### 國立臺灣大學電機資訊學院資訊工程研究所

### 碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

基於大型語言模型自非結構化文件建立微調資料集 Leverage Large Language Model Creating Fine-tune Dataset from Unstructured Data

吳葦誠

Wei-Cheng Wu

指導教授: 周承復 博士

Advisor: Chen-Fu Chou, Ph.D.

中華民國 113 年 7 月 July, 2024

## 國立臺灣大學碩士學位論文 口試委員會審定書 MASTER'S THESIS ACCEPTANCE CERTIFICATE NATIONAL TAIWAN UNIVERSITY

基於大型語言模型自非結構化文件建立微調資料集

Leverage Large Language Model Creating Fine-tune Dataset from Unstructured Data

本論文係<u>吳葦誠</u>君(學號 R11922164)在國立臺灣大學資訊工程學系完成之碩士學位論文,於民國 113 年 7 月 16 日承下列考試委員審查通過及口試及格,特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 16 July 2024 have examined a Master's thesis entitled above presented by WU, WEI-CHENG (student ID: R11922164) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

自身投 Advisor) 一多時支養各學 多托女

系主任/所長 Director:





### **Acknowledgements**

最一開始,我要先由衷的感謝我的指導教授周承復老師,讓我在碩士的時候 有機會接下台大醫院耳鼻喉科的合作計畫,從而發想出現在的碩士論文題目,也 謝謝老師在每次 meeting 的時候深入與我討論報告的論文內容並且教會我所不足 的地方以及做研究的精神。

再來是要感謝在做研究的路上,開導研究方向最多的楊大煒學長,我的碩士研究 過程中都是他帶領我去突破自己的。同樣也要感謝的博士班學長還有呂羿賢學長 跟廖家緯學長,他們都也分別在研究的路上給予我非常多的建議以及回饋。

此外,我還要感謝的一定還有N組的同學們吳佑衽、林鼎鈞、施君諺、高涵毅以及游凱翔以及其他CMLAB的同學們。感謝大家一起在研究所的期間互相幫忙,不管是修課還是做研究上,大家都曾給予過非常多的幫忙。很慶幸在研究所有大家一起度過。

謝謝台大醫院計畫主持人吳振吉醫師在每次開會都可以灌入新的想法帶領研究方向,更要謝謝李彥輝醫師,儘管他在最後階段先行離開計畫,但在計畫前期他絕對幫上非常大的幫助,與他合作計劃節奏非常好。

感謝家人們以及我的女朋友,這一路上他們的支持,不管遇到什麼問題他們都在 後面默默支持給予我自信,讓我順利的完成我的碩士學位以及碩士倫文。

最後感謝所有在這一路上曾經幫助我的所有人,在此致以我最大的感謝。





## 摘要

隨著最近兩年大型語言模型 (LLM) 在這兩年爆炸性的成長,許多領域都開始積極地導入 LLM 希望借助人工智慧 (AI) 的力量改變自己以及增強競爭力,當然醫學相關的領域也不例外。儘管外界都覺得 LLM 非常強大,但是 LLM 存在一些問題,例如要如何使用少量的資料讓 LLM 學習預訓練資料內並沒有的資訊。如果在 LLM 的預訓練資料中,相關的知識並不存在,LLM 時常會非常有自信的回答出錯誤的資訊,稱之為幻覺,反而造成誤導,也是一個需要被重視的問題。

為了應對上述的狀況,我們提出了一個架構,借助僅有一般對話能力的 LLM 的力量將非結構化的專業資料轉變成能夠用來微調出一個專業領域問答 LLM。其中我們的架構之中將涉及數個一般對話能力的 LLM,依照使用順需依序是用來產生問題的 Generator LLM、產生對應答案的 Generator LLM、用來評價產生之問題與答案的 Critique LLM 以及最後要被微調成專業領域問答的 Final LLM。在訓練完 Final LLM 之後,我們再搭配 RAG 系統使用來檢索相關文件來針對問題來回答,並且搭配 Prompt Engineering 手法讓 Final LLM 能夠如預期中的回答出正確以及專業的知識。

實驗中的專業領域知識文件我們與台大醫院合作,使用醫生在耳科遺傳性聽力損失相關出版文獻以及研究做為專業研究領域文件。最後以此 LLM 協助未來醫生以及求醫的患者能夠更好的認識疾病增加問診效率。實驗評估的部分則是由

台大醫院的專門的耳科醫師提出經常由問診期間病人提出之問題座位測試集。評分則是由人工評分回答問題的品質、完整性以及是否出現幻覺作為評分依據。

關鍵字:大型語言模型、檢索增強生成、微調、資料集、耳科、特定領域、遺傳 性聽損



### **Abstract**

With the explosive growth of large language models (LLMs) over the past two years, many fields have actively integrated LLMs, hoping to leverage the power of artificial intelligence (AI) to transform themselves and enhance their competitiveness. The medical field is no exception. Although LLMs are perceived as very powerful, they have certain issues, such as how to enable learning from a small amount of data about information not contained in their pre-training data. If relevant knowledge is absent in an LLM's pre-training data, it often confidently provides incorrect information, a phenomenon known as "hallucination," which can be misleading and requires attention.

To address the aforementioned issues, we propose a framework that uses LLMs, initially only capable of general conversation, to transform unstructured data into a domain-specific Q&A LLM. This framework involves several stages, starting with a **Generator LLM** for producing questions, followed by another **Generator LLM** for generating corresponding answers, a **Critique LLM** for evaluating the generated questions and answers,

and finally, the **Final LLM** which is fine-tuned for domain-specific Q&A. After training the **Final LLM**, we employ a Retrieval-Augmented Generation (RAG) system paired with Prompt Engineering techniques to retrieve relevant documents to answer questions, ensuring that the **Final LLM** can respond with correct and expert knowledge as anticipated.

In our experiments, we collaborate with the National Taiwan University Hospital, using medical literature and research documents on genetic hearing loss in otology as our domain-specific documents. Ultimately, this LLM aids doctors and patients in better understanding diseases and improving diagnostic efficiency. The experimental evaluation involves a test set of frequently asked questions during consultations, provided by specialized otologists at National Taiwan University Hospital. The scoring is based on human evaluations of the answers' quality, completeness, and the absence of hallucinations.

**Keywords:** LLM, RAG, Fine-tune, Dataset, Domain specific, Otology, Hereditary hearing Loss



## **Contents**

		F	Page
Verification	Letter from the Oral Examination Committee		i
Acknowledg	gements		iii
摘要			V
Abstract			vii
Contents			ix
List of Figur	res		xiii
List of Table	es		XV
Denotation			xvii
Chapter 1	Introduction		1
Chapter 2	Related Work		5
2.1	Large Language Models		5
2.1.1	Specific Large Language Model		6
2.1.2	Task adapting pre-training		7
2.1.3	Instruction Tuning		8
2.1.4	Quantization		9
2.1.5	Low-Rank Adaptation (LoRA)		9
2.2	Retrieval Augmentations Generation (RAG)		10

2.2.1	Data Chunking	. X		<b>41</b>
2.2.2	Massive Text Embedding			12
2.2.3	Vector Database		要。學	12
2.2.4	Keyword			13
2.2.5	RAFT			13
2.3	Data Augmentation			13
2.4	Conditional Task Generation			14
Chapter 3	Methodology			15
3.1	Data Pre-processing			15
3.2	Training Set Generation			16
3.3	Training and Inference			19
Chapter 4	Experiments			23
4.1	Data			23
4.1.1	Chinese close domain data			23
4.1.2	Open data			24
4.2	Settings			24
4.2.1	Model Choice			25
4.2.2	Text Split			26
4.2.3	Critique index			27
4.3	Evaluation Set			28
4.3.1	Key Word Detection			28
4.3.2	Answer similarity			30
4.4	Dataset Generating			30

4.5	Result	
Chapter 5	Conclusion	· 35
Chapter 6	Future Work	37
References		39
Appendix A	— Prompt	45
A.1	Prompt	45
A.1.1	Generation prompt	45
A.1.2	2 Critiques prompt	45
A.2	Dataset	46
A.2.1	PubMedQA	46
A.2.2	2 PrivacyOA	46





