Automatic Speech Recognition on Edge Devices

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Aidan Hijazi-Klop

Spring, 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Felix Lin, Department of Computer Science

Background

Historical speech or "conversational" AI models – voice assistants like Siri or Amazon Alexa, automated call centers, speech dictation systems, etc. – have utilized deep learning architectures, with Recurrent Neural Networks (RNN) and Hidden Markov Models (HMM) being the leading choices. The primary concern in the use of such architectures lies in the accuracy of speech processing, the common metric to measure this being word error rate (WER), especially when extending use of models beyond the English language. In response to this, recent development has shifted focus from "hybrid [or deep learning] models to end-to-end (E2E) models for [ASR]" with three leading approaches: "RNN transducer (RNN-T), RNN attention-based encoder-decoder (AED), and Transformer-AED" [12].

Introducing these E2E pipelines – while fixing accuracy concerns through improved feature extraction, more selective transcoding and decoding, and select NLP models (including but not limited to N-gram and BERT-based punctuation models) - produced a new concern: lack of necessary computing power on edge devices, the typical hosts for conversational AI models. To this end the technical capstone here will focus on the improvement of end-user accessibility of "[E2E] modeling for automatic speech recognition (ASR)" The move to improve accessibility to these state-of-the-art architectures by exploration of improved efficiency via various means, which will be expanded on below, is the core of the technical research problem.

Before delving into specifics of research goals, however, it is worth surveying key examples of these state-of-the -art models, the most notable being Meta's wave2vec 2.0, and Nvidia's Conformer-CTC and Jasper models. Wave2vec 2.0 is a self-supervised convolutional

neural network that optimizes labeled data using a connectionist temporal classification (CTC) loss function [2]. CTC Loss is a loss function designed to optimize neural nets focusing on sequence alignments (the scope of this is broad even including computational biology, but the use-case here is local alignment within encoded audio files. In a nutshell for audio work, the CTC function calculates loss between a segmented time-series audio stream and a specific encoded sequence by summing probabilities to produce a weighting for each input node; these loss values are differentiable with respect to the input node and form the weighted 'many-to-one' network between input nodes and the target. *C*areful application of CTC loss and the use of

The equation for a single target sequence from a set of input nodes is…

$$p(Y|X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} p_t(a_t|X)$$

sampling a Gumbel-Softmax distribution for learning discrete speech units in pre-training are core to the optimizations over wave2vec 1.0 which allow the model to "[achieve word error rate[s of] (WER) 4.8/8.2 on the clean/other test sets of Librispeech" [15]. Conformer-CTC, similar to wave2vec 2.0, use CTC loss to replace RNN-T loss in a traditional Conformer – convolution-augmented transformer, which is "designed to model both local and global dependencies of an audio sequence in a parameter-efficient way" [3] – model to improve sub-word and character level encodings and understandings. To conclude this survey, Jasper, another Nvidia produced model, is a time delay neural network TDNN that is built upon numerous layers of 1D convolutional layers [8]. While this survey is far from a sufficient exploration of each of these models, it is enough to establish the intricacy as these state-of-the-art architectures as context for the goals of this research.

There are two pertinent avenues of research, more likely to be explored serially than in parallel given their nature. The first is the profiling of the existing state-of-the-art models on a variety of systems, working down to the level of an edge device (or analogous amount of computational power). Based on the research done to date the natural approach to this seems to be to create a model agnostic benchmarking script and then use a combination of server-class resources and SoC devices (e.g., a Raspberry Pi) to run said script across a chosen selection of models that, ideally, utilize different underlying architectures; throughout this process, opportunities for architecture specific profiling may be discovered at which point discussions can be held on value in pursuing those additionally. The second avenue of research is looking at real-time speech decoding. Initially the thought was to look at direct waveform to label (to be used in a NLU) classification, how brief researched proved this to be out of scope for this capstone, as – per Chen et al. [4] and similar – the only forays into this are neuro-biological in nature and based on high-density electrocorticography. However, the queries were not without fruit as ML researchers do have some established work on the viability of "streamable Transformer-Transducer (T-T) models" [5]. This refocuses the second avenue instead on exploration, through profiling or otherwise, of the implications of streamable ASR on edge device computational performance. In addition to the well-defined paths above, a "moonshot" goal of sorts of this research project, should sufficient progress be made on avenues one and two, is to explore architecture optimization of the batch or E2E ASR models. The hope of the combination of these research efforts is to layout a path forward for accessible ASR on edge devices and take the first steps necessary in such development.

