DEGREE PROJECT

Chatting Over Course Material.

The Role of Retrieval Augmented Generation Systems in Enhancing Academic Chatbots

Hélder Monteiro

Master Programme in Applied Artificial Intelligence
2024

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

[This page intentionally left blank]

Abstract

Large Language Models (LLMs) have the potential to enhance learning among students. These tools can be used in chatbot systems allowing students to ask questions about course material, in particular when plugged with the so-called Retrieval Augmented Systems (RAGs). RAGs allow LLMs to access external knowledge, which improves tailored responses when used in a chatbot system. This thesis studies different RAGs through an experimentation approach where each RAG is constructed using different sets of parameters and tools, including small and large language models. We conclude by suggesting which of the RAGs best adapts to high school courses in Physics and undergraduate courses in Mathematics, such that the retrieval systems together with the LLMs are able to return the most relevant answers from provided course material. We conclude with two RAG-powered LLM with different configurations performing over 64% accuracy in physics and 66% in mathematics.

Preface

In this thesis, I explore retrieval-augmented generation systems (RAGs), which is an exciting technique for those working with or interested in large language models (LLMs) and are keen on augmenting their chatbots with external knowledge. Throughout the document, I walk you through the rationale for the experiments that I conducted and what they entail, and I conclude with some remarks on the results.

In the project, local LLMs were used: i) to generate synthetic question answer pairs on publicly available educational material from MIT's OpenCourseWare in order to experiment with different RAGs; ii) to run different RAGs, and iii) to evaluate the results.

The hope is that the results presented here are meaningful and can be used to further provide an understanding of RAGs, tailored for education material, such as class notes, videos, and audio.

Contents

List of Figures

List of Tables

1 Introduction

Since the rise of ChatGPT, there has been a lot of hype around Large Language Models (LLMs) and chatbot systems. These technologies have enabled us to improve our workflow (e.g. GitHub Copilot as code completion tool), and even being used as a study companion. LLMs are often seen as giant statistical models (Rosenfeld, 2000) trained on billions of texts from the internet made available through projects like Common Crawl[1]. They are capable of not only generating text in multiple natural languages but also code, do machine translation and even summarize texts.

There has been different research conducted within the use of LLMs in education settings (Vacalopoulou et al., 2024; Alexandra Farazouli and McGrath, 2024; Yu, 2023; Latif et al., 2024; Xiao et al., 2023; Nechakhin, D'Souza, and Eger, 2024; Yen and Hsu, 2023) with different focuses including mathematical learning (Yen and Hsu, 2023) and impact in teachers' assessments (Alexandra Farazouli and McGrath, 2024). In some cases, the use of LLMs is encouraged, such as at Stanford University's "Creativity and Design Thinking Program"[2] course, where students submit the prompt they used that gave rise to the solution, thereby evaluating their creativity in writing prompts (Klebahn and Krakowski, 2023; Leung[1] and Lo, 2024). Different universities and teachers see the tool differently, either as an enabler for better education (Gašević, Siemens, and Sadiq, 2023) or as a detractor for effective learning (Srishti, 2024), whereby students become dependent on the tools.

The research in this field has progressed at a steady pace, and has seen the rise of open-source LLMs and techniques to augment their knowledge using domain specific data. The availability of such models made its adoption on consumer hardware much easier, thanks to affordable Graphics Processing Unit (GPU) cards and techniques aimed at compressing them for inference on Central Processing Units (CPUs) only. This means that anyone with a decent computer can have locally run LLMs and chatbots that are powerful enough to generate text and allow customization for different tasks.

When it comes to augmenting the knowledge of an LLM, one idea is to use what is called retrieval augmented generation (RAG) system, which simply put is a way to integrate external knowledge into the LLM using different tools. This knowledge can come from different sources: documents, databases, media files, the internet, etc., such that the model can access them beforehand, in a preprocessed form, and create the

[1] `https://commoncrawl.org/`

[2] `https://online.stanford.edu/how-you-can-use-chatgpt-increase-your-creative-output`

means to take the user query and retrieve relevant answers that are then returned to the user in a polished way. To make knowledge searchable, the text has to be tokenized and converted into embeddings, which is a vector space representation of the text that enables semantic search with the user query.

In academia, RAGs-enabled chatbots can be an effective way to provide students with a full-time virtual teaching assistants (VTA), which would be available 24/7 to discuss class material and enhance learning among students. The question is: how can we navigate through various RAG systems and experiment with what works best so that academic chatbots can be improved to enhance student learning? Additionally, does it matter which open-source models are used, given that when chatting over class material, we are only interested in responses that match the external knowledge that we provide to the model and not the internal knowledge the model possesses from its training data? These question hasn't been discussed in the literature, but we believe an effective teaching assistant should know enough about the course material as provided by the human professor — for instance, in the learning platform, class material like lecture PDFs, PowerPoints, course literature, and media files — the VTA should be in a position to answer questions surrounding it. This would enable the student to learn more effectively and master the course material thoroughly.

Thus, the aim of this thesis project is to experiment with different RAG systems, taking into account course material related to subjects within Physics and Mathematics, to try to find the best and most effective RAG and open-source LLM that can be used within a chatbot system to enable effective learning for students taking these subjects. We do this by:

1. **Creating synthetic question-answer data**
   We start by creating synthetic question & answer pairs with local LLMs, based on the course material of interest: physics and mathematics. This data is used to test different RAGs to see if their response matches the ground-truth created from the synthetic process.

2. **Setting up the experiments**
   We choose different parameters, models both small and large, and two different semantic databases, and we compare the results to see how they perform for physics and mathematics.

3. **Choosing an orchestration tool**
   LLMs require orchestration tools to make them usable in practice. So, choosing the right open-source orchestration tool is important.