# **List of Figures**

1.1	Framework Overview	4
2.1	Typical RAG Overview	11
3.1	Data Pre-process	17
3.2	Chunking Result Example	18
3.3	Training Set Generation	19
3.4	Training and Inference	21
3.5	Training Set Example	22
4.1	Assistant Result Example	29
A.1	Groundedness Critique Prompt	45
A.2	Relevance Critique Prompt	46
A.3	Standalone Critique Prompt	46
A.4	Similar Critique Prompt	47
A.5	Example of PrivacyQA Format	48
A 6	Example of PrivacyOA Format	49

xiii





## **List of Tables**

4.1	Details of Open Data	24
4.2	Chunking break points	25
4.3	Choice of base model	26
4.4	Text Split choice	27
4.5	Average generation speed per query	31
4.6	Good Quality Ratio	31
4.7	Performance of different model. This is the study on the performance	
	of different language model act as Generator, Critique or Final LLM	33
4.8	Performance of different model without low quality example. This is	
	the study on the performance of different language model act as Generator,	
	Critique or Final LLM. In this set of experiment, we do not add the bad	
		33
4.9	Critique or Final LLM. In this set of experiment, we do not add the bad	33
4.9	Critique or Final LLM. In this set of experiment, we do not add the bad quality sample to see the performance on all positive evaluation set	
	Critique or Final LLM. In this set of experiment, we do not add the bad quality sample to see the performance on all positive evaluation set  Fine-tune performance. This is the comparison of performance on the	





## **Denotation**

LLM 大型語言模型 (Large Language Model)

RAG Retrieval Augmentation

GPU Graphics Processing Unit

GP-GPU General Purpose Graphics Processing Unit

VRAM Video random-access memory

NLP Natural Language Processing

LoRA Low-Rank Adaptation

SOTA State of the Art





## **Chapter 1** Introduction

Deep learning, a subset of machine learning, has its roots firmly planted in the research and developments of the 1980s. It gained significant traction and visibility due to the exponential increase in computational power, particularly after the advent of General Purpose Graphics Processing Units (GP-GPUs). These advancements have enabled the processing of complex deep learning architectures at speeds that were previously unattainable, making practical applications feasible.

In the field of Natural Language Processing (NLP), the evolution of transformer architectures has been groundbreaking. Initially developed as encoder-decoder models, transformers quickly set new standards for machine learning applications. This approach was refined with the advent of the GPT series, which utilizes only the decoder part of the transformer architecture to generate text. Specifically, OpenAI's ChatGPT, released in November 2022, exemplifies this innovation. ChatGPT employs the decoder-only approach of the GPT models, demonstrating remarkable capabilities in generating human-like text, which sparked widespread interest and a flurry of research into LLMs.

One key method to tailor an LLM for specific tasks or to generate responses in a particular manner is through fine-tuning. Fine-tuning involves adapting a pre-trained model on a smaller, domain-specific dataset to refine its capabilities according to specific needs.

1

This process is crucial for adapting the generic abilities of an LLM to specialized tasks, enhancing both its precision and utility in real-world applications. Through these continuous advancements, deep learning and LLMs are increasingly integrated into daily technology, making them an integral part of modern computational solutions.

Fine-tuning, however, needed lots of good examples for LLM to learn from. Traditionally, good examples means that we need lots of human annotation in order to fit human expectation. To solve this problem, more research has been trying to solve this problem through the generation power of LLMs. Prior work like self-instruct[22] augment tuning data from seed examples manually written by experts through few-shot learning. Evolve-Instruct [25] improves self-instruct using the technique by rewriting different parts in the instructions. GLAN [12] tries to generate instruction purely by LLM by dividing into more sub subjects using the pre-trained knowledge. However, in a domain of closed knowledge that hasn't been trained into the LLM, child hereditary hearing loss in this case, it is impossible to generate instruction dataset with knowledge in such field since LLM does not acquire such professional knowledge in the field and thus not increase the performance in learning the close domain knowledge even with RAG implemented after fine-tuned dataset created with these methods.

In this work, we introduce a framework that help those who wants to transform their close-domain knowledge into quality fine-tune dataset. The process can be seen in Figure 1.1. This process can be broken down into the following steps. First collect the domain related documents. This parts can be applied to all sort of documents, suck as pure text file, PDF file or even traditional paper documents after using the OCR to transform them into text file. Second, we will chunk up the existing documents into smaller chunks. After chunking, we can create the needed format based on these chunks leveraging the power

of LLM(Generator LLM and Critique LLM) automatically, in this work, we do experiments on Question and Answering since it is the most used scenario on a LLM. In this part, all sorts of based LLM can be used based of the language used as long as they are publicly open. Thirdly, we will fine-tune the **Final LLM** with the create fine-tune dataset base on the usage, in this work we fine-tuned **Final LLM** based the RAG usage. As we can see, the whole pipeline can be purely done by LLM, with this possibility, this framework can adapt to the newest LLM that has just been released in no time with a little fine-tune.

Finally, we employ our method on close domain dataset and open dataset individually to prove our work. The result show that LLM fine-tuning with dataset created with our method and the help of RAG can achieve a XXX improvement on our close domain dataset and YYY on open dataset.

This thesis is divided into several parts. Chapter 2 includes all related work for our work and other techniques that involves in our experiments in Chapter 4. Chapter 3 introduce the methodology of our work thoroughly. Chapter 4 shows the experiments outcome of our work compared result. Finally, Chapter 5 summarize the work and discuss the possible future works.

3



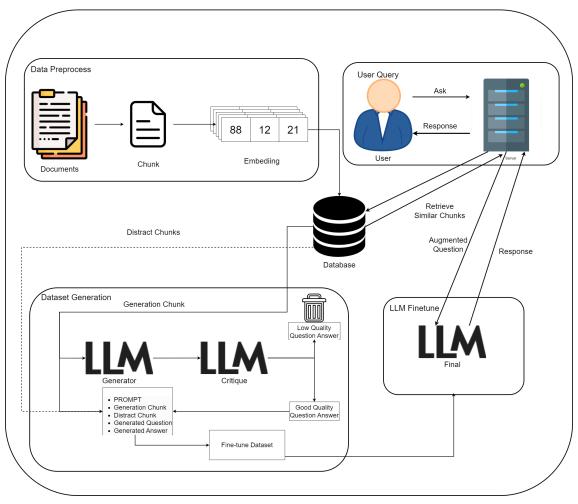


Figure 1.1: Framework Overview



## Chapter 2 Related Work

In this chapter, we will introduce the related research that inspired us to develop our study. This following topics: Large Language Models, Data Augmentation, Instruction Tuning, and Retrieval Augmentations will be covered, which are all crucial parts of our study. At the end, we will also introduce two method that try to adapt new knowledge to Large Language Model and we will be comparing our result in the chapter 4.

### 2.1 Large Language Models

Large language models have only come to public attention in the past two years, but they have become major stars in research. In this section, we will briefly introduce what is the Large Language model since this is a not only the crucial part of our research, but also the singularity of humanity.

A large language model (LLM) is a type of artificial intelligence based on the architecture of the transformer that processes and generates human-like text based on patterns learned from vast amounts of data. These models, such as OpenAI's GPT series, Mistral's mistral series or Meta's Llama series, use deep learning techniques to understand and produce language across various topics and styles. They are capable of answering questions, text generation, and even engaging in conversation, making them powerful tools for ap-

plications in customer service, content creation, and especially in the research.

#### 2.1.1 Specific Large Language Model

A Biomedical Large Language Model[10, 16, 17] represents a the intersection of artificial intelligence and healthcare. Leveraging vast amounts of biomedical literature, clinical data, and scientific research, these models are fine-tuned to comprehend and generate language specific to the biomedical field. Biomedical Large Language Models are designed to assist in tasks ranging from answering complex medical queries and summarizing research papers to predicting protein structures and suggesting possible treatment pathways. The potential of Biomedical Large Language Models to transform medical diagnostics, therapeutic development, and personalized medicine is vast, promising to enhance the accuracy and efficiency of healthcare services and research while staying abreast of the rapidly expanding biomedical knowledge base.

But since biomedical is still a field of life and death, the future of of AI doctor might not come that soon, it is important for a professional to judge the correctness of the outputs. That is why is our study, we work closely with the doctors to verify the result.

