Enhancing Domain-Specific ASR Performance Using Finetuning and Zero-Shot Prompting: A Study in the Medical Domain

Utsav Bandyopadhyay Maulik[1,†], Pabitra Mitra[1,*,†] and Sudeshna Sarkar[1,†]

[1] *Dept. of Computer Science and Engineering, IIT Kharagpur, West Bengal, India*

Abstract

Domain Adaptation has emerged as an important development in Speech Recognition systems for improving the transcription accuracy of the input audio. This study explores the enhancement of Domain-specific Automatic Speech Recognition performance through finetuning and postprocessing using Large Language Models, focusing specifically on the medical domain. We investigate how domain-specific finetuning and advanced text postprocessing techniques can significantly improve transcription accuracy in medical contexts, reducing errors in specialized terminology, acronyms, and abbreviations. Our findings highlight the benefits of integrating Large Language Model based postprocessing with Automatic Speech Recognition systems to achieve better results in complex domains.

Keywords

ASR, Finetuning, LLM, Postprocessing, Medical Domain, Domain Adaptation,

1. Introduction

Automated Speech Recognition (ASR) enables computers to transcribe spoken language into text and has evolved significantly from statistical methods [1, 2, 3] to advanced end-to-end deep learning models. Key advancements in this shift include works by [4] and [5], which highlighted deep learning's role in end-to-end ASR systems. Despite these improvements, challenges persist in domain-specific fields like medicine, where specialized vocabulary and jargon pose difficulties. Medical ASR faces constraints such as limited labeled data, complex terminologies, accents, dialect variations, unheard terms, and privacy concerns, causing state-of-the-art models to underperform. Domain Adaptation (DA) is therefore essential to address these limitations effectively.

Domain Adaptation [6] involves tailoring a machine learning model to perform effectively on data from a domain different from its training domain. In speech recognition, domain adaptation is crucial. For example, an ASR system trained on conversational English may struggle with legal proceedings or technical support calls, where language and context deviate significantly from the training data. Similarly, conversations in specialized fields like medicine contain complex terms, acronyms, and unique phrases that general ASR models may fail to transcribe accurately. For instance, a standard ASR system may incorrectly transcribe medical terms like “hypertension” or “tachycardia”, leading to confusion or errors. Domain adaptation addresses these challenges by integrating domain-specific knowledge into the model.

Even with fine-tuning, domain-specific ASR systems may produce imperfect outputs, making postprocessing crucial. Postprocessing improves ASR results by correcting errors, refining text, and

The 2024 Sixth Doctoral Symposium on Intelligence Enabled Research (DoSIER 2024), November 28–29, 2024, Jalpaiguri, India

[*] Corresponding author.

[†] These authors contributed equally.

utsav2000@gmail.com (U. B. Maulik); pabitra@gmail.com (P. Mitra); shudeshna@gmail.com (S. Sarkar)

enhancing accuracy and readability. In medical transcription, for instance, minor errors can lead to significant misunderstandings, highlighting the importance of postprocessing to identify mistakes, ensure proper formatting, and refine clinical notes or prescriptions.

Large Language Models (LLMs), trained on extensive text datasets, excel at learning patterns, context, and relationships between words. They have transformed postprocessing for ASR systems by intelligently improving raw transcriptions through context understanding, error correction, and word prediction. For example, in medical contexts, phrases like *high tension* can be accurately corrected to *hypertension* or *pencil in* to *penicillin* based on the context.

Traditional language models, such as n-grams, rely on fixed word sequences and statistical probabilities to predict text. These models analyze word frequency and patterns but struggle with complex or less frequent combinations. In contrast, LLMs like GPT-3, trained on massive datasets, capture not only word sequences but also deeper semantic meaning and context across sentences or paragraphs. This allows them to handle diverse tasks, from answering questions to generating detailed, coherent text, with greater versatility and accuracy. For instance, while a classical model might predict "he is going to" based on frequency, an LLM could predict "he is going to the hospital for surgery" by fully grasping the context.