4. **Integrating tools for ingesting PDF documents**
   Here we enable LLMs with RAG within a chatbot scenario to fetch knowledge from PDF documents. This gives the model the ability to use specific domain data and ingress them into a searchable form.

We strive to focus on open-source tools, as we believe that this is where the direction

will be in the future, rather than using commercial tools that do not take into account user privacy and comes at a price tag.

1.1 Goals

In this thesis project, our goal is to investigate an effective RAG system to power local LLMs for academia to help students study more effectively. We conduct a series of studies to investigate what has been done and argue against or in favor of existing solutions. We present our results of testing different combinations of RAGs with different parameters,. Students would benefit from having a virtual teaching assistant with the effective RAG that we identify in this project. Such assistant shall be able to answer questions related to class material.

1.2 Outline

This thesis is structured as follows: in chapter 2, we focus on the background and related work within RAGs, local LLMs, and semantic search, and in chapter 3, we explain how we will implement the system and test it in the usual way, which would include creating sets of question and answers and compare the system response. In chapter 4 we provide the results and performance evaluation from our system. And finally, we conclude in chapter 5 with a brief discussion on the system, and future work.

2 Background and related work

In this section, we introduce the background of the topic and related work. In section 2.1, we provide an overview of natural language processing (NLP) and large language models (LLMs), then in section 2.2 we further go over open-source LLMs commonly used to date. Then, we move over related work in chatbots in education in section 2.3 followed by RAG techniques in 2.4 and approaches to synthetic data generation in section 2.5.

2.1 From NLP to LLMs

Computers understand numbers and we want them to understand human language so we can use it for different tasks. We use Natural Language Processing (NLP) to enhance their ability to handle text data for different applications like scene understanding (Mokayed et al., 2020; Hum et al., 2022; Mokayed et al., 2014; Chai et al., 2016; Mokayed et al., 2021; Mokayed et al., 2022), banking and documents(Khalid, Yusof, and Mokayed, 2011; Nikolaidou et al., 2023; Alkhaled and Fei, 2023), IoT and smart cities (Javed et al., 2023b; Mokayed et al., 2023; Mokayed et al., 2024), and many others fields. In NLP, we take text of different types (multilingual text (Wang et al., 2023), audio, programming languages, etc.) and tokenize it by creating units or subunits called tokens that can be mapped to numbers. In some cases, we process text corpora and tokenize them into sequences (or sentences) depending on the downstream task. A standard practice in NLP is to keep track of the tokens created during tokenization by building a dictionary that maps them to numbers - this is also called the vocabulary. Common NLP application tasks include *sentiment analysis* (Nasukawa and Yi, 2003) for instance in text derived from product reviews or social media posts to understand how positive they are, *semantic similarity* (Curran, 2004; Li et al., 2006) common in comparing texts e.g., between a user query and a database to return matching results, *text summarization* (Radev, Hovy, and McKeown, 2002) and *machine translation* (Javed et al., 2023a; Allen, 2003) where we use the computer to translate between human languages. Despite many useful applications with natural language, NLP techniques are still limited due to inherent ambiguity of human language. For instance we might face lexical ambiguities where a word can have different meanings (Almeida and Libben, 2005) or syntax ambiguity where the sentence can be interpreted in different ways (Pickering and Van Gompel, 2006). These issues are always present and there is a whole area of research focused on language disambiguation (Ide and Véronis, 1998; Resnik, 1999; Banko and Brill, 2001;

Navigli, 2009).

A more recent approach to handle human language and potentially handle such ambiguities is the use of language models, either "small" or "large" language models (SLMs and LLMs respectively). These models are trained on massive amounts of text data, with tokens ranging from millions to trillions, and have one main characteristic: the ability to predict the next likely word. There are different types of language models, from classic ones such as *bags-of-words* (BoW) (Gale, Church, and Yarowsky, 1992; Jurafsky and Martin, n.d.) which captures frequency of individual token from the corpus it was built-on, the *n-gram model* (Jurafsky and Martin, n.d.) which captures the frequency of n-consecutive words, the *Recurrent Neural Networks* (Sutskever, Martens, and Hinton, 2011; Sutskever, 2013) and *Long Short-Term Memory* (LSTM) (Hochreiter and Schmidhuber, 1997) both of which handle sequence data and their context, where in LSTM the context is longer. However, none of these models are considered "large" in the traditional sense until the *transformer* architecture (Roy et al., 2023; Lieb and Goel, 2024) came in 2017, which included what is called self-attention mechanism (Vaswani et al., 2017) to compute the importance of individual tokens within large context windows. This architecture gave rise to well known large language models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019; Javed et al., 2022) and Generative Pre-trained Transformer (GPT) (Radford et al., 2018) including their variants ALBERT (Lan et al., 2020), RoBERTA(Liu et al., 2019), DistilBERT (Sanh et al., 2019) and TinyBERT (Jiao et al., 2019).

2.2 Open-source LLMs

Large Language Models (LLMs) have seen huge popularity with the release of OpenAI's ChatGPT[1] in the late 2022 (Minaee et al., 2024). The race to release proprietary LLMs had started thereafter with technology giants such as Google and Microsoft releasing their own proprietary chatbots. Microsoft based its Bing AI chatbot on OpenAI's GPT models (Gwon et al., 2024; Devi, Manjula, Pattewar, et al., 2023) whereas Google created its own called Bard (Ram and Verma, 2023), which was erroneous and far behind ChatGPT (Ahmed et al., 2023; Gunes and Cesur, 2024; Aydin and Karaarslan, 2023; Borji and Mohammadian, 2023) despite having internet access (Ahmed et al., 2023). On the other end, Meta AI developed the Llama (Touvron et al., 2023) series of open-source models, which were based on the transformer architecture (Vaswani et al., 2017) from Google, in use by most LLMs both open-source and proprietary. The first and subsequent LLama models resulted in many finetuned variants from the research community - these includes Stanford Alpaca (Taori et al., 2023) which was finetuned from LLaMA 7B using synthetic data generated by OpenAI's text-davinci-003 (Taori et al., 2023) model. This data was generated through a technique called Self-Instruct (Wang et al., 2022) which focus on using 175 human-annotated instructions (Taori et al., 2023; Wang et al., 2022) that are passed to a Teacher LLM to generate thousands of variants. Other Llama