Chinese Large Language Models[1, 3, 20, 27] are also a popular research topic due to the vast Chinese-speaking population, which comprises a significant portion of the global community. This interest is driven by the need to improve machine understanding and generate more accurate and contextually relevant responses in Mandarin and other Chinese dialects. The development of such models not only facilitates better user interaction but also opens up numerous possibilities for applications in education, business, and technology across Chinese-speaking regions. Moreover, advancements in this area help in

preserving and processing linguistic and cultural nuances that are unique to Chinese, further enriching the capabilities of artificial intelligence in diverse societal contexts.

In our study, our experiment's primary goal is to develop a Traditional Chinese specific Medical Large Language Model, specifically for medical applications, by fine-tuning it using unstructured documents. This task is particularly challenging considering that resources on the internet are predominantly in English. Moreover, in the Chinese context, over 99% of the online data is in Simplified Chinese. Fortunately, a major tech company in Taiwan, MediaTek and the National Science and Technology Council, R.O.C., have published their Traditional Chinese Model individually to help us address this issue.

#### 2.1.2 Task adapting pre-training

This concept has been proposed by gururangan[4] in the 2020. This is also commonly known as continuous pre-training. When we have lots of data that has not yet been learned by the large language Model, we can leverage such method to allow the model to absorb new knowledge comprehensively. There are lots of advantage for this method, like full parameters update allow the model and tokenizer to truly learn about new knowledge or the data doesn't need extract pre-process before use. But on the other hands, it also has some other disadvantage like the process is time and resource consuming, or hard to adapt due to the data size too small. Due to these disadvantage, nowadays, people tends to use instruction tuning over continuous pre-training.

#### 2.1.3 Instruction Tuning



Instruction Tuning is a critical technique in the development and enhancement of Large Language Models (LLMs), such as OpenAI's ChatGPT, which are employed by billions of people daily. The fine-tuning process of these models is essential to tailor them to specific tasks or improve their overall performance. Before initiating the fine-tuning process, it is crucial to clearly define the intended purpose of the model. Among the various types of LLMs, chat-based models like ChatGPT are perhaps the most visible and widely used. These models are designed to engage in dialogue with users, providing responses that are not only accurate and relevant but also contextually appropriate and engaging. The fine-tuning process for such models involves training them on a diverse dataset that includes a wide range of dialogues and interactions. This training helps the model learn the essence of human conversation, including idiomatic expressions, cultural references, and the subtle cues that convey politeness and empathy.

The fine-tuning process is not just about expanding the model's knowledge base or improving its understanding of language syntax. It is also about aligning the model's responses with ethical guidelines and ensuring that it can handle sensitive topics with care. Instruction Tuning, therefore, plays a pivotal role in the deployment of LLMs, ensuring that they can function effectively in real-world settings and provide meaningful interactions with users. This makes the model not just a technical tool but a sophisticated interface that enhances human-computer interaction.

#### 2.1.4 Quantization

Quantization significantly enhances the efficiency and applicability of large language models (LLMs) by reducing their size and speeding up their computation. This technique, which involves converting high-precision floating-point numbers into lower-precision formats, such as integers, decreases the memory requirements by up to 75%, FP32 to INT4 for instance. This reduction not only facilitates the storage and execution of LLMs on resource-constrained devices like smartphones and edge servers but also boosts processing speeds, making real-time applications more feasible.

Moreover, by adopting quantization, it's possible to deploy advanced language models in a more energy-efficient and cost-effective manner. The reduced computational load also means lower power consumption, which is crucial for sustainable and economically viable AI deployments. With strategies like post-training quantization and quantization-aware training, the balance between performance and efficiency can be finely tuned, preserving the model's effectiveness while leveraging the benefits of reduced precision.

#### 2.1.5 Low-Rank Adaptation (LoRA)

The technique of Low-Rank Adaptation (LoRA) [6] addresses the challenge of deploying state-of-the-art large language models by modifying them in a computationally efficient manner. LoRA leverages the architecture of transformer models, applying low-rank updates to specific weight matrices. This allows the adaptation of these models to specialized tasks without retraining the entire network, significantly reducing the computational burden and preserving the pre-existing strengths of the model.

LoRA's key innovation is its use of low-rank matrices to approximate necessary changes, dramatically reducing the number of trainable parameters. This makes it feasible to fine-tune large models on modest hardware, broadening the accessibility and applicability of advanced machine learning models to a variety of tasks, including natural language processing and beyond.

### 2.2 Retrieval Augmentations Generation (RAG)

Retrieval augmentations Generation is first introduced by Lewis [11] and is an innovative approach in the field of natural language processing that enhances the capabilities of generative models by incorporating external knowledge sources. This method leverages the power of both retrieval-based and generative AI models to produce more accurate, relevant, and contextually rich outputs. By integrating the retrieval of information from vast databases or corpora with the generative process, RAG models can achieve a significant improvement in tasks such as question answering, content creation, and language translation, where depth and precision of knowledge are crucial.

The first component of RAG is the retrieval phase. In this stage, the model queries an external database to fetch relevant documents or data snippets that are closely aligned with the input query or prompt. This retrieval is typically powered by a transformer-based encoder or a similar mechanism, which converts the input text into a vector space representation. The system then performs a similarity search among the knowledge database to identify the most pertinent information. This process ensures that the generative model is informed by up-to-date and context-specific data, enhancing the quality of its output by grounding it in factual content.

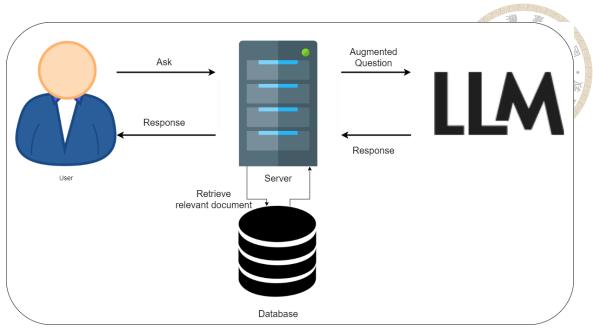


Figure 2.1: Typical RAG Overview

The second component is the generation phase, where the actual content creation takes place. Using the retrieved data as a contextual backbone, a generative model (LLM normally) synthesizes information into coherent and contextually appropriate responses. This step involves interpreting and integrating the retrieved information. The generative model's ability to merged various pieces of information and generate fluent text is what makes RAG particularly powerful, allowing for responses that are not only accurate but also nuanced and deeply informative. The general overview of RAG can be seen in figure 2.1.

#### 2.2.1 Data Chunking

Before we start to transform the raw data into embeddings, what we need to do is to cut the raw data onto smaller chunks. The reason is first, all LLM has a maximum context length that can takes, when we tries to work with RAG system without doing chunking first, the LLM will chunk up the content over the maximum length for us. Second, When the content is too long inside the searching process of retrieve, it will increase the possi-

bility it loose focus on the things we want. Different chunking method will also effects the result of retrieve, when importance messages are cut in between. Thus, various chunking method methods have been proposed to solve this problem.

#### 2.2.2 Massive Text Embedding

As mention in 2.2.3, when we try to establish a RAG system with fast searching algorithm, we have to transform the raw data into embeddings. Text embedding transforms raw text into numerical representations that capture the inherent semantics and contextual relationships of words and phrases. This process is crucial for enabling computational models to perform tasks such as classification, clustering, and sentiment analysis on text data. However, the scale of data in massive text embedding scenarios presents unique challenges, including the handling of high-dimensional data, the mitigation of noise, and the management of diverse linguistic features.

#### 2.2.3 Vector Database

One popular type of retrieval involves searching through a pre-established database to find data that closely matches the query. The data stored in these databases are typically in high-dimensional vector format rather than raw form, to expedite searches and conserve storage space. Given that the data is in vector format, it has become a important point in the field of Retrieval Augmentations, as it can process any type of raw data.

However, selecting the appropriate model or method to transform raw data into embeddings is a critical consideration. Ensuring that these embeddings are sufficiently similar is another area of intensive study, particularly when using a large language model

(LLM) based on transformer technology. Beyond the choice of model, the selection of a similarity algorithm is another crucial factor to consider.

### 2.2.4 Keyword

Beside pure embeddings to be saved in the database, some would also save some key words from some keyword table to make the searching more intuitive and prevent the bad choice of embedding model to messed up. One of the most commonly used Keyword will be the document title or the paragraph title to make the LLM contain these information in the outputs.