Selected Approach

Following sufficient establishment of background, four models were selected to be studied based on availability and known performance. Of the models selected two were pre-trained models - Wave2Vec [2] and Whisper [13] - and two were generic neural net architectures - Conformer and RNN-T [8]; the latter were selected on basis of having been the most robustly researched traditional architectures and the former two based on the significant leaps in model performance, measured by the metric of word error rate, as a result of pre-training and a shift to transformer based architectures. The Conformer and RNN-T models are detailed above in the background and while Wave2Vec 2.0 is touched on above as well, that and Whisper will be explained in brief to further contextualize analytical work.

Whisper, at least relative to other models discussed here, has a simple architecture in which the input audio stream is broken down into 30 second chunks, converted into a Log-Mel Spectrogram[1], and from there the spectrogram is passed through a pair of convolutional layers before being encoded (this architecture is visible below in Figure 1). Prediction of the next audio token is performed on the encoded audio tokens, drawing from a positionally encoded tokens in the training data (encoded tokens are often corresponded with phonemes[2], not the complete words themselves); this pattern of encoding and prediction persists across all languages, non-English languages making up ~33% of Whisper's 680k hours of training data [6].

---

[1] A Log-Mel spectrogram is a representation of the audio signal in which overlapping audio segments are superimposed using a fast fourier transform and then mapped to a logarithmic axis scale to map to the non-linear (Mel) scale that humans perceive frequencies along.

[2] A phoneme is a unit of sound that can distinguish one word from another in a particular language

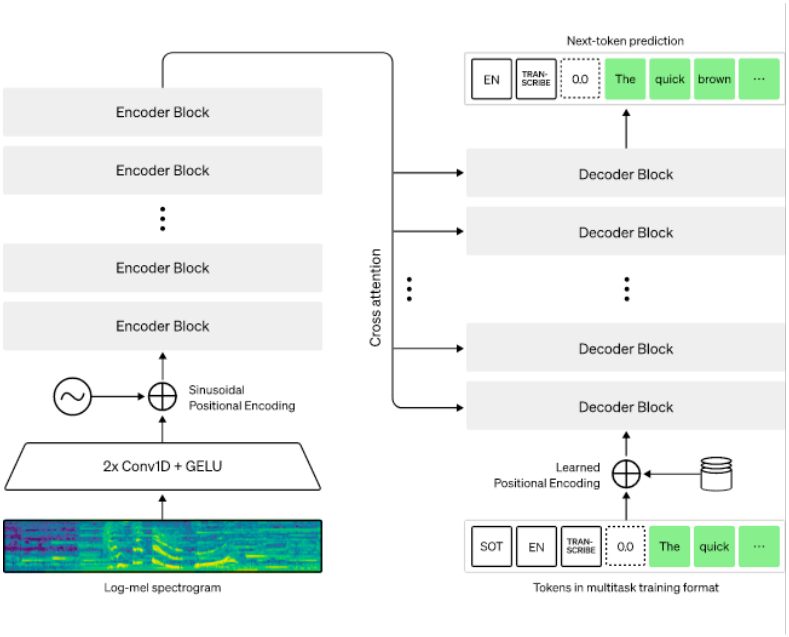

Figure 1: Whisper's Encoder-Decoder Transformer Architecture
Source: Adapted from [6]

Wave2Vec 2.0 while also a pre-trained model that also uses transformers, was trained on a significantly smaller corpus (of just under 1,000 hours) and uses the transformer and encoding architecture in a markedly different way to Whisper. The second iteration of Wave2Vec is based on the learning of a set of speech units that are smaller than phonemes. Wave2Vec also does initial processing using a stack of convolutional layers and encoders, the critical difference being in operating directly on the audio signal as opposed to a spectrogram. Based on the encoded speech, which is quantized to be operated on, some of the data is masked before being transformed and secondarily encoded and the inference of which speech units occurs as a result of a contrastive operation between masked and unmasked speech units [2].