[1] `https://en.wikipedia.org/wiki/ChatGPT`

variants includes MEDITRON (Chen et al., 2023) which focus on the medical field; the Vicuna models (Zheng et al., 2024) finetuned for chatting, and Wizardcoder (Luo et al., 2023) finetuned for coding tasks. However, most Llama models and variants cannot be freely used commercially without few restrictions imposed such as "user base of no more than 700 million monthly active users" (Roumeliotis, Tselikas, and Nasiopoulos, 2023) and restrictions on its use to generate synthetic data to fine tune models of different base architecture (Meta Platforms, Inc., 2023). Such restrictions should be considered when picking an open-source for commercial use, to ensure no limitations arise due to licensing issues.

There are other models that allow fully commercial use without restrictions. For instance, EleutherAI's Pythia (Biderman et al., 2023) series of models trained on The Pile's 825 GB (Gao et al., 2020) dataset and its variants such as Databrick's Dolly (Conover et al., 2023) are more permissive. Other base models also includes Mistral (Jiang et al., 2023; Jiang et al., 2024; Team et al., 2024) and its various sizes. Besides LLMs, there has been a focus on also training small language models (SLMs) (Fu et al., 2024; Zhang et al., 2024; Abdin et al., 2024) for more narrow tasks. For instance Zhang et al., 2024 introduced TinyLlama, which has the same model architecture as Llama models, but with just 1.1 billion parameters. On the other hand, Team et al., 2024 introduced Gemma whereas Abdin et al., 2024 introduced Phi-3 series of models with 2 and 3.8 billion parameters respectively. Such models provides an excellent platform to quickly finetune an LLM for more narrow tasks.

2.3 Chatbots in Education

Chatbots can be used for different applications. We focus on chatbots in education, with a virtual teacher assistant (VTA) that communicates with the student on-demand. The student can upload class material in the form of PDFs and ask questions about any topic within the uploaded documents and the chatbot replies with the context it extracted from the material.

There are commercial chatbots that enable the functionality of uploading documents and asking questions about them. For instance OpenAI's ChatGPT-4 and ChatGPT-4o (OpenAI et al., 2024) has a Retrieval-augmented generation (RAG) system (Vacalopoulou et al., 2024; Ferber et al., 2024) that can enable internet search or question and answering on documents. However, it is not an ideal platform to use in an education setting because it is not free to use or there is a cap in utilization. The tool provides RAG customization when ran using "custom GPTs" which requires a premium subscription to use. This is a drawback, not just for its cost, but on the reliance of third-parties to handle student data and the risk to threaten academic integrity (Shoufan, 2023) given that students would still be able to use the normal ChatGPT, which may not encourage the use of a customized chatbot for chatting over class material.

Lieb and Goel, 2024 introduces NewtBot, an academic chatbot for physics at secondary education. It is powered by GPT-3.5 model via REST API and uses a combination of system prompting to have the chatbot to be in either "Tutor" mode where the

assistant engages the student with questions in an encouraging manner, or "Feedback" mode to provide accurate and relevant feedback, while the system is useful, given that it is independent of ChatGPT UI, it still costs money to pay for REST API calls, and does not have a RAG system for chatting over class material, and does not use open-source LLMs which can be easily customizable for similar application.

Farah et al., 2023 discusses the same application area as Lieb and Goel, 2024 but focusing on teaching software engineering best practices. It also focuses on using OpenAI's GPT-3 to power the chatbot via REST API and does not use any RAG. However, as it is guided by prompts, such that the chatbot only answers questions related to the course content. But given that these chatbots can suffer from indirect prompt injection (Greshake et al., 2023), if attacked, students can ask more questions outside the scope of the system prompt defined by the Teachers.

Neupane et al., 2024 introduces BarkPlug v.2 chatbot which answers to users queries about campus resources at Mississippi State University. The system uses RAG to retrieve relevant information before generating a response, the latter of which is managed by OpenAI's GPT-3 REST API. The authors state that the system is prone to hallucinate if the RAG system fails. This is probably due to how the RAG system is designed and the way the system prompt is defined. The normal is to not provide any response if no context is given, or ackowledge that the query cannot be answered. Such checks are important in particular when paid REST API endpoints are used to avoid incurring unnecessary costs.

Maryamah et al., 2024 discusses chatbots in academia and includes different evaluation metrics. Their system includes RAGs, but uses commercial LLMs to generate response. Nonetheless, they provide excellent source of information on how to evaluate the retrieval and generation of the system using metrics such as BLEU and ROUGE scores for the generated answers.

Faruqui et al., 2024 present a protocol to develop SAMCares which is an adaptive chatbot that uses LLama-2 70B which is an open-source model. The system uses RAG to access and retrieve course content from Sam Houston State University's LMS and provide tailored educational support to students. However, the use of a giant model with 70 billion parameters, is too far outreached to what we plan to achieve, given that such models even if quantized down to 1 bit, would be extremely slow to operate without a powerful GPU. We believe that there is no valid reason to use bigger LLMs, unless we aim at using its internal knowledge together with the RAG. In such cases, bigger models can be useful, but not practical for running in consumer hardwares.

