UCLA
UCLA Electronic Theses and Dissertations

Title
Fine-Tune Whisper and Transformer Large Language Model for Meeting Summarization

Permalink
https://escholarship.org/uc/item/7z96d1nh

Author
Ge, Fei

Publication Date
2024

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
Los Angeles

Fine-Tune Whisper and Transformer Large Language Model for Meeting Summarization

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics and Data Science

by

Fei Ge

2024

© Copyright by
Fei Ge
2024

ABSTRACT OF THE THESIS

Fine-Tune Whisper and Transformer Large Language Model
for Meeting Summarization

by

Fei Ge
Master of Applied Statistics and Data Science
University of California, Los Angeles, 2024
Professor Yingnian Wu, Chair

With globalization escalating, multinational companies frequently hold meetings involving both domestic and international employees. However, time zone differences often result in international employees missing some meetings. This thesis explores an innovative solution to address this issue and ensure that colleagues who miss meetings can quickly catch up on the content. The core of this solution involves fine-tuning the Whisper model to convert audio recordings of meetings to text, followed by advanced summary transformers based on fine-tuning Llama3 and specific prompts to summarize the converted text. The resulting summaries provide a concise and comprehensive overview of the meeting's content, which can then be distributed to employees who could not attend due to time zone constraints. This approach not only enhances the efficiency of work communication among colleagues but also optimizes the global management and operational efficiency of the company.

The thesis of Fei Ge is approved.

Hongquan Xu

Qing Zhou

Yingnian Wu, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

1 Introduction **1**

2 Methodology **3**

2.1 Transformer 3

2.2 Fine-tune 6

2.3 Whisper 7

2.3.1 Pre-training Whisper 7

2.3.2 Fine-tuning Whisper 10

2.4 Large Language Model 11

2.4.1 Pre-training Llama3 12

2.4.2 Fine-tuning Llama3 16

3 Experiment **19**

3.1 Dataset 19

3.1.1 Data Source 19

3.1.2 Validation Data 20

3.1.3 Training Data 20

3.1.4 Prediction Data 21

3.2 Hyper-Parameters 23

3.2.1 PC Information 23

3.2.2 Parameters 24

3.3 Metrics 24

3.3.1 fine-tuning Whisper 24
3.4 Results 25

4 Discussion 27
4.1 Conclusion 27
4.2 Limitation 28
4.2.1 Data Collection 28
4.2.2 Domain-Specific Knowledge 29
4.3 Future Work 29
4.4 Applications 31

References 32

LIST OF FIGURES

2.1 The Transformer Model Architecture[2] 4
2.2 Encoder-Decoder Network[12] 5
2.3 Transformer architecture and training objectives[8] 6
2.4 Fine Tuning[6] 7
2.5 Overview of Whisper approach[7] 9
2.6 Whisper model family[7] 10
2.7 New System Level[1] 15
2.8 Llama3 family 16
2.9 API example[10] 18

3.1 Validation Data 20
3.2 Training Data 21
3.3 Prediction Data (before fine-tune) 22
3.4 Prediction Data (after fine-tune) 23

4.1 the process of ORPO 30

LIST OF TABLES

2.1 Llama model version 13

3.1 CER and WER of Text Before and After Fine-tuning Whisper 25

CHAPTER 1

Introduction

With the development of economic globalization, multinational companies, particularly those between China and the United States, frequently hold meetings involving both domestic and international employees. However, time zone differences often result in international employees missing some meetings. Additionally, language barriers prevent some employees who are not fluent in Chinese or English from fully understanding the content of these meetings. The topic of ensuring effective communication in multinational companies is critical because it directly impacts operational efficiency and employee inclusion. Meetings are an important aspect of corporate communication, where important decisions and information are shared. When employees miss meetings or cannot understand the content, it leads to gaps in knowledge, reduced productivity, and potential misunderstandings.

This thesis focuses on developing a solution to convert and summarize meeting content, making it accessible to all employees regardless of time zone or language proficiency. The research explores the use of advanced Large Language Models, specifically fine-tuning Whisper to convert audio to text, and fine-tuning Llama3 with prompt engineering to generate concise summaries of the transcribed text. Ensuring that all employees have access to meeting content, regardless of their location or language skills, improves collaboration and operational effectiveness. This research contributes to the broader goal of optimizing global management practices and enhancing employee engagement.

Previous work in this area has primarily focused on either transcription or summarization individually, often lacking integration and customization for specific corporate needs. For ex-

ample, Ashish Vaswani [2] introduced the Transformer model, which significantly improved the capabilities of text processing tasks but did not specifically address the integration of transcription and summarization for corporate meetings. Similarly, Jacob Devlin [4] developed BERT, which advanced the state-of-the-art in text understanding but was not tailored for the real-time needs of multinational companies. This thesis builds on previous efforts by integrating transcription and summarization into a unified process tailored for multinational companies. By fine-tuning Whisper for accurate audio-to-text conversion and using advanced models like Llama3 for text summarization, this research offers a more comprehensive and effective solution. The innovation lies in the seamless integration of these models and the ability to provide clear, concise summaries that are easily understandable by employees who missed the meetings or face language barriers. This approach not only improves communication efficiency but also sets a new benchmark for future research in this domain.

CHAPTER 2

Methodology

2.1 Transformer

The Transformer model, introduced by Vaswani [2] , represents a significant advancement in the field of natural language processing (NLP) and has become the foundation for many state-of-the-art NLP models, as shown in Figure 2.1, among which Generative Pre-trained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT) are the most popular and well-known to the public. Unlike traditional models such as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), which process data sequentially, the Transformer processes entire sequences of data simultaneously. This parallel processing capability greatly improves computational efficiency and allows the model to better handle long-range dependencies within the data.The fundamental innovation of the Transformer is its self-attention mechanism, which allows the model to process and generate text by considering the relationships between all words in a sequence simultaneously, rather than sequentially as done in older models like RNNs and LSTMs.

The Transformer consists of an encoder-decoder architecture [12], as shown in Figure 2.2. The encoder processes the input sequence and generates a context-aware representation, while the decoder uses this representation to generate the output sequence. Key components of the Transformer include multi-head self-attention, which allows the model to focus on different parts of the input simultaneously, and position-wise feed-forward networks, which process the information at each position independently.