In this work, we address the afore-mentioned challenges of domain specific ASR in the medical field. We focus on developing a method using open-source, publicly available models and data sets, making it ideally suited for use of the entire community. Pre-trained ASR models are used which is further fine-tuned on the domain specific datasets without hampering their generalizations. LLMs are then integrated on these fine-tuned ASR models to further enhance domain specific word recognition.

2. Related Work

Most modern ASR systems leverage deep learning, particularly Recurrent Neural Networks (RNNs) [7] [8] and Transformer models [9]. Tools like Google's Speech-to-Text API, Microsoft's Azure Speech Services, and OpenAI's Whisper model have made ASR scalable and accessible.

Earlier, RNNs, particularly Long Short-Term Memory (LSTM) networks [10], marked an early advancement in handling sequential data by maintaining context over longer text spans. [11] explores the use if LSTM-based models for such applications. More recently, Transformer-based models, [9], have revolutionized ASR by allowing for parallel processing and capturing much larger contexts in both directions. This self-attention mechanism used in transformers enables the model to consider the entire sentence or even multiple sentences when predicting the next word, thus significantly improving the system's ability to handle complex or domain-specific language. Advances in speech recognition have been driven by the rise of self supervised and unsupervised pre-training methods, such as Wav2Vec 2.0 [12]. Alec Radford et al in their work of the Whisper ASR model [13], for instance, employs such architectures to improve context understanding and achieve high transcription accuracy across various languages and domains.

Recent work on domain-specific ASR has focused on techniques like transfer learning, where a general ASR model is adapted to a specific domain by fine-tuning it on smaller, domain-specific datasets. Gulati et al in [14] propose Conformer models demonstrating the effectiveness of transfer learning for domain adaptation. Similarly, Chen et al in [15], explored fine-tuning pre-trained models like Wav2Vec 2.0, showing that this technique can significantly improve recognition accuracy, particularly for emotion detection in speech. Liu et al. in their paper [16] "Exploration of Whisper Fine-Tuning Strategies for Low-Resource ASR", explored various strategies for fine-tuning the Whisper ASR model in low-resource environments.

While finetuning can improve the performance of ASR models, errors still occur, especially in the transcription of medical jargon, acronyms, and abbreviations. Postprocessing techniques may be used to refine the ASR outputs to address this. Traditional rule-based correction systems and classical language models, such as n-gram models, were used to detect and correct errors in transcription. The advent of Large Language Models (LLMs) like GPT-3 [17], GPT-4, and LLAMA (Large Language Model Meta AI) [18] has opened up new possibilities for postprocessing ASR outputs. For example, [19], in their paper "The Sound of Healthcare: Improving Medical Transcription ASR Accuracy with Large Language Models" (Google Cloud), explored how integrating large language models can significantly enhance the accuracy of medical transcription produced by ASR systems. However, they have used only commercially available models.

The integration of ASR systems in the medical domain is not new. ASR technology has been used in radiology, electronic health record (EHR) documentation, and telemedicine. However, achieving high accuracy remains a challenge due to the complexities of medical speech, which often includes technical language, accents, bad quality audios, and noisy environments.

3. Methodology

As shown in Figure 1, our approach involves integrating Automatic Speech Recognition (ASR) models with Large Language Models (LLMs) to improve transcription accuracy, particularly in medical conversations. Initially, the pre-trained ASR models were used to generate raw text predictions from the speech inputs. These predictions were then passed through the LLMs, leveraging the prompt engineering techniques to refine the outputs.

Subsequently, we fine-tuned the pre-trained ASR models on 80% of the dataset, leaving the remaining 20% as unseen data for evaluation. The fine-tuned ASR models were then used to generate predictions on the unseen test data. These ASR outputs were passed into the LLMs, where prompt engineering techniques were applied once again. This final step enabled us to generate more accurate, contextually refined predictions for medical conversations, improving the system's overall performance.