Zimmermann and Rohrer, 2024 introduces Study Buddy which is an education chatbot focusing solely on open-source technologies and RAGs techniques. This project compares different approaches to run LLMs locally, and suggests that Ollama[2] is a better approach as it adapts to different "hardware configuration" (Zimmermann and Rohrer, 2024) which is important when considering building such applications for academic setting.

[2] `https://ollama.com`

2.4 RAG Techniques

The Retrieval-Augmented Generation (RAG) system is an essential component to enhance knowledge to a large language model (LLM). Having a RAG-powered LLM addresses issues such as outdated information and hallucinations (Ding et al., 2024) which would limit the LLM in providing relevant answers to the user, particularly in the context of a chatbot. The RAG system works through a combination of data indexing, retrieval and generation (Gao et al., 2024) in an end-to-end fashion.

In their survey paper, Gao et al., 2024 categorize RAGs into three types: Naive, Advanced and Modular. The naive consists solely in common pipelines for indexing, retrieving and generation. The advanced, optimizes the user query by rewriting it (Peng et al., 2024), so that relevant information is retrieved. This is useful when the query from the user is too wordy or unclear that may affect the system retrieval capability when searching for similar texts. The modular RAG is more customizable to integrate with different components within the RAG pipeline.

Es et al., 2023 proposes Ragas, a framework to evaluate RAG pipelines using metrics such as faithfulness, answer and context relevance. The framework can be used to generate synthetic data that can then be used to test RAG systems. Through their documentation[3], Ragas seem to require OpenAI's REST API to generate the data, there is no mention of usage with local LLMs but the framework gives an idea of what is possible, when it comes to evaluating RAG systems.

Salemi and Zamani, 2024 introduces eRAG, another framework to evaluate RAG pipelines. This tool evaluates returning answers from the RAG system against their ground-truth, meaning that if the retrieval system returns k responses, each of them is evaluated against the ground-truth and assigned a label, before the final answer is returned to the user. This differs from a more direct evaluation approach for instance, the one from Roucher, n.d. which evaluates solely on the final response from the RAG and not on each response from the retrieval system. Nonetheless, the authors argue that their framework is more efficient that existing approaches.

2.5 Synthetic Data Generation

Synthetic data generation is a common practice within the AI practice (Puri et al., 2020; Shakeri et al., 2020; Riabi et al., 2020; Alberti et al., 2019; Wang et al., 2022; Roucher, n.d.) to reduce reliance on human annotations which can be expensive.

Riabi et al., 2020 introduces an approach to cross-lingual synthetic data generation, making use of English question and answer (QA) model and translate the generated pairs into multiple languages. The data is then used to train better multilingual QA models. The authors say that their approach outperforms English-only baseline models (Riabi et al., 2020).

Shakeri et al., 2020 build on top of the SQuAD (Rajpurkar et al., 2016) dataset and generate additional synthetic QA pairs using a transformer-based model. The model

[3] `https://docs.ragas.io/en/stable/concepts/testset_generation.html`

not only generates the data but also filters the best candidates using likelihood score (Shakeri et al., 2020).

Puri et al., 2020 uses GPT-2 model to generate synthetic QA pairs. They breakdown text into paragraphs and pass those to an LLM to generate QA pairs. Quality checks are done using BERT (Devlin et al., 2019; Javed et al., 2022), which filters irrelevant couples (Puri et al., 2020). Similar approach is done by Roucher, n.d. which uses Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024) model having 56 billion parameters. The work of Roucher, n.d. slightly differs on that of Puri et al., 2020, where Mixtral model is used for everything: question and answer generation and evaluation, both of which done through prompts. The evaluation consists of three metrics: groundedness, relevance and standalone scores. The groundedness evaluates the truthfulness of the generated pair given the retrieved context. The relevance relates to the domain of the data, for instance if we want to evaluate the relevance for physics and mathematics, the model will be asked to assign a score based on the relevance for these fields. The standalone metric scores the QA pair on whether there is implicit mention of context in the question, which might indicate low quality of question generated. All these metrics by Roucher, n.d. take on the values between 1 to 5, which are then filtered out for a minimum of 4 across the three metrics.

These studies are important for our project, specially the work of Roucher, n.d. as it can be adapted to run with slightly smaller language models within a consumer hardware, for instance Llama 3 8B[4](AI@Meta, 2024) to generate and evaluate synthetic data, as well as test various RAG systems.

[4]`https://en.wikipedia.org/wiki/Llama_(language_model)`

3 Materials and Methods

3.1 Synthetic Data Generation

Since our goal is to experiment with different RAGs, we should have question and answer pairs to evaluate each RAG that we construct. In figure 3.1 we show the schematic of the pipeline to generate synthetic data.