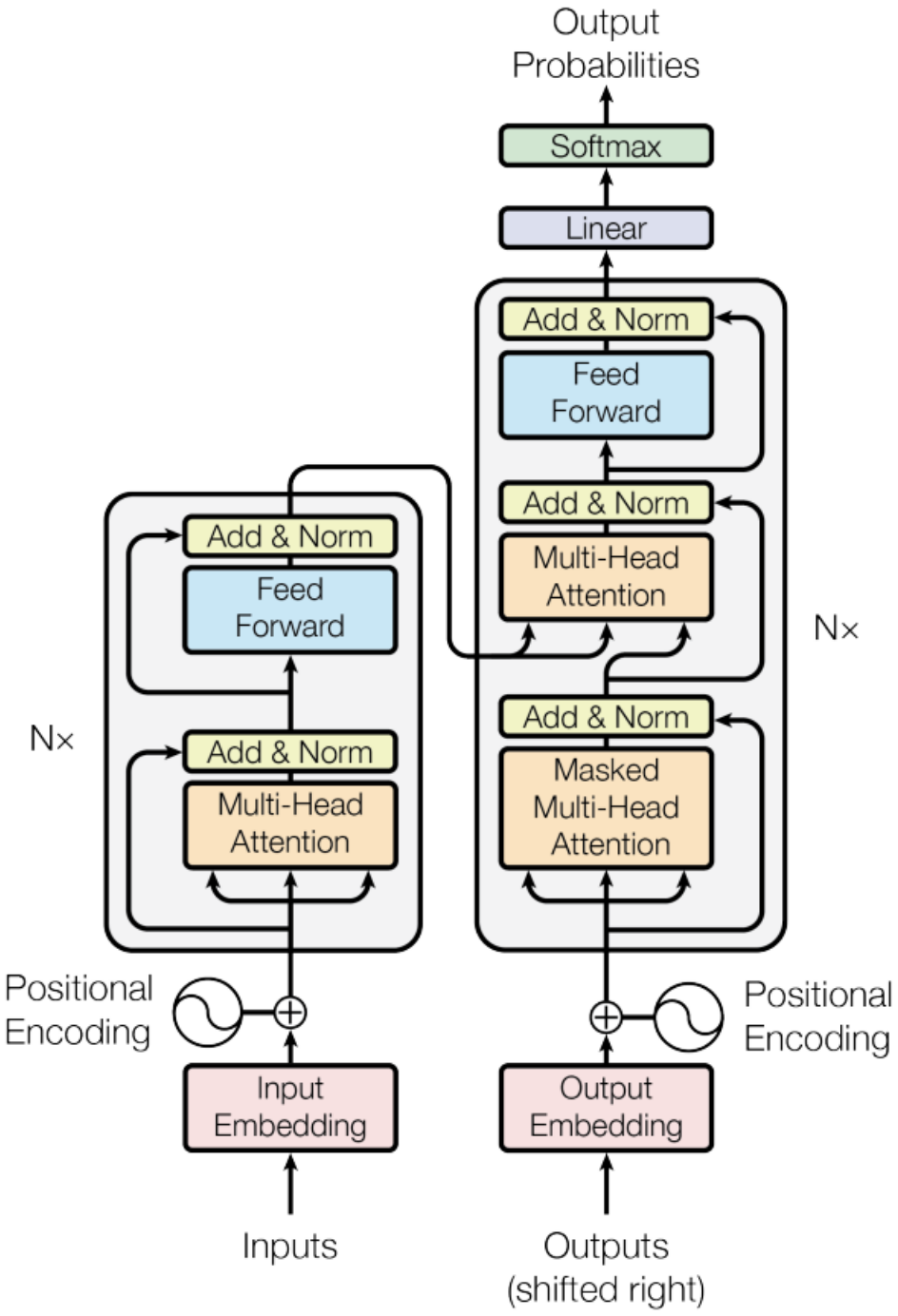

Figure 2.1: The Transformer Model Architecture[2]

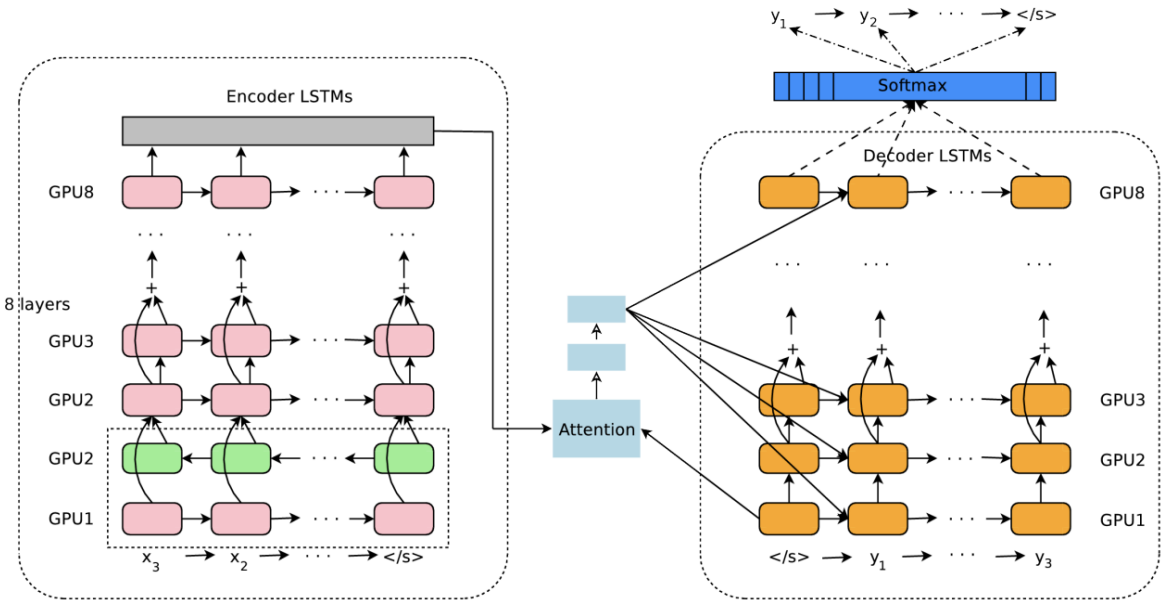

Figure 2.2: Encoder-Decoder Network[12]

The same primary components of all transformers can be tokenizers, a single embedding layer, transformer layer and unembedding layer (optional). Text is first converted into numerical representations called tokens, which are then transformed into vectors by using a word embedding table. In each layer of the model, tokens are contextualized with surrounding (unmasked) tokens within the context window through a parallel multi-head attention mechanism. This mechanism allows the model to highlight important tokens and reduce the influence of less significant ones. By doing so, the model can effectively capture the relationships and dependencies between words, improving its understanding and generation of natural language, as shown in Figure 2.3. The Whisper model and LLM mentioned below are both related to the Transformer model architecture [8], which will be discussed in detail in the following sections. Furthermore, in this thesis, the GPT API will also be utilized to optimize the outputs generated by the Llama3 model.