#### 2.2.5 **RAFT**

RAFT: Adapting Language Model to Domain Specific RAG [28] is a technique that has shown promising result on RAG system. The main process is to fine-tune the LLM with the question, documents and Answer along with the prompt to let the LLM not only memorize these documents through training process, but also tell the LLM it will be like taking a open-book exam during the inference process.

### 2.3 Data Augmentation

Data augmentation is a powerful technique used in the field of machine learning to enhance the size and quality of training datasets by introducing modifications and variations to the data. This method plays a crucial role in improving the robustness and generalization capabilities of machine learning models, particularly in domains like image processing, natural language processing, and audio analysis. In the domain of creating

custom instruction tuning dataset, this is a very important and highly used technique to enrich the created instruction tuning dataset. Some of the most common are giving seed example by human expert and do one or more than one of the followings: paraphrasing, translation, synonym replacement, tense rewrite, change sentence format or contrastive learning.

#### 2.4 Conditional Task Generation

The phrase conditional task generation is first shown in Nayak's [14] work. This is a task that convert's unannotated text in to task-specific instruction fine-tune dataset. In Nayak's work, they have proposed a fine-tuned model called "Bonito" that serve the purpose of generating such dataset. However, we found that there are some disadvantage of their work. First, their "Bonito still need training to serve the purpose, that is still not good enough. Second, by only create normal fine-tune dataset may not achieve well enough performance. We will compare the result of their and ours in the experiments part.

14



## **Chapter 3** Methodology

In this chapter, we will comprehensively introduce our work. First, we will describe how we collect and pre-process the data to prepare it for use. Next, we will discuss the main process of our work, which involves creating a fine-tuned dataset for fine-tuning domain-specific large language models (LLMs). Finally, we will describe the training and inference processes.

### 3.1 Data Pre-processing

This part we will introduce some simple pre-process we did in this work. We aims for as less pre-process as possible to make our work can be easily adapted to any kinds of source document. The detail data pre-process can be seen at Figure 3.1.

First, we collect the document and prepare them to be directly used by our pipeline. Documents of any domain and of any amount can be applied to our work. For most cases, the documents is usually in PDF format documents, but sometimes it could be in other format. In such case, we recommend to use any kinds of tool to convert the documents into text file or PDF to be easily and properly applied to our method. For the documents we do experiments on our work are mainly in PDF format with some text file also.

After getting the documents, we will chunk up all the text into chunks. Unlike traditional chunk, to make our chunks to make in context learning process during training and inference, we does not chunk up our text just by the length strictly. Instead, we use a method called recursive text splitter that can help us chunk up the chunks more beautifully. The recursive text splitter basically chunk up the word like the traditional way, but then it find the nearest break point that is set by the user to make the ending more beautiful and not loosing information cause by chunking. However, just by chunking up through predefined break point still could cause some information chunked since they might still be in two sentences. With these predefined break points, our chunking would make each chunk contains more sentimental meaning. A demonstrate can be seen in Figure 3.2.

After we chunk up all the documents, we also extract keyword from each chunks and save as one of the metadata in the document format. Here we have tried several ways of extracting keywords from the chunk, and we still finds LLM has the best understanding of keyword consider the length of our chunk. Other traditional method like TF-IDF also can perform well, but to stay to the topic of our research, we've decided to also leverage the power of LLM to also help us solve this problem.

Finally, we can save the chunks along with the keywords into database for later usage. We here followed regular RAG system by converting the chunk into the embedding space and save in a vector database.

### 3.2 Training Set Generation

In this section will be most the important part of our work. By leveraging the power of pre-trained LLM with strong ability of in-context learning, we can built up our fine-tune

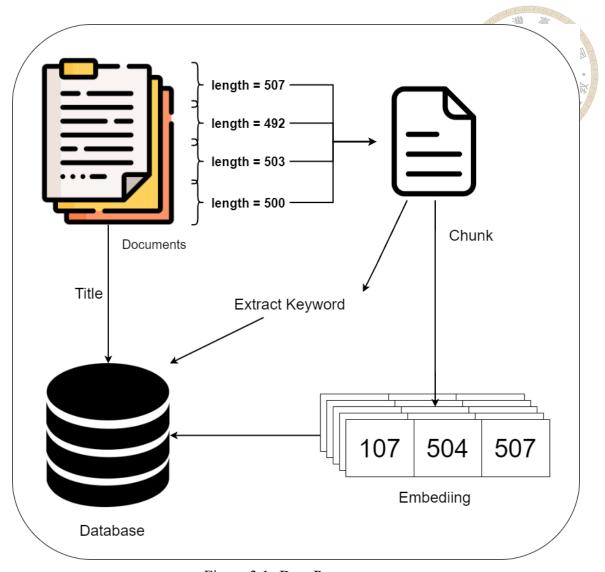


Figure 3.1: Data Pre-process

dataset from scratch and with out the help of human interference. The detail training set generation pipeline can be seen at Figure 3.3.

Inspire by previous work [18, 23, 26], we let the LLM role play as the character we need to be. For our experiment, our data is focus in the medical related field, and so does the characters. That give us high flexibility to generate other questions, answers or other needed template when the document field is changed.

First, the **Question Generator LLM** is to help us generate question, so it first act as an patient to ask about the illness or a junior doctor querying to know the knowledge.

#### Chunking Result Example

X性聯染色體隱性遺傳。女性子代之遺傳表現類似於體染色體隱性遺傳,只帶有一個致病點位為帶因者,不會表現疾病;帶有兩個致病點位才會表現出疾病。男性子代之遺傳表現類似於體染色體顯性遺傳,帶有一個致病點位就會表現出疾病。POU3F4(DFNX2)是最常見的性聯遺傳非症候群型聽損,位於坐落於 X 染色體的長臂上 (Xq21.1),其臨床表現為學語前混合型聽損,有先天性鐙骨固著 (Congenitalstapesfixation) 及耳蜗發育低下(cochlearhypoplasia)之變化,且聽力會逐漸變差 (progressive) 發展成極重度聽損,此一族群在鐙骨手術時,有較高的風險會發生外淋巴液井噴 (perilymphgusher)。第六節粒線體遺傳之非症候群型聽損粒線體 (Mitochondrial)為細胞質中的細胞能量工廠,主要功能是產生腺苷三磷酸 (adenosine triphosphate, ATP), 粒線體有獨立於細胞核的遺傳物質 (MitochondrialDNA, mtDNA),但缺乏細胞核中的 DNA 修復機制, 所以比起細胞核中的 DNA 有更高的機會累積基因變異。

keywords: 性聯遺傳, POU3F4(DFNX2), 粒線體遺傳

Figure 3.2: Chunking Result Example

We then give the "patient" a chunk of words to based on the chunk to ask us some questions. Other than generate question from the whole chunk, we want the question to have more direction of possibility and not constraint by the original direction of chunk, we also generate question purely based on single keyword.

Secondly, the **Answer Generator** would be a senior professional doctor that help the others and answer these questions. We take the questions generate by the **Question Generator LLM** one by one with the chunk and expect the "professional doctor" can answer our question purely based on the chunk. Finally, the **Critique Generator** will act as the factual judge, who can give anything a score and a comment in the given content. We will provide the chunk, question and answer for the "judge" to give a several score to evaluate whether the quality is good enough. The specific evaluations index for our document in the otology domain will be discuss in the 4.2.3.

We then save those question and answer that is evaluated as "Good Quality" and

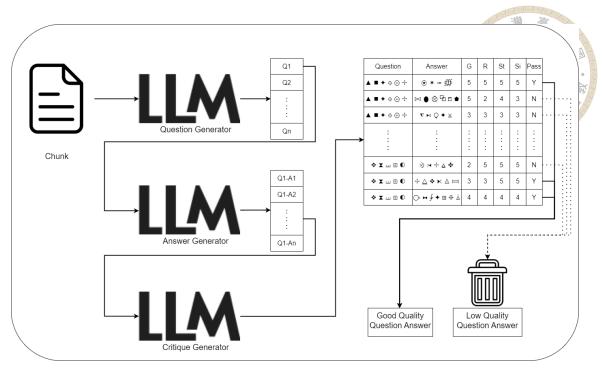


Figure 3.3: Training Set Generation

those evaluated as "Low Quality" separately. For those "Good question" can directly use as fine-tune dataset after slightly added with some prompt. As for those "Low quality", we filer some of the question out and modified the answer as unanswerable response and also add into the fine-tune dataset for out model can learn when to response one crucial response "I don't know the answer." when it need.