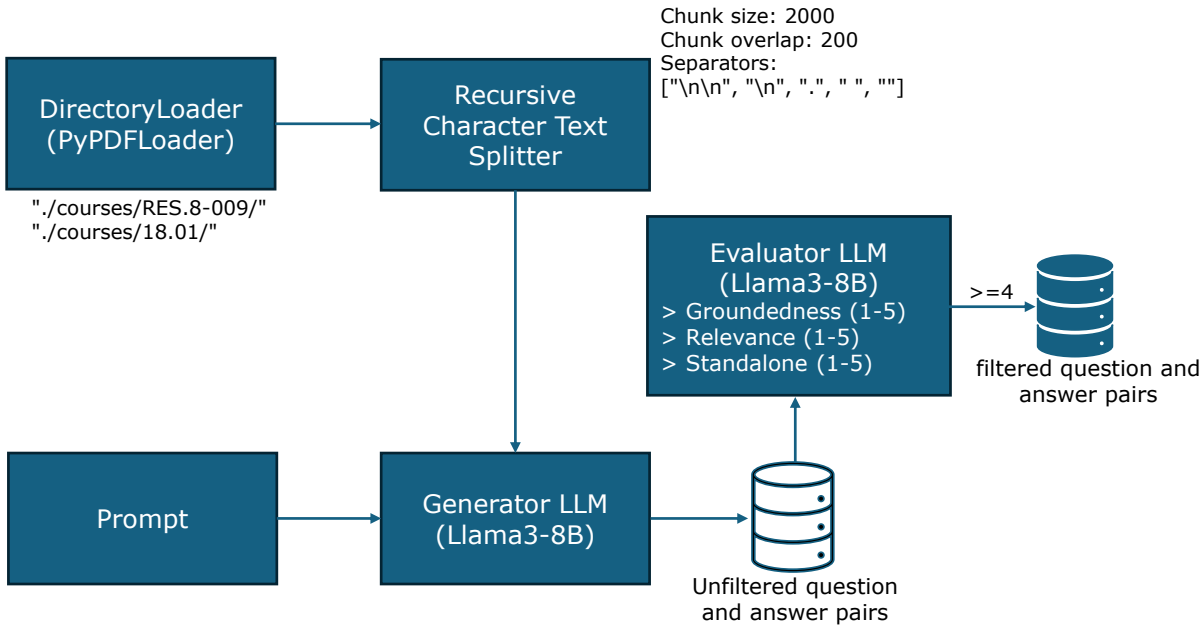

Figure 3.1: Schematic of the pipeline to generate synthetic data.

We follow the work of Roucher, n.d. that uses an open-source LLM to generate synthetic data. The author uses Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024) model whereas in our project we use Llama3-8B model as it has shown better performance compared to earlier variants of the series with comparable size (AI@Meta, 2024) and it can be run on the available computational resources that we have. We modify the code to allow loading PDF documents from a directory using DirectoryLoader module from

Langchain[1] where we pass PyPDFLoader module as class to guide DirectoryLoader that the expected files are of PDF type and that PyPDFLoader should be used as a parser. As seen from figure 3.1, after each course folder is loaded separately, the next step is to split the data. We use Langchain's Recursive Character splitter with the same parameters as Roucher, n.d., that is, chunk size of 2000 and chunk overlap of 200. The list of separators are default. These parameters were kept to ensure enough text is retrieved (e.g. 2000 characters per chunk, with overlap between chunks of 200 characters). The separators are the means to split the text, first starting with double newlines, followed by newline, full-stop, space and character level (no space). Each of the separators are such that the splits have the most text within the maximum allowed chunk size, and the splitter iterates over the separators to find the best that keeps relevant chunks together.

For the generator LLM which generates the QA pairs, and the evaluator LLM which provide scores to the QA pairs across three metrics (groundedness, relevance and standalone) were used with Llama3-8B model using the same prompts of Roucher, n.d. However, the prompt for relevance, we modified to include that the scoring should ensure that the synthetic questions are relevant for physics and mathematics.

- **Relevance score (Physics)**
  The relevance score is given depending on how useful this question can be to high school seniors taking the course Introduction To Oscillations And Waves.
- **Relevance score (Mathematics)**
  The relevance score is given depending on how useful this question can be to undergraduate students taking the course Single Variable Calculus.

When the question pairs are scored, we filter them to only retain those with score greatet than or equal to 4.

3.2 Tools and Languages

Throughout the project we use Python[2] programming language to generate synthetic data and test multiple RAG pipelines. Orchestration tools are required to allow us to make use of the LLMs for building applications. Most components needed for building RAG-powered LLMs are provided by the orchestration tool. These include components for text splitting, retrieval and storage in semantic databases. Usually these components are third-party libraries that are integrated into the orchestration tool. For our project, we use LangChain as orchestration tool as it is one of the easiest and comprehensive tools that current exists besides LlamaIndex, for programmatically interact with LLMs.

In order to run local LLMs we use Ollama as it best optimizes running local LLMs for different hardwares (Zimmermann and Rohrer, 2024) with or without GPU, and for

[1] `https://en.wikipedia.org/wiki/LangChain`
[2] `https://www.python.org/`

its simplicity and ease of integration with different orchestration tools like LangChain. Ollama works in a similar manner as Docker, allowing to "pull" models from a repository and running them as local LLM REST APIs.

3.3 Experiment Design

Major part in the project consist of experimentation. First by generation of synthetic QA data and then evaluation of different RAG systems that we carefully designed considering time and resource constraints. To generate the synthetic data and test our RAG, we focus on subjects related to undergraduate Mathematics course on Single Variable Calculus (Jerison, 2006) and high school Physics course on Introduction to Oscillation and Waves (Williams, 2017) both from MIT's OpenCourseWare. We picked these subjects as we find that designing RAG-powered LLMs for mathematics and natural science subjects like Physics are more interesting from application point-of-view as these subjects are hard to master and thus it would be useful to students in general to enhance their learning with a RAG-powered LLM tailored for this type of educational material.

In table 3.1 we have the parameters used in our experiments. Considering Cartesian product of the count of each parameter, we have a total of 64 scenarios for RAGs that we test each per course subject. For detailed combinations, see table A.1 in the appendix.

Parameter	**Values**
Chunk Sizes	500, 1000
Overlaps	50, 100
Vector stores	Chroma, FAISS
Models	Phi3, Llama3
Embedding Models	mxbai-embed-large, llama3
Text Splitters	CharacterTextSplitter, RecursiveCharacterTextSplitter

Table 3.1: Experiment parameters used in the study

The choice of chunk sizes, the idea was to have a balance between small (500) and large (1000). The overlaps were chosen in similar fashion, small (50) and large (100). The vector stores is where we store the external knowledge to do semantic search. We used two that are popular, Chroma[3] and Facebook AI Similarity Search (FAISS) (Douze et al., 2024). These are used without any customization, meaning that we use default parameters when using them in our experimentation pipeline, as we are focusing on finding the best RAG solely using default parameters as they are, as these can be fine-tuned later on once the RAG is in use.