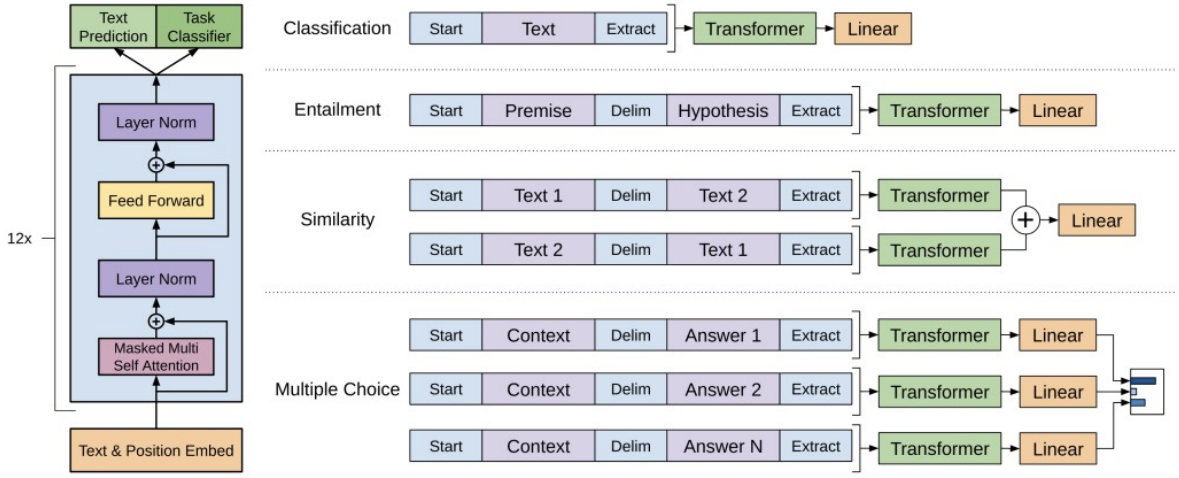

Figure 2.3: Transformer architecture and training objectives[8]

2.2 Fine-tune

Fine-tuning is a crucial process in machine learning that involves refining a pre-trained model to enhance its performance on a specific task or domain. In deep learning, fine-tuning is an approach to transfer learning in which the parameters of a pre-trained model are trained on new data [6]. This begins with a model that has already undergone extensive training on a large and diverse dataset, giving it a broad understanding of general features. The model is then further trained on a smaller, specialized dataset that is more representative of the specific application it will be used for. During this phase, the model's parameters are adjusted to better capture the patterns and nuances of the new data, such as domain-specific terminology and contextual usage. This targeted training helps the model to adapt its learned representations, thereby significantly improving its accuracy and effectiveness in the specialized task. Fine-tuning allows the model to leverage its broad initial training while gaining the specificity needed for optimal performance in particular real-world tasks, as shown in Figure 2.4. For example, fine-tuning a language model on legal documents can enhance its ability to understand and generate legal text, providing more accurate and relevant results for legal applications. This process is akin to the way humans refine their

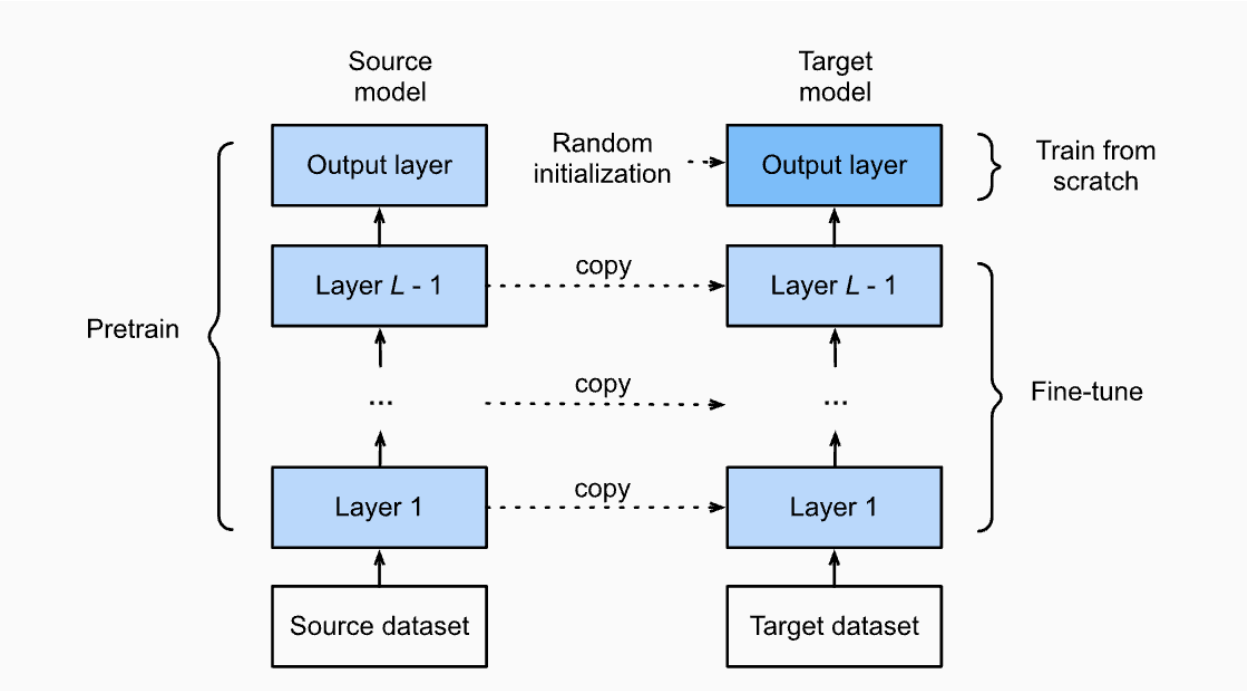

Figure 2.4: Fine Tuning[6]

skills in a specific area after gaining general knowledge [6].

2.3 Whisper

The whisper model is also fundamentally built upon the principles of the Transformer architecture. Whisper utilizes the Transformer model's encoder-decoder framework. This architecture allows the model to efficiently process sequential audio data by transforming it into a series of encoded representations, which are then decoded into textual output [2].

2.3.1 Pre-training Whisper

The Whisper model is an advanced speech recognition system developed by OpenAI and first released as open-source software in September 2022. This model is designed to convert spoken language into written text. Whisper is particularly notable for its ability to handle

a wide variety of languages and accents, making it highly versatile and effective for global applications. From now, the Whisper can handle over 90 languages, making it one of the most versatile and comprehensive speech recognition models available. This extensive language support is achieved through training on a diverse dataset that includes a broad spectrum of languages and dialects, allowing the model to generalize well across different linguistic contexts [7].

Moreover, a key feature of Whisper is its self-attention mechanism. This component allows the model to weigh the importance of different segments of the input audio, enabling it to focus on relevant parts of the speech. By dynamically adjusting these weights, the model can more accurately transcribe spoken language [2].

Firstly, the process begins with pre-processing the audio data. This involves converting the audio waveform into a detailed representation of the frequencies present over time, similar to an image that depicts sound intensity at various frequencies and times. This step is fundamental for transforming the audio into a format suitable for the model to process.

Secondly, this detailed audio representation is then divided into smaller segments, called tokens. Each token represents a short time frame of the audio signal. These tokens are also important for the model to process the audio in manageable pieces.

Thirdly, in the encoding phase, the Transformer's encoder processes these tokens to generate a series of encoded representations. This involves multiple layers of self-attention and feed-forward neural networks, which help capture the contextual information from the entire audio sequence. The encoded representations are then passed to the decoder. In the decoding phase, The decoder generates text output by predicting the next word in the sequence, using the context provided by the encoded audio tokens. This prediction process continues until the entire speech input is transcribed.