# 3.3 Training and Inference

In this work, we aim to develop a system capable of extracting factual answers from provided documents. Additionally, it should respond accurately even when the retrieval process fails to locate the correct documents. The detail training and inference pipeline can be seen at Figure 3.4.

During the inference process, the system is designed to operate as if taking an openbook exam, extracting factual answers from concatenated documents. Similarly to humans with a dataset formatted to mimic this learning environment. We utilize the Retrieval-Augmented Generation (RAG) system, enabling the system to learn from factual documents and provide factual answers.

The training format as shown in picture is {P, C, Q, A} (where P stands for prompt, C for chunks, Q for question, and A for answer) for the fine-tuning process, and {P, C, Q} for the inference process. We expect the system to generate the {A} part, representing the answer that will be output to the user.

Specifically, P (prompt) includes system prompts informing the LLM of its role and other information, as well as user prompts dictating the rules and the output format of this conversation. The C (chunks) varies slightly between training and inference; during training, it consists of factual chunks used to generate the Q (question) and A (answer) in Section 3.2, as well as distracting chunks randomly selected from the database. Following Zhang's [28] settings, only 80% of C (chunks) during training contains factual chunks, with the remainder replaced by another distracting chunk. For inference, the chunks consist of the top-k chunks retrieved from the vector database using similarity search. The Q for training involves questions with factual chunks generated in Section 3.2, while for inference, it simply consists of user input. Finally, A is also generated in Section 3.2.

After creating the training dataset, we will begin training the LLM using supervised instruction tuning [22]. The instruction format can be seen in Figure 3.5.

We follow regular training loss for fine-tune LLM, cross-entropy as in equation 3.1. The q(x) represent the probability distribution of the possible next token, as for the p(x) represent the actual distribution of the model output. The goal is to make to token predic-

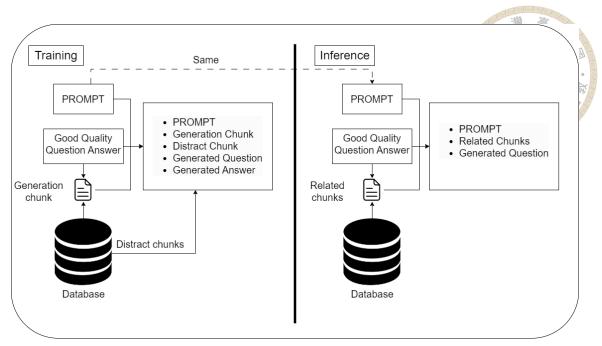


Figure 3.4: Training and Inference

tion's cross entropy loss be as small as possible. This will generally aligned the model to a specific way of response.

$$H(p,q) = -\sum_{x \in \text{tokens}} p(x) \log q(x) \tag{3.1}$$



#### Training Set Example

#### **SYSTEM:**

我是一個來自國立臺灣大學醫院的耳鼻喉科專業醫生,樂於以我專業的醫學見解的立場幫助使用者,並且我會在中華民國臺灣的文化背景下下用繁體中文回答問題。

#### **PROMPT:**

你將會被提供一個問題 (Question) 以及五段內文 (Context)。根據下面提供的資訊,回答這個問題。以下是需要注意的事項:

以簡潔的形式回答三句話以內的答案,但是在遇上專有名詞的時候可以特別說明並且陳列相關數據。

請勿重複跳針回答,並且不要輸出無意義的空白。

對於內文不相關關的問題,請回答"非常不好意思,目前沒辦法回答您的問題"。

#### **DOCUMENTS:**

Content too long, omitted.

#### **OUESTION:**

在兒童人工電子耳植入的過程中,哪些評估和檢查是術前、術中和術後的重要步驟?

#### **ANSWER:**

在兒童人工電子耳植入的過程中,重要的術前、術中和術後評估和檢查包括:

- 1. 聽性穩定誘發反應檢查 (Auditory Steady State Response, ASSR) 用於檢測電生理活動和確認聽神經功能。
- 2. 腦幹波峰衰減聽力圖 (APG) 測量聽覺傳導路徑的完整性和靈敏度。
- 3. 音叉測試-檢查內耳結構的完整性和感覺細胞的存在。
- 4. 影像學研究如磁振造影 (MRI) 或電腦掃描 (CT) 評估內耳構造和大腦皮質。
- 5. 語言能力評估-了解患者聽覺理解程度。
- 6. 心理社會情緒評估-評估患者及其家庭對聽覺重建的需求和期望。

Figure 3.5: Training Set Example



# **Chapter 4** Experiments

To prove our study, we have establish a series of experiments to verify our work. Since our work are mainly focus on creating close-domain fine-tune dataset that can help to increase the user experience on using the LLM with the extra knowledge that has grown from after fine-tuning. We will first fine-tune the model using our problem then do related experience of different setting and then compare with LLM that does not use our method. During the generation of **Final LLM**, we will also prove that our method's robustness and ability of easy to use when generating dataset.

## 4.1 Data

#### 4.1.1 Chinese close domain data

We decided to collaborate with the Department of Otolaryngology from National Taiwan University Hospital, they provide us with specific documents on the topic of sensorineural hearing impairment to help us do experiments on proving our works.

They have provide us several related documents that can not be found online and they are all in traditional Chinese making them more impossible to be pre-trained in the pre-train data. The total data are composed of 12 PDF documents and of total 112 pages

not counting the reference pages if the document is a thesis paper.



## 4.1.2 Open data

We here test our method using public dataset PubMedQA [8] to show our method's usability, robustness and performance. We here mainly compare with the Nayak's [14] methods, which is also another method involves generate instruction dataset automatically. To adapt out method to the open dataset like PubMedQA [8], we extract the **CONTEXTS** part of the dataset and save it into the database then use our method to create question and answer pair to the content.

Dataset	Train	Test
PubMedQA	211,269	10563
PrivacyQA	10923	10923

Table 4.1: Details of Open Data.

# 4.2 Settings

This section we will provided the specific setting of our experiments on this work.

First, for the chunking process, we use recursive text splitter with the followed break points in Table 4.2 to better fit for the the traditional Chinese documents with better chunking effect. It is recommended to use our setting for Chinese or Japanese documents, as for other language, please modify based of the context. Then we save the chunks after turning in to embeddings using INTSIG's [7] model and is based on kusupati's [9] method, which we pick from the leading board [13] that has competitive result but also easy to use.

Second, for the **Generator LLM**, we generate 5 questions for each chunk and also use keywords to generate two other question only related to that keyword. That is total of

break point	text representation	code representation
newline	"\n"	\u000a
double newline	"\n \n"	\u000a \u000a
space	"	\u0020
full stop	""	\u002e
comma	""	\u002c
Zero-width space	" "	\u200B"
Fullwidth comma	","	"\uff0c"
Iographic comma	"、"	"\u3001"
Fullwidth full stop	"•"	"\uff0e"
Ideographic full stop	" 。"	"\u3002"

Table 4.2: Chunking break points.

a 11 questions per chunk. Finally we generate 3 answers for each question generated. For each **Generator LLM**, we set the max new token length to be 600.

Third, we set 4 indicator for the **Critique LLM** with groundedness, relevance, standalone and similarity. The full score for each indicator is 5 points. Each question and answer pair need to achieve at least 3 for each indicator and a total of 13 points.

Finally, for the training process, we set the k of top-k when setting the number of distract chunks 4, meaning the total chunks pass to LLM with a total of 5 whether with a generation chunk or not. During the training process, we set the max sequence length for the supervised fine-tune trainer to 1600 to allow the model to be able to fit it all the documents. We also use the Qlora[2, 6] technique to reduce the memory usage during the whole training process and for the rank we set as 64.

### 4.2.1 Model Choice

We have choose the two most well known Traditional Chinese base LLM to be our base model, they are BREEZE-7B [5] and TAIDE-8B [19]. They have both demonstrate great performance and fine-tune using only traditional Chinese content from their original

base model. We will be doing experiments on testing the ability of them individually. To be more specific, we will use these two model for cross-referencing: **Generator LLM**, **Critique LLM**, and **Final LLM**. We aim to compare their performances to determine which one performs best.

**Model Choice** Breeze-7B TAIDE-8B

Table 4.3: Choice of base model.