With the critical models explained, discussion now is focused on the actual performance analysis. For the initial exploratory run each model was run against the OSR_us_0030_8k.wav file, an American English speech file part of the Open Speech Repository [16], and profile using python annotations with memory_profiler. The Wave2Vec, Conformer, and RNN-T models were

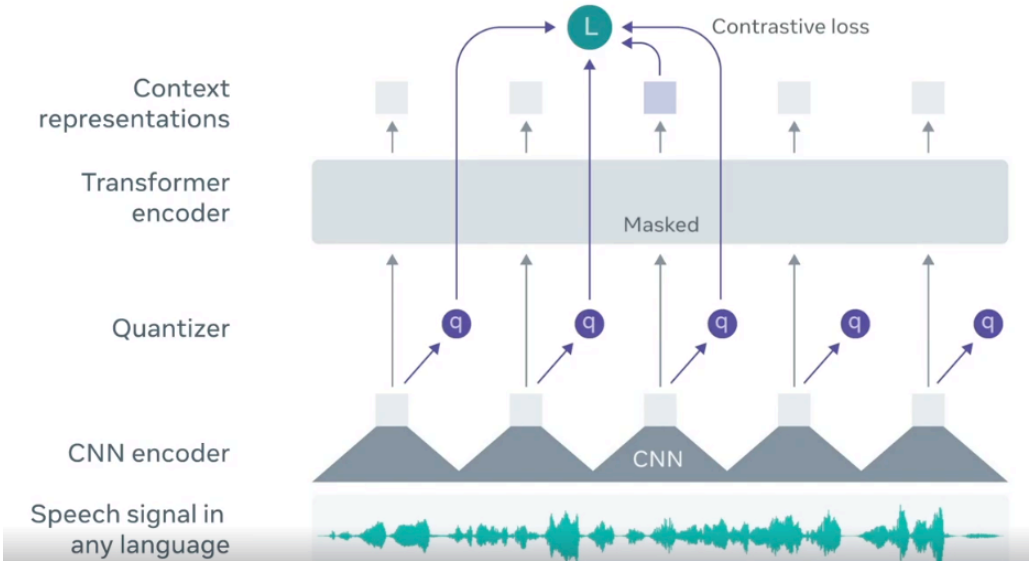

Figure 2: Illustration of Wave2Vec 2.0's architecture
Source: Adapated from [2]

called using PyTorch Audio and Whisper was called using the API's provided by OpenAI through HuggingFace. The metrics considered were time (s) to run the recognition, percentage of self CPU used, memory (Mb) used over the course of the call, the number of memory calls, and finally - and most importantly - word error rate. This initial and all future profiling work was run *on a RaspberrPi 3 over UART with 64-bit quad-core ARM Cortex-A53 @ 1200 MHz*

Model Name	Time (s)	Self CPU (%)	Mem (Mb)	WER (%)	# of Calls
Wave2Vec 2.0	19	54.7%	430	8.2%	297
Conformer	31	67.2%	925	11.4%	2680
RNN-T	26	61.9%	810	10.5%	5576
Whisper (HF)	21	46.3%	385	8.4%	487

Table 1: Initial Benchmarking Results

After completion of preliminary benchmarking, additional research of existing work and the journal "Performance Evaluation of Offline Speech Recognition on Edge Device" was

encountered. Similar to the work conducted here this group aimed to evaluate “resource constraints on smaller edge devices may pose challenges for achieving state-of-the-art speech recognition results” [10]. The methodologies in use in this paper served a twofold purpose: one, to shift my benchmarking tool to PyProfiler which yielded higher fidelity results, and as a product of being a more refined interface was easy to run over a much larger set of test speech data, 20 files from the Open Speech Repository to be exact. These results are included below.