Lastly, the raw text output from the decoder may undergo post-processing to improve readability and accuracy. This can include correcting common transcription errors, adjusting

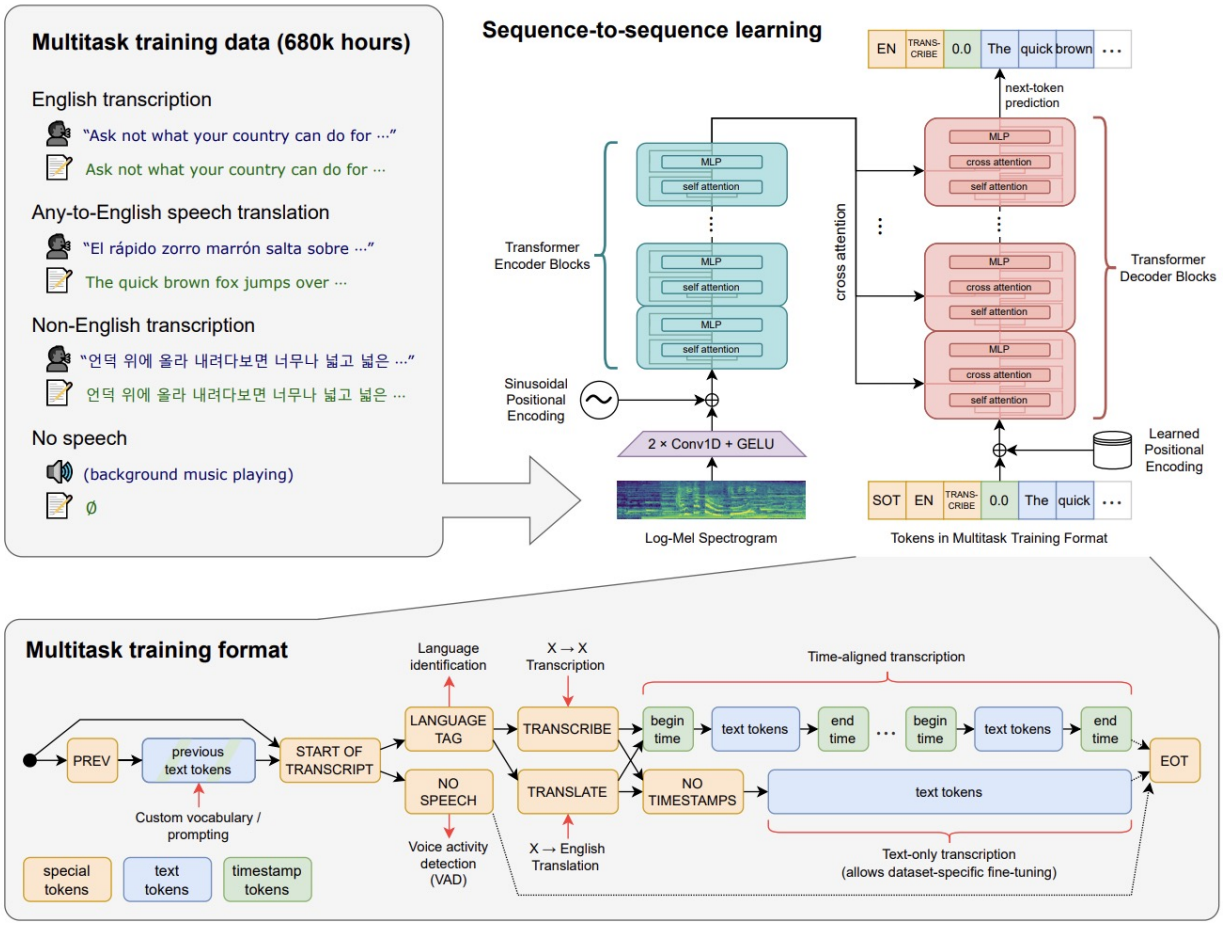

Figure 2.5: Overview of Whisper approach[7]

punctuation, and formatting the text according to specific requirements. And the process is shown in Figure 2.5.

The Whisper model comes in five different configurations, each with varying sizes and capacities. These configurations are designed to balance between performance and computational efficiency, allowing users to select a model that best fits their specific needs [7]. And the descriptions of Whisper model family are shown in Figure 2.6.

In this thesis, based on our data, requirements and time, we utilize the Whisper small model. For Whisper small model, the model consists of 12 layers. Each layer includes attention mechanisms and feed-forward neural networks that process the input data and

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Figure 2.6: Whisper model family[7]

capture complex patterns within the speech. The width of this model is 768, indicating the dimensionality of the hidden representations. This width determines the size of the vectors that are used to represent the speech data at each layer. And also this model employs 12 attention heads. Each head operates independently to focus on different parts of the input data, allowing the model to capture a wide range of linguistic features and dependencies. Moreover, the Whisper small model has 244 million parameters. These parameters are learned during the training process and are used to transform the input data into accurate transcriptions [7].

2.3.2 Fine-tuning Whisper

Fine-tuning Whisper involves taking the pre-trained Whisper model and further training it on a specific dataset tailored to a particular application or domain. This process adjusts the model's parameters to better suit the specific requirements of a given task, enhancing its performance in that specific context. Fine-tuning involves training the Whisper model further on a more specific and focused dataset that is relevant to a particular task or domain. This could include industry-specific jargon, specialized accents, or context-specific dialogues. By tailoring the model to the particularities of the task at hand, fine-tuning improves the

model's accuracy and effectiveness in that specific area.

Fine-tuning Whisper enhances its ability to understand and transcribe domain-specific language accurately. The model becomes more relevant and useful for specific applications, such as transcribing and summarizing multinational company meetings. Improves overall efficiency in processing and understanding meeting content, leading to better communication and collaboration within the company.

In this thesis, the reason for utilizing a fine-tuned Whisper model is due to the diverse accents and speaking habits of individuals in meetings. For instance, participants have different ways of pausing and varying speech speeds. This is particularly relevant in multinational companies between China and the United States, where employees often mix Chinese and English during meetings, each with distinct accents and pronunciations. Additionally, there are company-specific terminologies that need to be accurately recognized. To improve the accuracy of converting meeting audio to text and to prevent misinterpretations of specialized terms that could alter the meaning of the meetings, we will use a large amount of data to fine-tune the Whisper model. This fine-tuning process aims to enhance transcription accuracy by training the model on domain-specific data.

2.4 Large Language Model

A large language model (LLM) is an advanced type of artificial intelligence designed to understand and generate human language. These models are typically trained on vast amounts of text data, which allows them to learn the complexities and nuances of language, including grammar, syntax, semantics, and even some level of context and reasoning. The primary benefit of large language models is their ability to generate human-like text and understand complex language tasks, making them valuable tools in fields like customer service, content creation, and data analysis.

A large language model (LLM) is an advanced type of artificial intelligence designed

to understand and generate human language. These models are typically trained on vast amounts of text data, which allows them to learn the complexities and nuances of language, including grammar, syntax, semantics, and even some level of context and reasoning. These models are trained on extensive and diverse datasets that include a wide range of text sources such as books, articles, websites, and more [3]. This diversity helps the model to generalize well across different types of text and applications, making it versatile for various NLP tasks. Most LLMs use the Transformer architecture, which relies on self-attention mechanisms to process input data. This architecture allows the model to handle long-range dependencies and relationships within the text, providing a more coherent and contextually relevant output. Well-known models like GPT, BERT, and Llama are all large, transformer-based language models.