The models we use are Phi-3 (Abdin et al., 2024) from Microsoft and Llama-3 (AI@Meta, 2024) from Meta AI. These two models provide a good balance between small (3.8B parameters in Phi-3) and large (8B parameters in Llama-3) so we can study if model size affects the generation quality within the RAG-powered LLM. The same can

[3] `https://docs.trychroma.com/`

be studied with the embedding models which are responsable for creating a vector space for which semantic search can be carried out. We test two models mxbai-embed-large (Sean Lee, 2024; Li and Li, 2023) from MixedBread AI that has only 335M parameters and Llama-3. The LLMs should be passed with prompts to guide them through the task. The following prompt was used in the experiments:

```
Answer the question using only on the provided context.
Only respond to what was asked without repeating the question.
The response should be concise and 'straight to the point'.
If you are unable to answer the question, say "I don't know".

Context:
{context}

Question:
{question}
```

For text splitting, we are using character-level and recursive character-based text splitters. The difference between them is that the character-level splitter splits chunks of text on each character, whereas recursive splitter allow us to decide on a list of text separators to consider, where each of one is tried until chunk size usage is maximized. If a recursive character splitter has empty string separator, it becomes a character splitter.

Evaluation of the RAGs are done using the prompt with the five scores proposed by Roucher, n.d., ranging from 1 for *completely incorrect/inaccurate* to 5 for *completely correct/accurate*. The prompt are passed to the local LLM, in our case LLama3-8B that acts like a judge on the generation quality of the RAGs compared against the ground-truth answers. In order to calculate accuracy, we choose score of 3 as the cut-off for accurate results as the the score implies *somewhat correct/accurate* response from the RAG.

4 Results

In this section, we present the results of the experiments, where we ran a pipeline to test different RAG combinations.

4.1 Synthetic QA data

We generated a total of 183 QA pairs for physics and 200 for mathematics. After filtering for relevance, groundedness and standalone scores greater than equal to 4, we obtained 119 QA pairs for physics and 137 pairs for mathematics, see figure 4.1 for details.

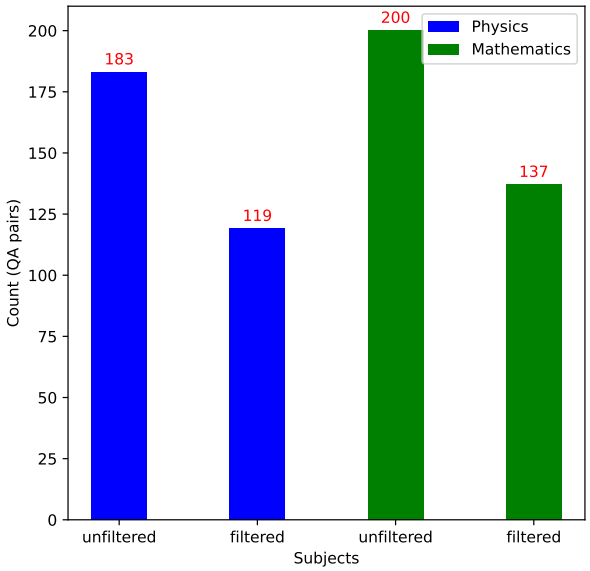

Figure 4.1: Count of QA pairs generated.

This generated data was then used to test the 128 RAGs that we created with different parameters.

4.2 Retrieval capability

After running the generated data for each of the 128 RAGs, we obtained interesting results. For Physics RAGs (see figure 4.2), the maximum accuracy was 64% and that was achieved with RAG #2 having chunk size of 500, overlap 50, chroma as vector store, CharacterTextSplitter as text splitter, and embedding model was mxbai-embed-large and main LLM was Llama-3.

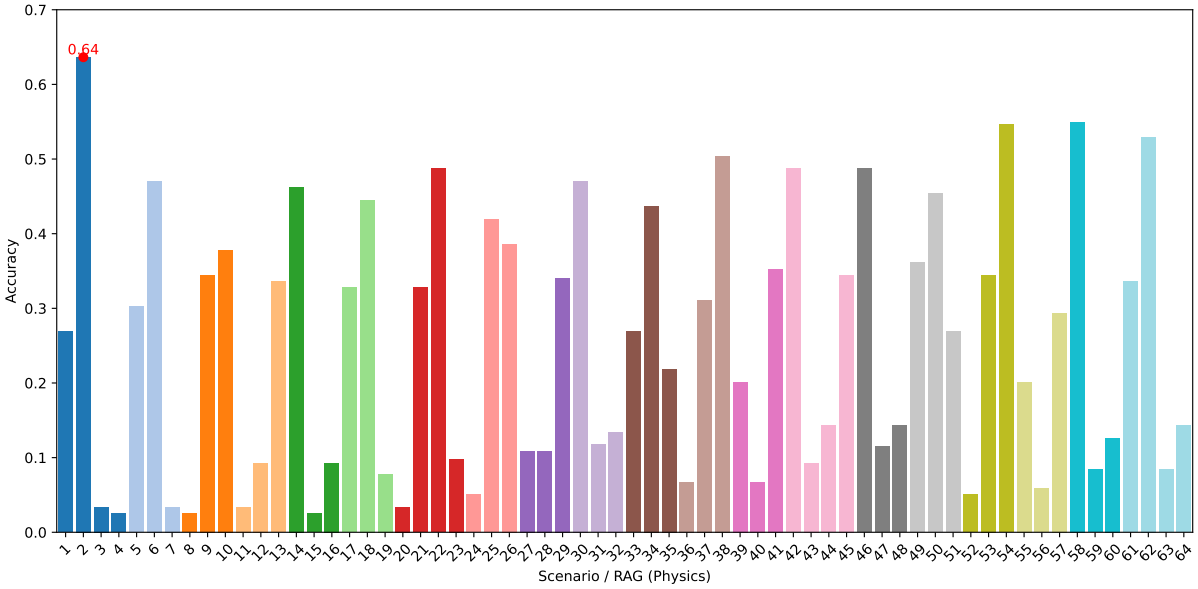

Figure 4.2: Accuracy per each Physics RAG (See table A.1 for detailed configuration of the RAGs shown the figure).