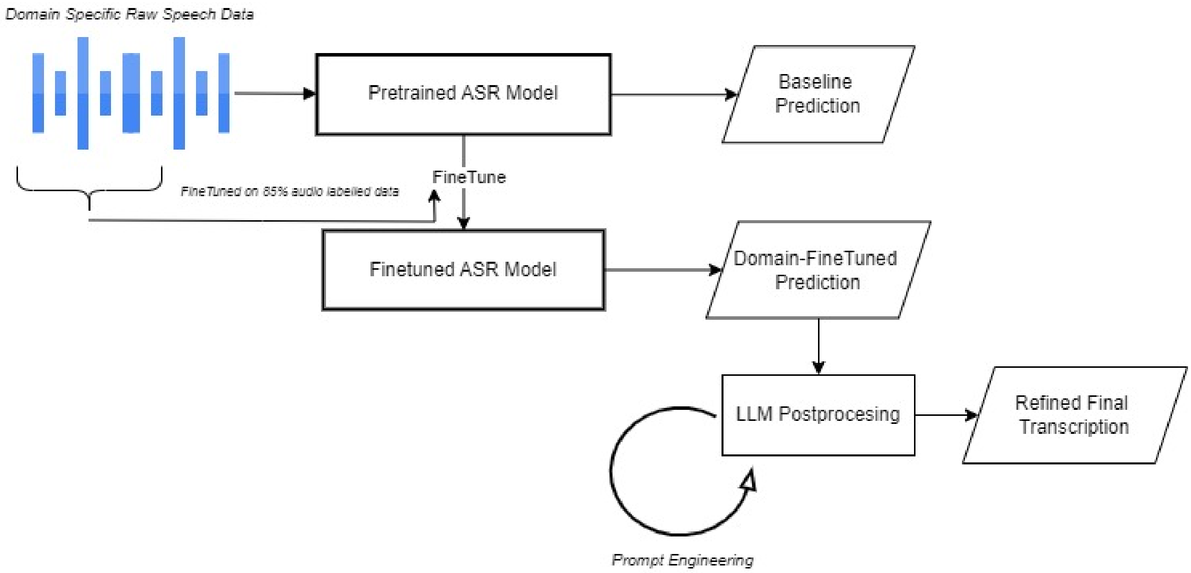
Domain Specific Raw Speech Data
Pretrained ASR Model
Baseline Prediction
FineTune
FineTuned on 85% audio labelled data
Finetuned ASR Model
Domain-FineTuned Prediction
LLM Postprocesing
Refined Final Transcription
Prompt Engineering

Figure 1: Method Pipeline

3.1. Automatic Speech Recognition

This study evaluates four prominent ASR models: *wav2vec2-base-960h* and *wav2vec2-large-960h* from Facebook [12], and *whisper-small* and *whisper-large* from OpenAI [13].

Wav2Vec 2.0 [12], developed by Facebook AI, is a state-of-the-art self-supervised ASR model designed to learn speech representations from large amounts of unlabeled audio data. The wav2vec 2.0 model learns general speech representations by masking random portions of the input waveform and training the model to predict the masked regions. This approach is akin to BERT-style pre-training for natural language processing.

- **Base (wav2vec2-base-960h)**: The base model has 95 million parameters and is trained on the 960-hour Librispeech dataset. It consists of a convolutional feature encoder followed by Transformer layers.
- **Large (wav2vec2-large-960h)**: The large model has 317 million parameters, offering more capacity and improved performance due to the deeper Transformer architecture. It's also trained on the 960-hour Librispeech dataset.

Whisper [13] is a model developed by OpenAI, designed as a general-purpose speech recognition system. Unlike many ASR models, Whisper is capable of multilingual transcription and translation tasks, making it highly versatile. Whisper is trained in a fully supervised manner on an extensive dataset of 680,000 hours of labeled speech data sourced from the web.

- **Whisper Small**: This model variant contains 244 million parameters, using an encoder-decoder architecture similar to those found in sequence-to-sequence models.
- **Whisper Large**: This model variant contains approximately 1.5 billion parameters, utilizing an encoder-decoder architecture akin to those used in sequence-to-sequence models. Its extensive parameter count enables it to capture more intricate patterns in audio, resulting in improved accuracy and robustness in diverse acoustic environments.

All of these ASR models we used are open-sourced, and downloaded from the HuggingFace library.