2.4.1 Pre-training Llama3

Large Language Model Meta AI (Llama) is a family of autoregressive large language models released by Meta AI starting in February 2023 [11] and a state-of-the-art large language model developed to perform a variety of natural language processing (NLP) tasks. Similar to other advanced language models, Llama leverages the Transformer architecture to understand and generate human language effectively. Llama uses the Transformer architecture, which includes self-attention mechanisms that allow the model to process and generate text by understanding the context and relationships between words. Llama's ability to generate coherent and contextually relevant text makes it a powerful tool for various applications. The model is pre-trained on a vast corpus of text data, which helps it learn the general structure and nuances of language. After pre-training, Llama can be fine-tuned on specific datasets to adapt it to particular tasks or domains, enhancing its performance for specific applications [9].

In this thesis, the latest version of the model, Llama3, released in April 2024, has been chosen. And all the information of Llama model is listed in the Table 2.1. Llama3 represents

Table 2.1: Llama model version

Model	Release Date	Context Length	Corpus Size
LLaMA	Feburary 24, 2023	2048	1 - 1.4 T
Llama2	July 18, 2023	4096	2 T
Llama3	April 18, 2024	8912	15 T

a significant advancement over its predecessors, Llama1 and Llama2, with improvements in architecture, performance, and capabilities.

Llama3 marks the significant progression from its predecessors, Llama1 and Llama2, through a series of enhancements in architecture, scalability, training data, and performance. Initially, Llama1 utilized the basic Transformer framework with conventional self-attention and feed-forward layers, making it effective for general language tasks but limited in handling more complex linguistic structures. Llama2 improved upon this by increasing the depth and number of parameters, enabling a better grasp of sophisticated language patterns and contexts. Llama3 takes these advancements further by integrating state-of-the-art techniques such as advanced attention mechanisms, refined layer normalization, and optimized feed-forward layers, greatly improving its capability to manage long-range dependencies and intricate language constructs.

In terms of parameter scaling, Llama1 had a moderate parameter count, while Llama2 significantly expanded this, enhancing its contextual understanding and language generation capabilities. Llama3 escalates this further, with billions of parameters that allow it to excel in a broad range of NLP tasks by capturing subtle nuances and complexities of human language more effectively [11]. The training data and methodologies have also evolved; Llama1 was trained on a substantial but somewhat limited dataset, Llama2 used a more extensive and diverse set, and Llama3 leverages an even larger and more varied corpus, employing sophisticated training techniques like curriculum learning and robust data augmentation. This comprehensive training ensures better generalization and adaptation to diverse languages

and specific domains.

Performance and efficiency have consistently improved with each version. Llama1 offered solid baseline performance, Llama2 enhanced efficiency and broadened application scope, while Llama3 delivers top-tier performance optimized for accuracy and speed, making it highly suitable for real-time applications and large-scale deployments. The applications for each version have also expanded: Llama1 was primarily used for general text generation and basic language tasks, Llama2 extended its capabilities to more complex tasks such as advanced text analysis and detailed content creation, and Llama3 excels in domains requiring deep contextual understanding, precise language generation, and complex interactive tasks, making it ideal for specialized areas like legal and medical text processing.

The Llama3 models are designed to be highly effective while ensuring a responsible deployment approach through a system-level strategy for their development and application. These models are part of a broader system, allowing developers to tailor them to specific goals. Instruction fine-tuning plays a crucial role in ensuring model safety, with rigorous safety testing, or "red-teaming," involving human experts and automated methods to create adversarial prompts and assess risks, including Chemical, Biological, and Cyber Security threats. The insights from these tests inform the iterative safety fine-tuning of the models. Llama Guard models serve as a foundation for prompt and response safety and can be fine-tuned for specific applications, with Llama Guard 2 adopting the MLCommons taxonomy to support industry standards. CyberSecEval 2 assesses vulnerabilities related to code interpreter and cybersecurity, while Code Shield introduces real-time filtering of insecure code generated by LLMs. An open approach to generative AI is emphasized to unify the ecosystem and mitigate potential harms, supported by an updated Responsible Use Guide (RUG) that provides comprehensive guidelines for responsible LLM development, recommending thorough checking and filtering of all inputs and outputs and the use of content moderation APIs and tools from cloud service providers [1], as shown in Figure 2.7.

The Llama 3 instruction tuned models, a collection of pretrained and instruction tuned

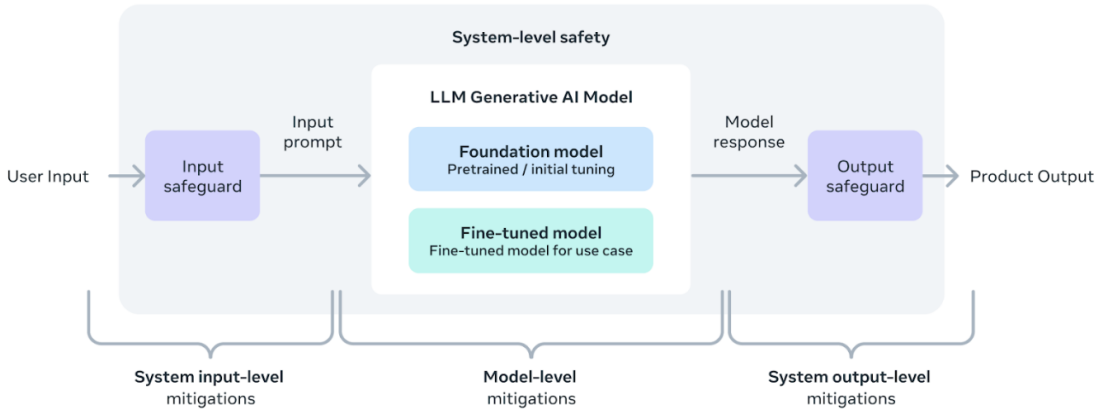

Figure 2.7: New System Level[1]

generative text models in 8 and 70B sizes, are optimized for dialogue use cases and outperform many of the available open source chat models on common industry benchmarks. It comes in two sizes — 8B and 70B parameters — in pre-trained and instruction tuned variants, as shown in Figure 2.8. Llama 3 is an auto-regressive language model that uses an optimized transformer architecture. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences for helpfulness and safety. And also token counts refer to pretraining data only. Both the 8 and 70B versions use Grouped-Query Attention (GQA) for improved inference scalability [1].

In this thesis, based on the computational capabilities of the GPU and processing speed, the Llama3 8B model has been selected for text summarization tasks. This model strikes a balance between performance and efficiency, making it well-suited for handling large-scale natural language processing tasks with high accuracy and speed. The choice of Llama3 8B ensures that the summarization process can leverage advanced features of the model while staying within the computational limits provided by the available hardware. This selection allows for effective and efficient summarization, meeting the requirements of the study while optimizing the use of resources.