For mathematics RAGs (see figure 4.3), 66% maximum accuracy was achieved for RAG #57 having chunk size of 1000, overlap 100, Chroma as vector store, RecursiveCharacter as text splitter, and embedding model was mxbai-embed-large and main LLM was Phi-3. This was completely opposed to what we obtained for the Physics RAGs.

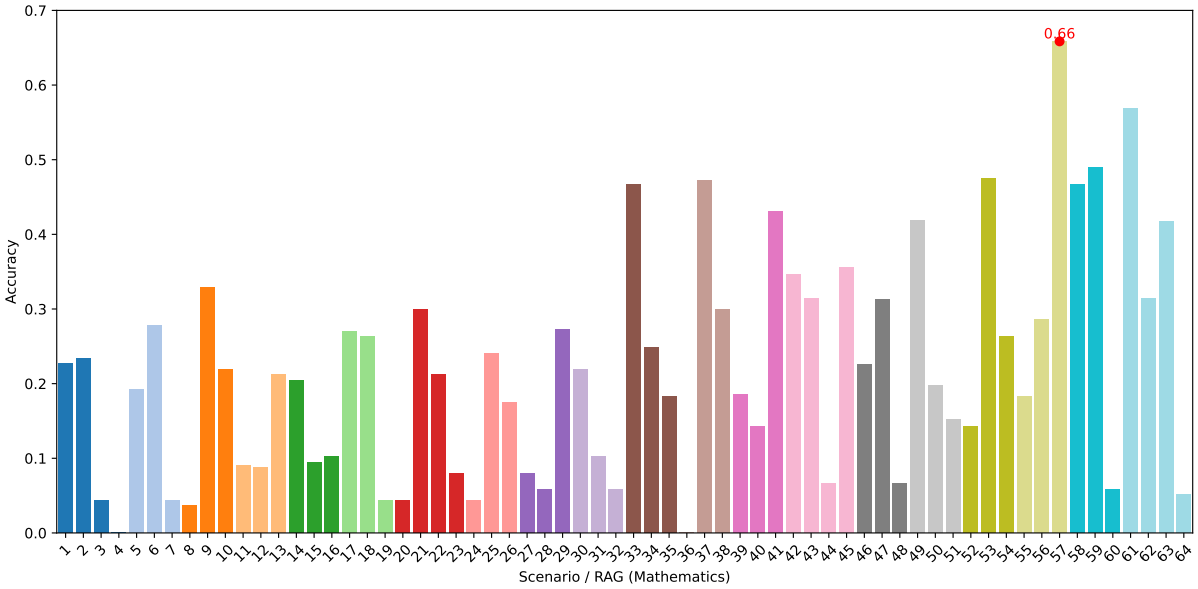

Figure 4.3: Accuracy per each Maths RAG (See table A.1 for detailed configuration of the RAGs shown the figure).

4.3 Q&A Evaluation

For the highly performing RAGs for physics and mathematics we plotted the character count for the question created during synthetic data generation process, and compare the count of the ground-truth answer and the answer returned by the individual RAGs. From figure 4.4, we see that the RAG answers are of comparable size to the ground truth, even though there are some picks in either of them. On the other hand, the character count for mathematics RAG (see figure 4.5) that performed best, the number of characters count is highest in most cases for the generated answer. The ground-truth answers were relatively short.

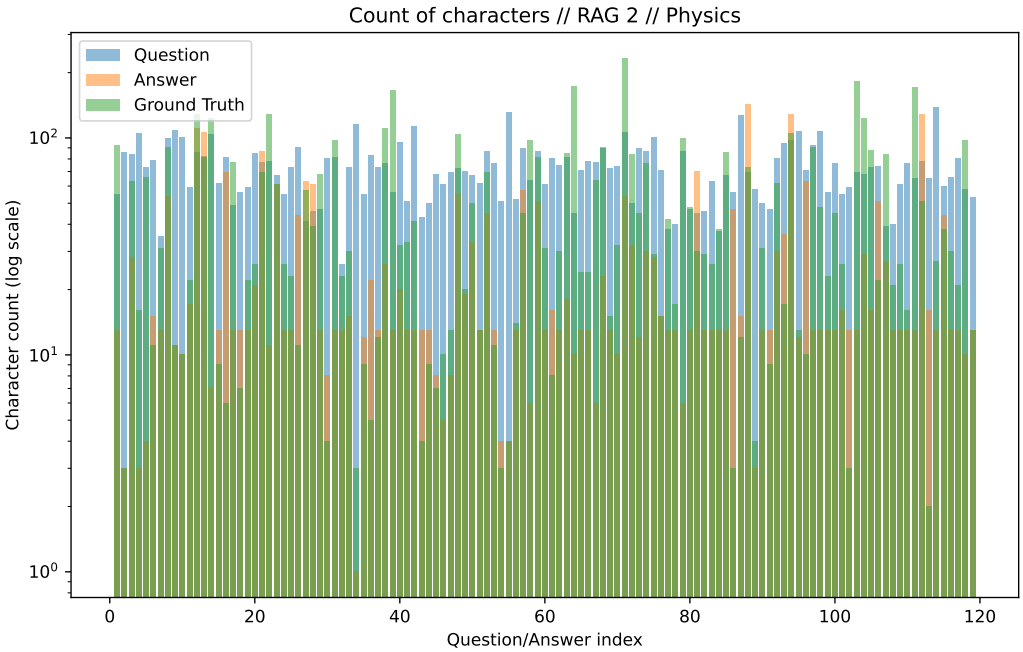

Figure 4.4: Count of characters (log-scale) of question, answer (RAG) and ground-truth answer for the high performing Physics RAG (#2).