## **4.2.2 Text Split**

We also do experiments on the setting that can get the most correct chunk in the retrieval process. First we chunk up the documents based on four setting, then we ask also two different Chinese LLM to generate question, then we use that one question to retrieve documents to see which setting has the highest ratio of getting back the original chunk. The chunk size is 256 or 512, and the overlap size is 0 or 100. For each {chunk size, overlap size}, we list the chunk total chunk up chunk size: {256, 0}: 458, {256, 100}: 688, {256, 0}: 243 and {256, 0}: 265. During the test, we do not filter the quality with the critique LLM, because the answer is not generated. The top-5 and top-1 hit ratio result can be seen in the Table 4.4. To be more specific, top-1 means that the generation chunk use for the question is retrieve as the chunk with highest similarity, while top-5 mean that the generation chunk is retrieve in one of the five chunk. As shown in the result, our method can achieve a 100% top-5 hit and at most a 67.48% of hit as the most relevant question in the setting of taking BREEZE as Generator LLM with chunk length as 512 and 0 overlap. In this experiments, we see that 0 overlapping results all have a better result than those with 100 overlap. We attribute this result to our break point strategy to make the embeddings with stronger sentimental meaning.

overlap	0	100
256	0.6289	0.5174
512	0.6748	0.5720

(a) Ratio of hit on BREEZE generation chunk in top-1 retrieve chunk.

overlap	0	100
256	0.5942	5032
512	0.6482	0.5714

<sup>(</sup>c) Ratio of hit on TAIDE generation chunk in top-1 retrieve chunk

overlap	0 *	100
256	1.000	1.000
512	1.000	1.000
	Alch.	100

(b) Ratio of hit on BREEZE generation chunk in top-5 retrieve chunk.

overlap	0	100
256	1.000	1.000
512	1.000	1.000

<sup>(</sup>d) Ratio of hit on TAIDE generation chunk in top-5 retrieve chunk.

Table 4.4: Text Split choice

### 4.2.3 Critique index

As stated in 3.2 and 4.2, we have four index, groundedness, relevance, standalone and similarity, in our experiment for our generated question and answer to evaluate them to be a pair of good quality data. Here we will introduce these four index with more detail. All four index is done by the LLM and is asked to answer in the specific format for us to collect score and critique individually.

The first index, groundedness, is set to see if the answer can be properly answer with the content, since we are training the model to mainly followed the content to answer and not hallucinate one. So to properly train for this purpose, we need to train with those answer that can be found inside the content for the good quality dataset. Second index, the relevance index, is set to see if the question would be ask by a patient that does not have related background. The LLM tends to ask question too specific professional background which is not how human ask question is why we set this index. Third index, the standalone index, is to see if the answer itself can be understand easily and doesn't require too much background knowledge or further look up. Last but not least, is the similar index,

We want to filter out those answer and question are too similar in the sentence structure, which means its basically answering the question by repeat the question. This could highly happen when a question's quality is not good enough.

The specific prompts can bee seen in Appendix A.1.2.

#### 4.3 Evaluation Set

Evaluation of open question has been a very hard quest since LLM can have unpredictable output even under the exact same setting. Also, our experiments is being done on the very specific domain that can't not easy to be judge by the LLM nor normal human without related background. But we still try to do it to see the result. The example of the output of our work can be seen in Figure 4.1.

First, the evaluation set can be divided into several parts but all of them is set up by our research team after reading carefully the related document provided by the hospital and then verify by the doctors to meet the hardness the correctness. First, we generate these question and categories them into two type: professional question. and introduction questions. Then we also crawl down some common seen related to otology to also test the basic otology knowledge, in which all of them is categories as introduction questions.

# 4.3.1 Key Word Detection

While we write down questions and answers for the evaluate set, we also extract at most 3 keywords from each answer to be detect in the real evaluation stage. These keywords are also verify by the doctors.

### Assistant Result Example

User: 次世代基因定序可以分成哪些步驟?

Ground Truth: 定序而同時定序許多 DNA 片段, 因此可節省大量的時間與經費。次世代定序的基本原理可區分為: 製備 template、定序 (sequencing)、及基因體重組 (genome assembly, 即資料分析) 等三大步驟。

Keyword: 製備、定序、基因體重組

Assistant: 次世代基因定序(Next-Generation Sequencing, NGS) 的步驟主要包括:NGS 樣本前處理:包含 DNA 萃取 (DNA extraction)、DNA 破碎化 (DNA fragmentation)、以及端酶修剪 (endonuclease trimming)等,目的是得到適當大小的 DNA 片段。NGS 感測器:在這個步驟,DNA 碎片會被黏接上特定的模板 (template),並且進行標靶 (targeted) 或全基因體 (whole-genome) 的複製過程。NGS 流量細胞 (flow cell):經複製完成的 DNA 片段會在流量細胞上順流排空,並經由影像感測器 (imager) 的觀測,在 DNA 序列進行氫離子 (proton) 的定序反應。NGS 資料處理及分析:經過影像感測器的數據被儲存與分解,將基因定序資料轉換為有意義的序列資料,接著開始進行生物資訊分析,如序列對齊 (sequence alignment)、物種比對 (species comparison)、和基因功能預測 (gene function prediction)等。這些步驟彼此相互關聯,最終目的是獲得完整且準確的基因序列資料。然而,由於 NGS 技術仍在迅速發展中,相關技術及分析方法也持續被創新及改良,因此上述步驟可能隨著技術發展而改變。

Figure 4.1: Assistant Result Example

In the parts of keywords detection, we have calculate the precision (4.1), recall (4.2) and F1 score (4.3) for this metric. The precision is considered one of the most important since we think avoiding false positive case is more important in this study. We defined the TP(True Positive) as the pre-defined keywords properly appear in the LLM response, FP(False Positive) as the pre-defined keyword show none and we set to one for each question meaning it's high possible of wrong direction of response, and FN(False Negative) as the pre-defined keywords that fail to appear in the response. As for TN(True Negative) is relatively not important or hard to defined in our task and is not account in both precision and recall.

$$Precision = \frac{True \ Positives \ (TP)}{True \ Positives \ (TP) + False \ Positives \ (FP)}$$

$$(4.1)$$

Recall = 
$$\frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$
 (4.2)

F1 Score = 
$$2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right)$$
 (4.3)

## 4.3.2 Answer similarity

We also compare the ground truth answer and the LLM output using semantic similarity to check the output is similar to the ground truth answer in the semantic level and not just the word level. The model we use is Xiao's [24] BAAI/bge-large-zh-v1.5, which is one of the most common embedding model for Chinese. The similarity function is cosine similarity (4.4), where A is the embedding for the ground truth answer and B is the LLM output.

Cosine Similarity = 
$$\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$
 (4.4)

# 4.4 Dataset Generating

For the process of dataset generation, we have do experiments on aspect of the generation stability of the generation process to prove that our methods can be adapt to use easily. First, we have record the generation speed per query, a question, answer or critique

is counted as a query, and error rate of total query. The generation speed and error rate that restart cause by the output not following the prompt can be seen at Table 4.5.

We've also done some research on the generation and critique performance of two different Chinese base model. As the result shown in 4.6, when we select BREEZE[5] as the **Critique LLM**, it is usually a more strict one compare to TAIDE[19]. So when we think that the dataset created when BREEZE[5] act as the **Critique LLM** should be more of a fit when we want our model to be professional. However, ther reason behind this need more research on the characteristic of the individual LLM. One can choose the best fit model to their usage.

The generation time for one fine-tune dataset takes less than 1 week and 4 to 8 hours depends on the ration of the good quality dataset. Comparing to a fully human annotated dataset usually cause weeks or months, this is very time efficient.

	speed	error rate
BREEZE	3.259s	0.161%
TAIDE	4 .280s	0.176%

Table 4.5: Average generation speed per query

Critique LLM Generate LLM	BREEZE	TAIDE
BREEZE	35.73%	88.45%
TAIDE	37.26%	79.91%

Table 4.6: Good Quality Ratio

## 4.5 Result

Here we will show the performance of our method on different model and settings. First, in figure 4.7 we see that the setting of TAIDE as **Generator LLM**, BREEZE as **Critique LLM** and BREEZE as **Final LLM** has the best performance. We think this

combination contributes from TAIDE has from TAIDE's greater parameters and more official document in the pre-train data, while the BREEZE has a more stable and keen model at the scope of in context learning. We've also test fine-tune dataset that does not contain the "Sorry, I do not know the answer" generate by the low quality questions in 4.8 since the evaluation does not contain such results. The without bad quality experiments result again prove our critique does work to preserve the good quality question-answer pair.