	Training Data	Params	Context length	GQA	Token count	Knowledge cutoff
Llama 3	A new mix of publicly available online data.	8B	8k	Yes	15T+	March, 2023
		70B	8k	Yes		December, 2023

Figure 2.8: Llama3 family

2.4.2 Fine-tuning Llama3

Fine-tuning Llama3 involves adapting a pre-trained language model to a specific task or domain by further training it on a relevant dataset. This process starts with Llama3, which has already been trained on a broad corpus of text data, giving it a wide-ranging understanding of language. Fine-tuning tailors this general knowledge to more specific applications, enhancing the model's performance in those areas. Full parameter fine-tuning is a method that fine-tunes all the parameters of all the layers of the pre-trained model. In general, it can achieve the best performance but it is also the most resource-intensive and time consuming: it requires most GPU resources and takes the longest.

In this research, due to time constraints and the superior performance of the GPT model in text summarization, the decision was made to utilize the GPT Application Programming Iterface (API) for fine-tuning summary texts. This approach aims to enhance the output of Llama3 by leveraging the advanced capabilities of the GPT model, ensuring more accurate and coherent summaries. The integration of GPT's summarization prowess with Llama3's capabilities provides an optimized solution for generating high-quality summaries efficiently.

Application Programming Interface, is a set of protocols and tools that allows different

software applications to communicate with each other. It defines the methods and data formats that applications can use to request and exchange information [10]. And in Figure 2.9, it shows the example application dependent on APIs from three libraries. The GPT API belongs to the category of RESTful APIs, which stands for Representational State Transfer. RESTful APIs are designed to work over HTTP and are commonly used for web services due to their simplicity and scalability.

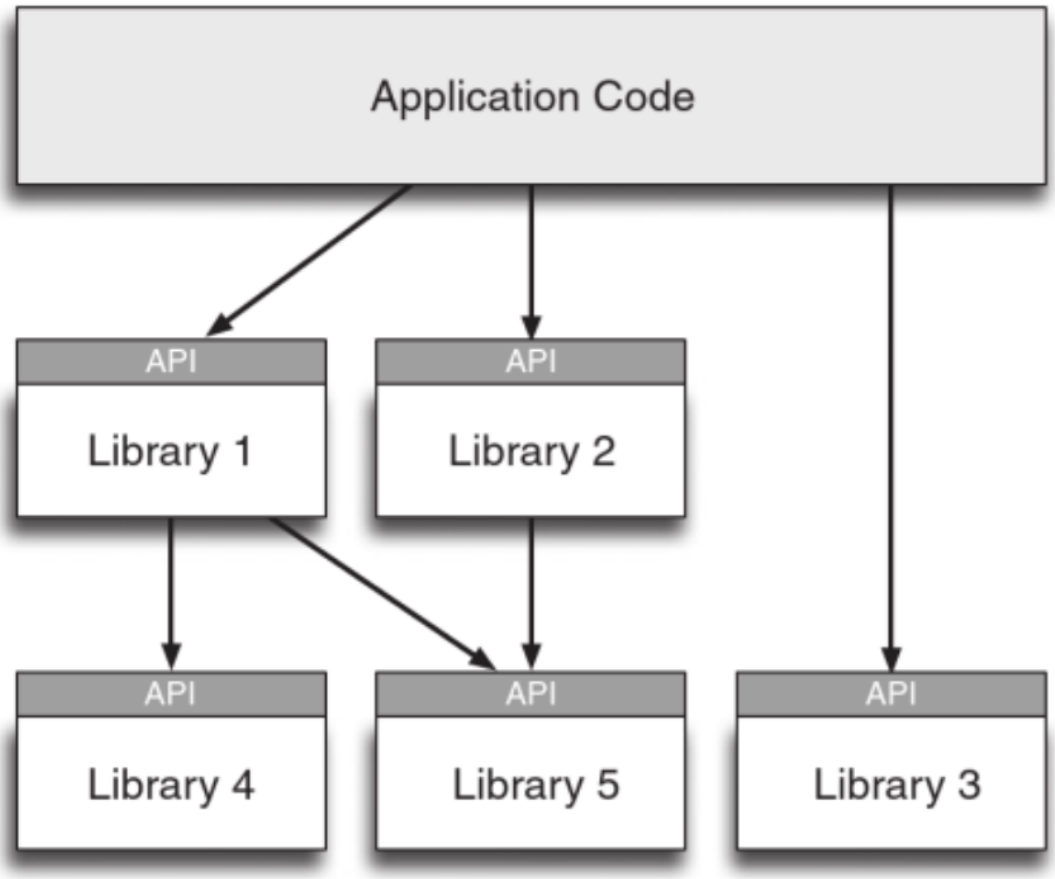

Figure 2.9: API example[10]

CHAPTER 3

Experiment

3.1 Dataset

This experiment uses a dataset composed of audio recordings from company meetings, featuring a mix of English and Chinese. The audio dataset is preprocessed and stored in JSON format for subsequent model training and evaluation.

3.1.1 Data Source

The audio data comes from 30-minute recordings of company meetings, which include a mixture of English and Chinese speech. These recordings capture various discussion topics and natural speech patterns in a professional setting. The dataset includes 30-minute audio recordings in formats such as WAV and M4A. Each recording contains mixed-language dialogue, with varying lengths of speech segments.

First, the audio files are processed using Whisper to generate outputs with timestamps and IDs, referred to as "origin text." Next, the audio is manually reviewed and edited against the "origin text" to create the "text," which serves as our validation data. Based on the segmentation timestamps and IDs, the audio is then divided into segments. In this experiment, we segmented the audio into 700 parts, each labeled as "a+id."

Figure 3.1: Validation Data

3.1.2 Validation Data

For the validation text, it is stored in JSON file format, with columns including: id, start time, end time, and text, as shown in Figure 3.1.

3.1.3 Training Data

When fine-tuning Whisper, 80% of the validation data is used as training text, paired with the corresponding audio segments, to train the fine-tuned Whisper model. During this process, the dataset is converted into a list and stored in JSON format, with columns including: id, audio path and validation text, as shown in Figure 3.2. And the training rate is set to 0.8.

Figure 3.2: Training Data

3.1.4 Prediction Data

In this thesis, prediction data is divided into two main categories: output data before fine-tuning and output data after fine-tuning.

3.1.4.1 Before Fine-Tuning

In this study, due to the time constraints for fine-tuning, we selected the Whisper-small model. After setting certain parameters and running the model, the output text constitutes the prediction data before fine-tuning, as shown in Figure 3.3.

3.1.4.2 After Fine-Tuning

In the fine-tuned model based on Whisper-small, the output text generated by running the model with the same parameters as mentioned above constitutes the prediction data after

Figure 3.3: Prediction Data (before fine-tune)

Figure 3.4: Prediction Data (after fine-tune)

fine-tuning, which serves as the final output data in this study.