3.2. Large Language Model

In our study, two primary variants of LLaMA 3 (Large Language Model Meta AI) were utilized for postprocessing tasks: LLaMA 3 (8 billion parameters) and LLaMA 3 (70 billion parameters). These models, developed by Meta AI are part of the LLaMA series, which are state-of-the-art transformer-based language models. The 8 billion and 70 billion parameter variants of LLaMA differ primarily in their scale and capacity. While both models leverage the same underlying architecture based on the Transformer model the larger 70B variant is more capable of understanding complex relationships in language due to its greater number of parameters.

3.3. Fine-tuning

Fine-tuning ASR models, like Whisper and Wav2Vec 2.0, involves adapting the pre-trained model to a specific dataset by continuing the training process on a smaller, domain-specific dataset. In our study, we used 80% of the data for fine-tuning and kept 20% for testing.

We set specific training arguments such as the learning rate, batch size, and the number of epochs, using the *TrainingArguments* package from the transformers library. In our case, learning rates are typically set in the range of 1e-5 to 4e-5 to avoid overfitting, while batch sizes are tuned based on the model and hardware limitations. We used Batch Gradient Descent by using a batch size of 8 or 16 or 32, depending on GPU capacity. We also used warm-up steps where the learning rate is gradually increased, save steps and eval steps were kept between 500 and 1000 according to how frequently the parameters were saved and evaluated. Number of training epochs was set to 30. We used regularization by setting the weight-decay as 0.005. *Gradient checkpointing* was kept *True* to reduce the memory requirements during the backpropagation phase of training.

During fine-tuning, it is common to freeze certain layers or parameters of the model, particularly those that capture general language knowledge. This helps speed up training and prevents overfitting on the smaller, domain-specific dataset. In our case, the feature encoder of the ASR model is typically frozen during fine-tuning. This component of the model processes raw audio input into latent speech representations. When fine-tuning the ASR model, the $model.freeze\_feature\_encoder()$ function freezes the parameters of the feature encoder, which means these layers are not updated during the training process. Fine-tuning focuses instead on the top transformer layers that map the latent representations to text outputs, allowing the model to specialize in a specific task or domain and not completely changing the pre-trained checkpoints.

3.4. Prompt Engineering

In our approach with LLAMA 3, various prompt engineering strategies were explored to optimize model performance. Initially, we experimented with Zero-Shot prompting. Simple generalized prompts were given, followed by enhancing the prompt, pushing it further towards the domain, and the type of input, using key phrases like “medical consultations”, “doctor-patient conversations” and “health-based discussions” to provide domain-specific contextual guidance.

In addition, we experimented with passing different chunk sizes to the LLMs. We processed one sentence at a time and compared this approach with larger 5-line and 10-line chunks to determine whether longer inputs helped the model grasp the context of medical conversations more effectively, especially in doctor-patient interactions.

4. Experimental Results

4.1. Dataset

Our research utilized the PriMock57 [20] dataset by Babylon Health, comprising 57 mock medical consultations totaling 9 hours of recorded speech. These consultations span diverse medical scenarios typical of clinical practice, with an average of 1500 spoken words per session. The dataset is balanced by gender between clinicians and patient actors, with participants aged 25-45 years. It includes various accents: clinicians primarily speak British English, while patients represent Indian and European dialects, reflecting the linguistic diversity of UK healthcare.

To simulate real-world clinical settings, we combined separate audio tracks for doctors and patients into a single file. This step was essential to capture the natural flow of medical dialogues and evaluate ASR performance in noisy healthcare environments.

4.2. Evaluation Metric

In this work, we use Word Error Rate (WER) as the primary evaluation metric to measure the performance of the transcription system. WER is a common metric used to assess the accuracy of ASR systems. It calculates the minimum number of word-level edits (insertions, deletions, and substitutions) required to transform the system’s transcription into the reference text.