Second, we will use different setting with the best model we test in the prior experiment, which is TAIDE\_BREEZE\_BREEZE combination, as the vanilla LLM. We will be will mainly focus on doing experiments on four categories. First is the vanilla LLM without any fine-tuned and without RAG. Second is the Vanilla LLM only with RAG. Third is the fine-tuned LLM without RAG. Final is the Fine-tuned LLM with RAG. The final result in shown in figure 4.9. The result shown that using our method achieve at least 0.2 precision score, 0.16 recall score and 0.19 F1 score compare to any setting.

Finally, we've also do experiments on open data like PubMedQA[8] and PrivacyQA[15] and compare the F1 score on the Yes and No benchmark, the result can be seen in Table 4.10. We here replace our model to llama2[21] and all prompt to English and modify the desire output we need based on the dataset to fairly compare the result with the prior work [4, 14].

Generator	Critique	Final	precision	Recall	F1	Avg. Similarity
DI	BREEZE	BREEZE	0.6585	0.3011	0.4132	73.55%
BREEZE	DKEEZE	TAIDE	0.5804	0.2416	0.3412	<b>72.58%</b>
DKEEZE	TAIDE	BREEZE	0.6083	0.2714	0.3753	74.34%
IAIDE	TAIDE	0.5882	0.2402	0.3608	73.34%	
	BREEZE	BREEZE	0.6667	0.3048	0.4184	74.32%
TAIDE	DKEEZE	TAIDE	0.5391	0.2305	0.3229	73.70%
	TAIDE	BREEZE	0.6446	0.3000	0.400	75.00%
	IAIDE	TAIDE	0.4732	0.1970	0.2782	72.91%

Table 4.7: **Performance of different model.** This is the study on the performance of different language model act as Generator, Critique or Final LLM.

Generator	Critique	Final	precision	Recall	F1	Avg. Similarity
	BREEZE	BREEZE	0.7639	0.4089	0.5327	76.10%
BREEZE	DKEEZE	TAIDE	0.6692	0.3234	0.4361	72.49%
DKEEZE	TAIDE	BREEZE	0.7297	0.4015	0.5180	74.76%
	IAIDE	TAIDE	0.6929	0.3606	0.4743	72.77%
	BREEZE	BREEZE	0.7931	0.4275	0.5556	75.35%
TAIDE	DKEELE	TAIDE	0.7273	0.3569	0.4788	75.33%
IAIDE	TAIDE	BREEZE	0.7464	0.3829	0.5061	75.07%
	IAIDE	TAIDE	0.6434	0.3086	0.4171	74.28%

Table 4.8: **Performance of different model without low quality example.** This is the study on the performance of different language model act as Generator, Critique or Final LLM. In this set of experiment, we do not add the bad quality sample to see the performance on all positive evaluation set.

Final LLM	Precision	Recall	F1-Score	Avg. Similarity
BREEZE	0.5650	0.2577	0.3592	73.75%
BREEZE w/ fine-tuned	0.5882	0.2602	0.3608	74.13%
BREEZE w/ RAG	0.5726	0.2491	0.3471	73.96%
BREEZE w/ fine-tuned & RAG	0.7931	0.4275	0.5556	75.35%

Table 4.9: **Fine-tune performance.** This is the comparison of performance on the four settings.

	MedPubQA	PrivacyQA
TAPT	25.1	42.0
Bonito	27.0	56.9
Our Method	95.6	66.3

Table 4.10: **Open data performance(F1 score).** We compare our method in the F1 score with Gururangan [4] and Nayak 's [14] work on the open-source data.





# **Chapter 5** Conclusion

In this work, we have successfully leverage the power of open source Large Language Model as Generator LLM and Critique LLM to help us generate fine-tune dataset that can help us to fine-tune custom domain knowledge knowledge Final LLM to perform better on the factual question answering with the help of RAG system implemented. Our work is extremely flexible to be adapted to any source of documents as input and convert to fine-tune dataset. Also, it's modularized pipeline also allowed the user to replace any LLM inside the pipeline anytime a more competitive base model has emerged. Traditionally, creating a fine-tine dataset is very time consuming and require intense human labor, now with our work, any document can be turn into quality fine-tune dataset with almost no human interference and much faster.

In conclusion, our work can help any domain of expertise to fine-tune their own expert model to answer related question as long as there are existing documents knowledge.

As the experiments shown, we have helped the National Taiwan University to develop such system to increase the efficiency during the clinical time. with great response.





# **Chapter 6** Future Work

Since our work is very modularized, every parts can be improved as the technology improved. Though our work has shown great performance, we want to point out some parts we think that has great potential to be improve.

First, the RAG system, it's retriever is now also a hot topic of research, perhaps fine-tuning one specific for the pipeline and increase the hit rate would increase the performance. After all, our documents in the experiments isn't actually that much compare to normal RAG system and that is why the hit ratio could be 1.000 in the hit ration experiments.

Second, the generation process involves lots of prompt engineering to extracts the power of LLM no matter it's **Generator LLM** or **Critique LLM**. It is worth keep studying on how to come up with a more robustness and also general usable prompt to generate any format of response on the answer we need. The other part is to come up more and more critique index that help us evaluate the output of the LLM and determine what is actually usable. For example, give weight to different weight that weights differently would be a way of study.

Finally, since we proposed a three stage pipeline with three types of LLM to do their job, we could better come up with a recommendation of capacity a **Generator LLM**,

# Critique LLM or Final LLM should have.





# References

- [1] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.
- [2] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient fine-tuning of quantized llms, 2023.
- [3] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang. Glm: General language model pretraining with autoregressive blank infilling. In <u>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</u>, pages 320–335, 2022.
- [4] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, <u>Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics</u>, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

- [5] C.-J. Hsu, C.-L. Liu, F.-T. Liao, P.-C. Hsu, Y.-C. Chen, and D.-S. Shiu. Breeze-7b technical report. 2024.
- [6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. <a href="mailto:arXiv:2106.09685"><u>arXiv:preprint</u></a> arXiv:2106.09685, 2021.
- [7] INTSIG. aspire/acge text embedding, 2024.
- [8] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu. PubMedQA: A dataset for biomedical research question answering. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2567–2577, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.
- [9] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, et al. Matryoshka representation learning. <u>Advances in Neural Information Processing Systems</u>, 35:30233–30249, 2022.
- [10] Y. Labrak, A. Bazoge, E. Morin, P.-A. Gourraud, M. Rouvier, and R. Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains, 2024.
- [11] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. <u>Advances in Neural Information Processing Systems</u>, 33:9459–9474, 2020.

- [12] H. Li, Q. Dong, Z. Tang, C. Wang, X. Zhang, H. Huang, S. Huang, X. Huang, Z. Huang, D. Zhang, et al. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. arXiv preprint arXiv:2402.13064, 2024.
- [13] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. Mteb: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316, 2022.
- [14] N. V. Nayak, Y. Nan, A. Trost, and S. H. Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation, 2024.
- [15] A. Ravichander, A. W. Black, S. Wilson, T. Norton, and N. Sadeh. Question answering for privacy policies: Combining computational and legal perspectives, 2019.
- [16] K. Saab, T. Tu, W.-H. Weng, R. Tanno, D. Stutz, E. Wulczyn, F. Zhang, T. Strother, C. Park, E. Vedadi, et al. Capabilities of gemini models in medicine. <u>arXiv preprint</u> arXiv:2404.18416, 2024.
- [17] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, C. Kelly, N. Scharli, A. Chowdhery, P. Mansfield, B. A. y Arcas, D. Webster, G. S. Corrado, Y. Matias, K. Chou, J. Gottweis, N. Tomasev, Y. Liu, A. Rajkomar, J. Barral, C. Semturs, A. Karthikesalingam, and V. Natarajan. Large language models encode clinical knowledge, 2022.
- [18] H. Sun, Y. Liu, C. Wu, H. Yan, C. Tai, X. Gao, S. Shang, and R. Yan. Harnessing multi-role capabilities of large language models for open-domain question answering. In Proceedings of the ACM on Web Conference 2024, pages 4372–4382, 2024.
- [19] TAIDE. taide/llama3-taide-lx-8b-chat-alpha1, 2024.