3.2 Hyper-Parameters

3.2.1 PC Information

In this research, the computer configuration for running the model is as follows:

1. **CPU**: 12th Gen Intel(R) Core(TM) i7-12700K
2. **GPU**: NVIDIA Corporation GA102GL [RTX A6000]
3. **Memory**: 2x32GB DDR5 4800MHz
4. **Storage**: NVMe M.2 SSD 2TB

3.2.2 Parameters

In this study, the model parameters are set as follows: The temperature for sampling is set to 0.15, with an increment of 0.2 in case of fallback when decoding fails to meet specified thresholds. The model generates the best of 5 candidates during sampling with non-zero temperature, while using a beam size of 8 for beam search when the temperature is zero. Patience is set to 1.0, equivalent to conventional beam search, and the length penalty coefficient is -0.05, applying simple length normalization. Tokens to be suppressed during sampling are specified by a comma-separated list, with "-1" suppressing most special characters except common punctuations. An optional initial prompt can be provided for the first window, and the model conditions on previous text to ensure consistency. Inference is performed in fp16, and decoding is considered failed if the gzip compression ratio exceeds 2.4 or the average log probability is below -1.0. Additionally, if the probability of the silence token is higher than 0.6 and decoding fails due to the logprob threshold, the segment is treated as silence.

3.3 Metrics

3.3.1 fine-tuning Whisper

To evaluate the performance of fine-tuning Whisper model in this thesis, several key metrics will be employed:

1. **Word Error Rate(WER)** WER measures the difference between the transcribed text generated by the ASR system and the reference (correct) text. WER is calculated as the ratio of the total number of errors (insertions, deletions, and substitutions) to the total number of words in the reference text. The formula for WER is:

   WER = $\frac{S+D+I}{N_1}$, where S is the number of substitutions, D is the number of deletions, I is the number of insertions, N_1 is the total number of **words** in the reference text.

Table 3.1: CER and WER of Text Before and After Fine-tuning Whisper

Model Version	**Text Version**	**Word Error Rate**	**Character Error Rate**
Whisper-small	Before Fine-Tuning	0.8510	0.1944
Whisper-small	After Fine-Tuning	0.9779	0.1853
Whisper-large-v2	Before Fine-Tuning	0.6468	0.0930
Whisper-large-v3	Before Fine-Tuning	0.9658	0.1737

2. **Character Error Rate(CER)** CER measures the number of incorrect characters in the transcribed text compared to the reference text. It accounts for insertions, deletions, and substitutions of characters. The formula for CER is:

   CER = $\frac{S+D+I}{N_2}$, where S is the number of substitutions, D is the number of deletions, I is the number of insertions, N_1 is the total number of **characters** in the reference text.

3.4 Results

The results of the study are presented in Table 3.1, which shows the Character Error Rate (CER) and Word Error Rate (WER) of text before and after fine-tuning the Whisper models.

1. Whisper-small Model:
   (a) Before Fine-Tuning: The WER was 0.8510, and the CER was 0.1944.
   (b) After Fine-Tuning: The WER increased to 0.9779, while the CER slightly decreased to 0.1853.
2. Whisper-large-v2 Model:
   (a) Before Fine-Tuning: The WER was 0.6468, and the CER was 0.0930.
3. Whisper-large-v3 Model:

(a) Before Fine-Tuning: The WER was 0.9658, and the CER was 0.1737.

Given that our audio data is a mix of Chinese and English, with a larger proportion of Chinese, the CER results are more indicative of the model's performance. The increase in WER after fine-tuning the Whisper-small model is understandable in this context. The best results were achieved by the Whisper-large-v2 model, which had the lowest WER and CER before fine-tuning, indicating superior accuracy.

However, due to our computer's configuration, fine-tuning the Whisper-large-v2 model was impractical because it required excessive memory and took too long to run. Consequently, we chose to work with the Whisper-small model. Despite fine-tuning, the CER did not decrease significantly, which can be attributed to the limited size of our training dataset.

These results highlight the importance of model selection and the constraints imposed by computational resources. While the Whisper-large-v2 model demonstrated the best performance, practical limitations necessitated the use of the Whisper-small model, emphasizing the need for a balance between model capability and available resources. Further improvements could be achieved with larger and more diverse training datasets and better computational resources.

CHAPTER 4

Discussion

4.1 Conclusion

In this study, we explored the performance of Whisper models for transcribing mixed Chinese and English audio data, focusing on the Whisper-small and Whisper-large versions. Due to computational constraints, we selected the Whisper-small model for fine-tuning, despite the Whisper-large-v2 model showing superior initial performance. Our findings indicate that while fine-tuning the Whisper-small model led to a slight improvement in Character Error Rate (CER), it also resulted in an increased Word Error Rate (WER), reflecting the challenges of fine-tuning with a limited dataset.

The results highlight the importance of model size and computational resources in achieving high transcription accuracy. The Whisper-large-v2 model, with its significantly lower CER and WER, demonstrated the potential benefits of using larger models. However, practical limitations such as memory usage and processing time necessitated the use of the Whisper-small model.

Given that our dataset consisted primarily of Chinese, the CER was a more relevant metric for assessing performance. The limited improvement in CER after fine-tuning underscores the need for larger and more representative training datasets to better capture the nuances of mixed-language audio.

Future work should focus on acquiring more extensive and diverse datasets, optimizing fine-tuning techniques, and exploring ways to leverage more powerful computational re-

sources. These steps will help to enhance the model's performance, making it more effective for real-world applications where accuracy and efficiency are paramount.

4.2 Limitation

Even though the Whisper model and Llama3 are currently among the most accurate models, there are still the following limitations due to time constraints:

4.2.1 Data Collection

Collecting data for training requires a substantial amount of audio data, along with manually prepared validation text corresponding to each audio segment. Due to the variability in accents among different speakers, it's necessary to gather extensive audio data from each individual to achieve higher accuracy and reduce the model's error rate. This comprehensive dataset helps the model learn the unique speech patterns and nuances of various speakers, which is crucial for achieving higher accuracy. This process is time-consuming and resource-intensive but is necessary to reduce the model's error rate and enhance its overall performance and reliability. This ensures the model can effectively learn the distinct speech patterns of various speakers, thereby improving its overall performance and reliability.

For this thesis, because the data comes from internal company meetings, ensuring data privacy and security is paramount. This restricts the availability of public meeting data that can be used, limiting the dataset and consequently contributing to a higher error rate and lower accuracy in the model.

Additionally, manual annotation and labeling of data for supervised learning are time-consuming and prone to errors. Incomplete or inaccurately labeled data can impair the fine-tuning process and degrade the performance of the models. Moreover, since our audio data contains a mix of Chinese and English, the performance of Whisper and Llama3 in handling and translating the Chinese portions has not been particularly impressive.

4.2.2 Domain-Specific Knowledge

Domain-specific knowledge refers to the specialized understanding needed to accurately interpret and generate content within specific fields such as legal, medical, or technical domains. Models like Whisper and Llama3 must be trained on domain-specific data to handle complex terminologies and nuances unique to these areas. Without adequate and diverse training data, these models may produce inaccurate or incomplete outputs, leading to misunderstandings or errors. Regular updates and re-training are necessary to keep up with the evolving knowledge in these fields. This process requires ongoing data collection, careful curation of training datasets, and guidance from domain experts to ensure the models provide reliable and contextually accurate outputs.