The formula for WER is as follows:

$$\text{WER} = \frac{S + D + I}{N}$$

S : Number of substitutions

$$
\begin{aligned}
D &: \text{Number of deletions} \\
I &: \text{Number of insertions} \\
N &: \text{Total number of words in the reference text}
\end{aligned}
$$

4.3. Results and Comparison

Table 1 compares the Word Error Rate (WER) of several speech recognition models namely both versions of wav2vec2 and the small and large versions of whisper, both pre-trained and fine-tuned using the domain specific dataset.

Table 1
Effects of Fine-Tuning Pretrained ASR Models

Model Used	Fine-Tuning	WER
wav2vec2-base-960h	None	47.90
wav2vec2-large-960h	None	44.92
wav2vec2-base-finetuned	ft. using 80% data	29.70
whisper-small	None	36.70
whisper-large	None	34.70
whisper-small fine-tuned	ft. using 80% data	20.30

- The **wav2vec2-base-960h** model, without any fine-tuning, achieves a WER of 47.90, indicating moderate performance. Its larger variant, **wav2vec2-large-960h**, slightly improves this with a WER of 44.92, demonstrating the impact of increased model capacity on performance.
- Fine-tuning the **wav2vec2-base** model with 80% of the data leads to a significant improvement, reducing the WER to 29.70. This highlights the effectiveness of fine-tuning in enhancing the model's ability to generalize to the specific data it is trained on.
- **Whisper-small** and **whisper-large** models, which are not fine-tuned, achieve WERs of 36.70 and 34.70, respectively. The larger model benefits from greater capacity.
- Fine-tuning the **whisper-small** model using 80% of the data brings about the largest improvement in performance, reducing the WER to 20.30. This further emphasizes the importance of model fine-tuning for domain-specific tasks.

We have seen how finetuning enhances the model performance hugely. However, finetuning is dataset dependant and at times may force the model weights to overfit the training data. Let us now try to see the effects of using an LLM to postprocess the outputs provided by the ASR models.

We tested providing input as single sentences, chunks of n sentences, and all sentences together. Single sentences performed poorly, offering no improvement in ASR output. Chunks of 10–20 sentences performed better, with the best results at $n = 20$, as medical consultation context improves error correction. Beyond $n = 20$, performance declines. Due to LLAMA token limits, all sentences cannot be input at once.

Table 2 summarizes the performance of various automatic speech recognition (ASR) models based on their Word Error Rate after postprocessing the raw outputs using LLAMA 3 with Zero Shot Prompting.

Model Comparison:

- Each of the wav2vec2 models performs significantly better after the post processing step. Using the Zero Shot prompt, the wav2vec2 base model has a reduced WER of 35.5 from 47.90, which represents a reduction of 25.8% The wav2vec2 large model has an improved WER of 28.7 from

Table 2
Comparison of WER across Models with Zero Shot Prompt Post-Processing

Model Name	Variants	WER (after LLM Postprocessing)
Wav2vec 2.0	wav2vec2-base-960h	35.5
	wav2vec2-large-960h	28.7
	wav2vec2-base-finetuned	21.9
Whisper	whisper-small	36.3
	whisper-large	34.7
	whisper-small finetuned	22.7

44.92 which accounts for a reduction of 36.11%. The wav2vec2-base-finetuned model achieves the lowest WER across all models, with an improved score of 21.9 from 29.70. This suggests that LLM postprocessing after the fine-tuning process significantly enhances the model's ability to understand and transcribe spoken language accurately.

- Conversely, as of now, the whisper-small model exhibits the highest WER. As we see, whisper was producing much better results than wav2vec models, but the LLM post processing setup does not provide any improvement here. At times, it works adversely, showing that, although some errors are corrected by the LLM, it also changes many words that were originally correct. Also, whisper does correct most of the domain specific words and most of the errors are due to the informal nature of the consultations and filler words which are hard for the LLM to refine. Moreover, whisper produces a lot of punctuations which contribute to the WER as well.