- [20] B. Team. Bluelm: An open multilingual 7b language model. https://github.com/vivo-ai-lab/BlueLM, 2023.
- [21] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov,
  S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen,
  G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami,
  N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez,
  M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee,
  D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog,
  Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M.
  Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan,
  P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez,
  R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [22] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. <a href="mailto:arXiv"><u>arXiv</u></a> preprint arXiv:2212.10560, 2022.
- [23] Z. M. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, M. Zhang, et al. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. <a href="arXiv preprint arXiv:2310.00746">arXiv preprint arXiv:2310.00746</a>, 2023.
- [24] S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.
- [25] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, and D. Jiang. Wizardlm:

- Empowering large language models to follow complex instructions. <u>arXiv preprint</u> <u>arXiv:2304.12244</u>, 2023.
- [26] R. Xu, X. Wang, J. Chen, S. Yuan, X. Yuan, J. Liang, Z. Chen, X. Dong, and Y. Xiao. Character is destiny: Can large language models simulate persona-driven decisions in role-playing? arXiv preprint arXiv:2404.12138, 2024.
- [27] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, et al. Glm-130b: An open bilingual pre-trained model. <a href="mailto:arXiv:2210.02414"><u>arXiv:2210.02414</u></a>, 2022.
- [28] T. Zhang, S. G. Patil, N. Jain, S. Shen, M. Zaharia, I. Stoica, and J. E. Gonzalez. Raft: Adapting language model to domain specific rag. arXiv preprint arXiv:2403.10131, 2024.





# Appendix A — Prompt

# A.1 Prompt

## A.1.1 Generation prompt

## A.1.2 Critiques prompt

The individual prompt for groundedness, relevance, standalone and similarity mentioned in 4.2.3 can be seen in figure A.1, figure A.2, figure A.3 and figure A.4.

#### Groundedness Critique Prompt

你將會被提供一段內文 (Content) 以及一個問題 (Question)。你的任務是提供一個"評分"用來評比這個問題 (Question) 是否足以使用內文 (Content) 毫不含糊的回答。你的答案將會是  $1\sim5$  之間的整數,其中 1 代表根據提供的內文 (Content) 無法回答,5 則代表可以根據內文 (Content) 清楚明確的回答。回答格式如下評分: (一個  $1\sim5$  之間的整數) 評估: (請使用文字描述為何給出這個評分)

你"必須"在你的回答之中包含"評分:"與"評估:"兩段開頭。

接著是問題 (Question) 與內文 (Context):

問題 (Question): question 內文 (Content): context

Figure A.1: Groundedness Critique Prompt

### Relevance Critique Prompt

你將會被提供一個問題 (Question)。你的任務是提供一個"評分"用來評比這個問題 (Question) 是否是一個正常的人在向醫生問診的時候會向醫生詢問的問題。你的答案將會是 1~5 之間的整數,其中 1 代表這個問題不會被一個不舒服的患者所問出,5 則代表非常有可能被一個患者所問出。

回答格式如下評分: (一個 1~5 之間的整數) 評估: (請使用文字描述為何給出這個評分)

你"必須"在你的回答之中包含"評分:"與"評估:"。

接著是問題 (Question):

問題 (Question): question

Figure A.2: Relevance Critique Prompt

#### Standalone Critique Prompt

你將會被提供一個問題 (Question)。你的任務是提供一個"評分"用來評比這個問題 (Question)是否能獨立存在。你的答案將會是 1~5 之間的整數,其中1 代表這個問題需要搭配額外訊息才能回答,5 則代表這個問題本身訊息量已經充足。例如:如果問題指出特定環境如"在內文內文 (Content)中"或是"在檔案中",則評分必須是 1。問題之中可以常見的醫學名詞或是縮寫:例如說聽損、基因型、染色體。必須要是對於一個沒有醫療相關背景可以理解的內容。

使用繁體中文提供你的答案如下列格式:

回答格式如下評分: (一個 1~5 之間的整數) 評估: (請使用文字描述為何給出這個評分)

你"必須"在你的回答之中包含"評分:"與"評估:"。

接著是問題 (Question):

問題 (Question): question

Figure A.3: Standalone Critique Prompt

### A.2 Dataset

#### A.2.1 PubMedQA

The demonstration of the PubMedQA dataset can be seen in figure A.5.

# A.2.2 PrivacyQA

The demonstration of the PrivacyQA dataset can be seen in figure A.6.



#### Similar Critique Prompt

你將會被提供一個問題 (Question) 以及一個回答 (Answer)。你的任務是提供一個"評分"用來評比這個問題 (Question) 以及回答 (Answer) 是否有回答出品質夠高的問題。你的答案將會是 1~5 之間的整數,其中 1 代表這個問題與回答本身在語句結構上過於相似,並不構成一個正常的問題,5 則代表這個問題本與答案語句結構上並不會相近並且有很好的回答問題。

例如:如果答案 (Answer) 幾乎是在重複問題 (Question) 的內容而非進行實質的回答,請給出 1分。如果答案 (Answer) 能跟根據問題給出專業的回覆並且包含說明,請給出 5分。

使用繁體中文提供你的答案如下列格式:

回答格式如下評分: (一個 1~5 之間的整數) 評估: (請使用文字描述為何給出這個評分)

你"必須"在你的回答之中包含"評分:"與"評估:"。

接著是問題 (Question):

問題 (Question): question 回答 (Answer): answer

Figure A.4: Similar Critique Prompt



#### MubMedQA Example

"QUESTION": "Are group 2 innate lymphoid cells ( ILC2s ) increased in chronic rhinosinusitis with nasal polyps or eosinophilia?",

"CONTEXTS": [

"Chronic rhinosinusitis (CRS) is a heterogeneous disease with an uncertain pathogenesis. Group 2 innate lymphoid cells (ILC2s) represent a recently discovered cell population which has been implicated in driving Th2 inflammation in CRS; however, their relationship with clinical disease characteristics has yet to be investigated.",

"The aim of this study was to identify ILC2s in sinus mucosa in patients with CRS and controls and compare ILC2s across characteristics of disease.",

"A cross-sectional study of patients with CRS undergoing endoscopic sinus surgery was conducted. Sinus mucosal biopsies were obtained during surgery and control tissue from patients undergoing pituitary tumour resection through transphenoidal approach. ILC2s were identified as CD45(+) Lin(-) CD127(+) CD4(-) CD8(-) CRTH2(CD294)(+) CD161(+) cells in single cell suspensions through flow cytometry. ILC2 frequencies, measured as a percentage of CD45(+) cells, were compared across CRS phenotype, endotype, inflammatory CRS subtype and other disease characteristics including blood eosinophils, serum IgE, asthma status and nasal symptom score.",

"35 patients (40% female, age 48 00b1 17 years) including 13 with eosinophilic CRS (eCRS), 13 with non-eCRS and 9 controls were recruited. ILC2 frequencies were associated with the presence of nasal polyps (P = 0.002) as well as high tissue eosinophilia (P = 0.004) and eosinophil-dominant CRS (P = 0.001) (Mann-Whitney U). They were also associated with increased blood eosinophilia (P = 0.005). There were no significant associations found between ILC2s and serum total IgE and allergic disease. In the CRS with nasal polyps (CRSwNP) population, ILC2s were increased in patients with co-existing asthma (P = 0.03). ILC2s were also correlated with worsening nasal symptom score in CRS (P = 0.04)."

"LONG\_ANSWER": "As ILC2s are elevated in patients with CRSwNP, they may drive nasal polyp formation in CRS. ILC2s are also linked with high tissue and blood eosinophilia and have a potential role in the activation and survival of eosinophils during the Th2 immune response. The association of innate lymphoid cells in CRS provides insights into its pathogenesis.",

"final decision": "yes"

Figure A.5: Example of PrivacyQA Format



#### MubMedQA Example

'answer': 'Relevant',

'question': 'will my personal details be shared with third party companies?',

'text': 'The information may be disclosed to: (i) provide joint content and our services (eg, registration, coordination of membership accounts between the Viber corporate family, transactions, analytics and customer support); (ii) help detect and prevent potentially illegal acts, violations of our policies, fraud and/or data security breaches.'

Figure A.6: Example of PrivacyQA Format