4.3 Future Work

Future work for this project will focus on several key areas to enhance the performance and applicability of Whisper and Llama3 models. First, expanding the dataset with more diverse and representative audio samples from various accents and dialects will be crucial. This will involve collecting additional high-quality recordings and corresponding transcriptions to improve the models' ability to handle a wide range of speech patterns and languages, particularly in multilingual and multinational contexts.

Second, addressing the limitations in data privacy and security will be a priority. Developing methods to anonymize sensitive data while maintaining its utility for training will help increase the amount of usable data from internal meetings without compromising privacy. This may involve collaboration with experts in data security and privacy-preserving machine learning techniques.

Third, improving the interpretability and transparency of the models will be essential for building trust and ensuring responsible deployment. This could include integrating advanced interpretability tools, refining model documentation, and creating user-friendly interfaces

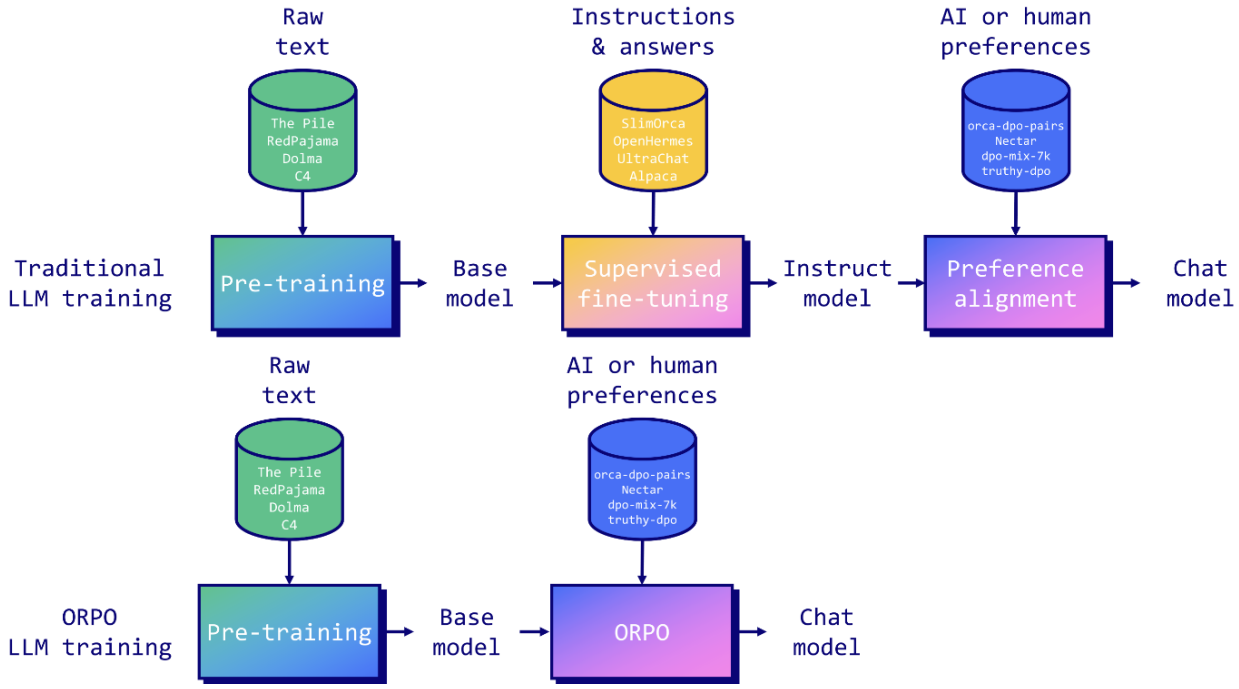

Figure 4.1: the process of ORPO

that allow stakeholders to better understand how decisions are made by the models.

Fourth, domain-specific fine-tuning will be further explored. This involves not only gathering more specialized datasets but also developing techniques for continuous learning, allowing models to stay updated with the latest developments in their respective fields. Collaboration with domain experts will be essential to ensure the models are accurately capturing and generating relevant information.

Finally, exploring hybrid approaches that combine the strengths of Whisper and Llama3 with other tools, such as the fine-tuning Llama3 with and ORPO [5] and Q-Lora for output optimization, could lead to even more robust and versatile solutions, and the process of ORPO has been shown in Figure 4.1. These efforts will be guided by ongoing evaluations and feedback, ensuring that the models are continually refined and improved to meet the evolving needs of users and applications.

4.4 Applications

The Whisper and Llama3 models offer significant potential across various applications. In multinational companies, these models can transcribe and summarize meetings accurately, overcoming language barriers and time zone differences to ensure all employees have access to crucial information. In healthcare, they can assist in transcribing medical consultations and generating concise patient summaries, improving communication and record-keeping. In the legal field, these models can transcribe court proceedings and summarize legal documents, enhancing efficiency and accessibility. Additionally, they can be used in customer service to provide real-time transcriptions and summaries of interactions, improving service quality and response times. By fine-tuning these models for specific domains, their application can be tailored to meet the unique needs of different industries, driving better outcomes and efficiencies.

REFERENCES

[1] Meta AI. Introducing meta llama 3: The most capable openly available llm to date. 2024.

[2] Niki Parmar Jakob Uszkoreit Llion Jones-Aidan N. Gomez Lukasz Kaiser Ashish Vaswani, Noam Shazeer and Illia Polosukhin. Attention is all you need. 2017.

[3] Mann B. Ryder N. Subbiah M. Kaplan-J. Dhariwal P. ... Amodei D. Brown, T. B. Language models are few-shot learners. 2020.

[4] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[5] Maxime Labonne. Fine-tune llama 3 with orpo. 2024.

[6] Joanne Quinn. Dive into deep learning: tools for engagement. 2022.

[7] Jong Wook; Xu Tao; Brockman Greg; McLeavey-Christine; Sutskever Ilya Radford, Alec; Kim. Robust speech recognition via large-scale weak supervision. 2022.

[8] Karthik; Salimans Tim; Sutskever Ilya Radford, Alec; Narasimhan. Improving language understanding by generative pre-training. 2018.

[9] Wu J. Child R. Luan D. Amodei D.- Sutskever I Radford, A. Language models are unsupervised multitask learners. 2019.

[10] Martin Reddy. Api design for c++. 2011.

[11] Thibaut; Izacard Gautier; Martinet Xavier; Lachaux Marie-Anne; Lacroix Timothée; Rozière Baptiste; Goyal Naman; Hambro Eric; Azhar Faisal; Rodriguez Aurelien; Joulin Armand; Grave Edouard; Lample Guillaume Touvron, Hugo; Lavril. Llama: Open and efficient foundation language models. 2023.

[12] Zhifeng Chen Quoc V Le Mohammad Norouzi Wolfgang Macherey Maxim Krikun Yuan Cao Qin Gao Klaus Macherey et al. Yonghui Wu, Mike Schuster. Google's neural machine translation system: Bridging the gap between human and machine translation. 2016.