vec2.0 Model:
  - Total training parameters: ~61,000,000
- Small-Sized Wav2Vec2.0 Model:
  - Total training parameters: ~36,000,000

FINE-TUNING RESULTS ON TI-MIT

	WER	Loss	Total Time
Original Wav2Vec2.0 (~95M trainable parameters)	**16.298%**	**0.1254**	**17:14:36**
Medium Wav2Vec2.0 (~61M trainable parameters)	31.578%	69.999	13:48:26
Small Wav2Vec2.0 (~36M trainable parameters)	53.434%	103.774	12:40:58

SPEECH CLASSIFICATION RESULTS ON RAVDESS

	Accuracy	Loss	Total Time
Original Wav2Vec2.0 (~95M trainable parameters)	**89.121%**	**0.205**	**31:21:56**
Medium Wav2Vec2.0 (~61M trainable parameters)	88.897%	0.223	25:20:33
Small Wav2Vec2.0 (~36M trainable parameters)	85.776%	0.309	23:47:32

LIMITATIONS & FUTURE WORK

- **Limitations**
  - Computation Power
    - Pre-training is computationally expensive, and to properly pre-train requires lots of training time
    - Fine-tuning and classification could obtain better results
  - Financial Resources
    - Cloud Computing (GCP)
- **Future Work**
  - Pre-train Wav2Vec2.0
    - Apply custom pre-trained model to TI-MIT and RAVDESS datasets and compare results
  - Hyperparameter tuning on both TI-MIT and RAVDESS for better results, along with longer training time due to complexity of Wav2Vec2.0

CONCLUSION

- Accomplished several technical aspects of a building Machine Learning pipeline all from scratch
- Learned the details about pre-training a model from scratch
- Successfully fine-tuned Wav2Vec2.0 for speech recognition
  - Original model produced 16.298% WER
  - Obtained WER of 31.578% on custom medium-sized model
- Successfully applied a speech classification head to Wav2Vec2.0
  - Original Model produced 89.121% accuracy
  - Obtained 88.897% accuracy on custom medium-sized model
- Achieved competitive results against other professional Machine Learning engineers