Zero-Shot Prompt Example:

> You are a text refining model. There are medical consultation audios between doctors and patients and the transcribed text of the speech is your input. You are expected to use your advanced understanding of medical terminology, conversational context and sentence structure to refine the input text. Note that you just need to refine misspelt or inaccurate words according to its context. Do not make any grammatical changes. We need to calculate Word Error Rate, hence you are expected to refine a word but not change its position or generate anything new. Do not ask for confirmation. Do not give any reasoning or justification in the output, just the refined sentence is expected. If it is not possible to understand the context or meaning or language of the input sentence, just return the sentence as it is. Do not return empty sentences or make drastic changes.

5. Discussion and Conclusion

In this study, we extensively utilized pre-trained open-source ASR models, open-source LLM models, and their combinations. For transcribing domain-specific audio conversations, we observed that the best results were achieved by fine-tuning the Whisper ASR model. For wav2vec 2.0, the lowest Word Error Rate was obtained using a combined pipeline of fine-tuning followed by Zero Shot-prompted LLM postprocessing. The present study considers only Zero shot prompts. In future, few shot, chain of thought and several other promoting techniques need to be explored for further improving the performance of the proposed model.

There are certain limitations of this work. Firstly, fine-tuning is highly dataset-specific and can result in overfitting. Therefore, as our results indicate, if fine-tuning is not feasible, the most effective performance is achieved using the wav2vec 2.0 large version followed by Zero-Shot-prompted LLM postprocessing. Although Whisper Large significantly outperforms wav2vec 2.0 Large, this study has not yet observed the enhancement of Whisper's domain-specific transcripts using LLM postprocessing. On the contrary, we observed adverse effects when attempting to use LLMs to refine the Whisper transcripts. These findings underscore the need for further research aimed at reducing Word Error Rates more effectively and reliably.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly to rephrase and perform Grammar and spelling checks. After using these tools, the authors reviewed and edited the content as needed. The authors take full responsibility for the publication's content.

References

[1] F. Jelinek, Statistical methods for speech recognition, MIT press, 1998.

[2] M. Gales, S. Young, et al., The application of hidden markov models in speech recognition, Foundations and Trends® in Signal Processing 1 (2008) 195–304.

[3] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., The kaldi speech recognition toolkit, in: IEEE 2011 workshop on automatic speech recognition and understanding, IEEE Signal Processing Society, 2011.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal processing magazine 29 (2012) 82–97.

[5] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., Deep speech 2: End-to-end speech recognition in english and mandarin, in: International conference on machine learning, PMLR, 2016, pp. 173–182.

[6] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine learning 79 (2010) 151–175.

[7] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (1986) 533–536.

[8] L. R. Medsker, L. Jain, et al., Recurrent neural networks, Design and Applications 5 (2001) 2.

[9] A. Vaswani, Attention is all you need, Advances in Neural Information Processing Systems (2017).

[10] S. Hochreiter, Long short-term memory, Neural Computation MIT-Press (1997).

[11] A. Graves, A. Graves, Long short-term memory, Supervised sequence labelling with recurrent neural networks (2012) 37–45.

[12] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, Advances in neural information processing systems 33 (2020) 12449–12460.

[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision, in: International conference on machine learning, PMLR, 2023, pp. 28492–28518.

[14] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al., Conformer: Convolution-augmented transformer for speech recognition, arXiv preprint arXiv:2005.08100 (2020).

[15] L.-W. Chen, A. Rudnicky, Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023, pp. 1–5.

[16] Y. Liu, X. Yang, D. Qu, Exploration of whisper fine-tuning strategies for low-resource asr, EURASIP Journal on Audio, Speech, and Music Processing 2024 (2024) 29.

[17] T. B. Brown, Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).

[18] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[19] A. Adedeji, S. Joshi, B. Doohan, The sound of healthcare: Improving medical transcription asr accuracy with large language models, arXiv preprint arXiv:2402.07658 (2024).

[20] A. P. Korfiatis, F. Moramarco, R. Sarac, A. Savkov, Primock57: A dataset of primary care mock consultations, arXiv preprint arXiv:2204.00333 (2022).

A. Online Resources

- Primock 57 - Medical COnversation Dataset,
- Wav2vec2 - Pretrained ASR Model by facebook,
- Whisper - Pretrained ASR Model by OpenAI,
- LLAMA - Large Language Model