Model Name	Time (s)	Self CPU (%)	Mem (Mb)	WER (%)	# of Calls
Wave2Vec 2.0	19.34	56.57%	440.05	8.64%	278
Conformer	31.15	67.59%	933.12	12.75	2798
RNN-T	26.57	64.83%	813.85	11.62%	5613
Whisper (HF)	20.61	45.59%	387.58	8.34%	488

Table 2: Larger corpus profiling results w/ PyProfiler

Commentary on Outcome

For purposes of further research there are two avenues of focus. One, is the application of the previously introduced CTC loss to a variety of pre-trained models as up until now it has only been notably experimented with for Wave2Vec; the performance benefits of CTC loss are large, making it an low computational cost addition to ASR pipelines that warrants further pursuit. The second scope of further research is on device domain training of pre-trained models, particularly Whisper, and optimization of the high dimensional probability space of the hyper-parameters available for Whisper and related models.

The benchmarking of the selected models revealed a clear pattern that the pre-trained, transformer based models notably outperformed the more traditional, non pre-trained models in

every metric, by significant margins in some cases. This conclusion is unsurprising as when a major computational burden is shifted off of the edge device (the original training for Whisper and Wave2Vec 2.0) it allows that step to not be constrained by computational limitations, in this case of the Raspberry Pi. In short this yields the conclusion that as the use of ASR models expands there should be a continued shift to pre-trained models leaving any on device training to be very domain specific; the more power that can, in a sense, be front-loaded into ASR models better sets up equitable access and democratization of speech models.

References

[1]
M. Mhiri, S. Myer, and V. S. Tomar, "A Low Latency ASR-Free End to End Spoken Language Understanding System," in *Interspeech 2020*, ISCA, Oct. 2020, pp. 1947–1951. doi: 10.21437/Interspeech.2020-1449.

[2]
J. Boigne, "An Illustrated Tour of Wav2vec 2.0," *Jonathan Bgn*, Sep. 30, 2021. https://jonathanbgn.com/2021/09/30/illustrated-wav2vec-2.html (accessed Sep. 21, 2022).

[3]
A. Gulati *et al*., "Conformer: Convolution-augmented Transformer for Speech Recognition." arXiv, May 16, 2020. Accessed: Sep. 21, 2022. [Online]. Available: http://arxiv.org/abs/2005.08100

[4]
X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing Real-Time Streaming Transformer Transducer for Speech Recognition on Large-Scale Dataset," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 5904–5908. doi: 10.1109/ICASSP39728.2021.9413535.

[5]
"End-to-End Speech AI Pipelines," *NVIDIA*. https://resources.nvidia.com/en-us-speech-ai-ebooks/speech-ai-using-asr-and-tts (accessed Sep. 21, 2022).

[6]
"Introducing Whisper." https://openai.com/research/whisper (accessed May 03, 2023).

[7]
"Jasper for PyTorch | NVIDIA NGC," *NVIDIA NGC Catalog*. https://catalog.ngc.nvidia.com/orgs/nvidia/resources/jasper_for_pytorch (accessed Sep. 21, 2022).

[8]
"Models — NVIDIA NeMo." https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/main/asr/models.html (accessed Sep. 21, 2022).

[9]
J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, "[PDF] On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition | Semantic Scholar." https://www.semanticscholar.org/reader/be934c378c897e5bc3b3767376a59a4679093286 (accessed Sep. 12, 2022).

[10]
S. Gondi and V. Pratap, "Performance Evaluation of Offline Speech Recognition on Edge Devices," *Electronics*, vol. 10, no. 21, Art. no. 21, Jan. 2021, doi: 10.3390/electronics10212697.

[11]
"Real-time decoding of question-and-answer speech dialogue using human cortical activity - PMC." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6667454/ (accessed Sep. 21, 2022).

[12]

J. Li, “Recent Advances in End-to-End Automatic Speech Recognition,” *SIP*, vol. 11, no. 1, Apr. 2022, doi: 10.1561/116.00000050.

[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” p. 28.

[14] “STT En Conformer-CTC Medium | NVIDIA NGC,” *NVIDIA NGC Catalog*. https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_conformer_ctc_medium (accessed Sep. 21, 2022).

[15] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.” arXiv, Oct. 22, 2020. Accessed: Sep. 21, 2022. [Online]. Available: http://arxiv.org/abs/2006.11477

[16] “Open Speech Repository |.” http://www.voiptroubleshooter.com/open_speech/american.html (accessed May 04, 2023).