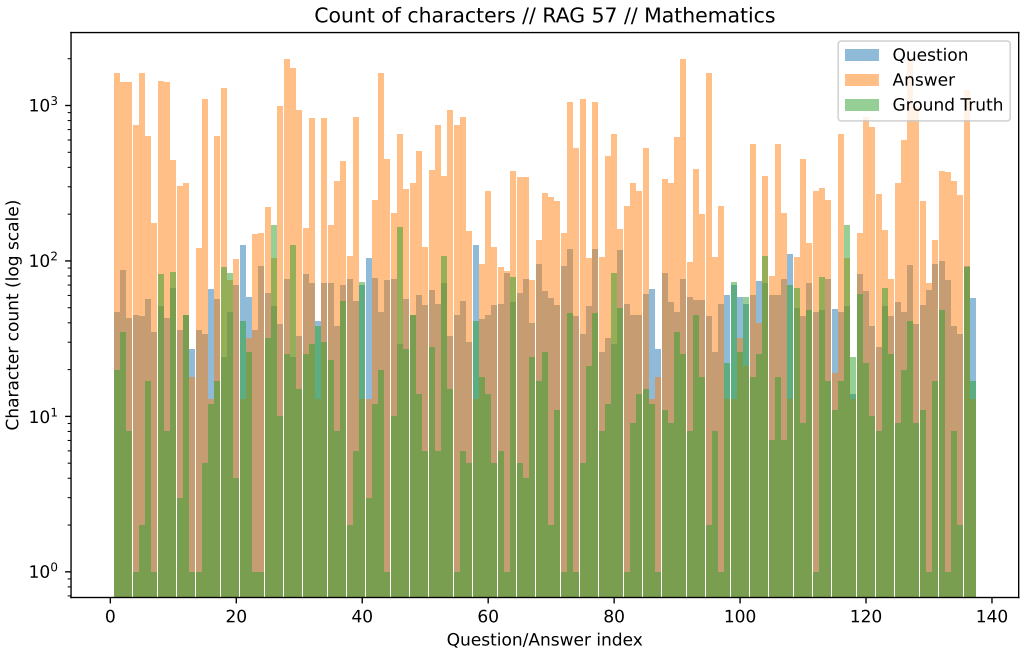

Figure 4.5: Count of characters (log-scale) of question, answer (RAG) and ground-truth answer for the high performing Mathematics RAG (#57).

5 Discussion and Conclusion

5.1 Discussion

In the experiments, we have seen that a RAG for physics is not optimal for mathematics. The RAG configuration #2 showed the highest accuracy of 64% for physics, but the same configuration gave only 23.35 % accuracy for mathematics. However, configuration #57 for mathematics had over 66% compared to 29.4% for physics on the same configuration.

Analysing the character counts for the best performing RAG for physics (see Figure 4.4), we can see that the answer and ground-truth are close to each other in character lengths. No strange behavior is noticeable. For the same configuration (#2) but for mathematics (see Figure B.1 in appendix), we see that most of the answers have 10 characters. This is due to the retrieval system being unable to match the query with the knowledge, thereby returning "I don't know" as answer. An interesting aspect in the data is the fact the ground-truth have cases of very short responses of less than 3 characters. Some of these cases are highlighted in table A.2 in appendix, where in many cases RAG #2 for mathematics replies with "I don't know". These cases reflect on the fact that some questions in mathematics that were generated synthetically required giving one answer.

Better considerations should be given when generating synthetic data, that the QA pairs, especially the ground-truth answers be detailed as per the context, although it does not affect the evaluation of the RAGs, given that LLMs can easily compare two answers to see if the generated response contains answer closer to the ground truth, even if the latter is too short. We can also infer that perhaps character text splitter is not good for mathematics, compared to the recursive method which gave it highest accuracy on RAG #57. One interesting aspect is that this high performing RAG for mathematics, had longer chunk size and overlap and used a smaller language model (Phi-3). This means that when provided with the context and prompt, the SLM is capable of finding the answer in the unstructured context returned by the retriever. Using a SLM can greatly speed up the adoption of a RAG-powered LLM in consumer hardware, making such systems affordable to students to run at no cost.

Looking at Figure B.2 showing the physics RAG with same configuration as the high performing mathematics RAG, we have the opposite of Figure B.1. The generated answers from the RAG are the longest in most of the questions, but yet the accuracy was not as bad as the mathematics RAG configuration on high performing physics RAG.

5.2 Conclusion

In this project, we ran many experiments aimed at understanding how different RAGs perform for two different subjects, physics and mathematics. We have seen that physics RAG with highest accuracy (RAG #2) of 64%, whereas for mathematics the highest accuracy of 66% was for RAG# 57. The results were not aimed at achieving any state-of-the-art but to understand how varying different components of the RAG system affects the generation accuracy. We learned that mathematics performs best with higher chunk sizes compared to physics, and small models like Phi-3 can handle a complex subject quite well. We hope these results provide a ground work for future work in designing chatbots that enhance learning in complex subjects.

5.3 Future work

The world of RAG is an ongoing research. After doing this project, as future work, we aim to continue to build on the results by evaluating a chatbot system built using open-source tools. There are prototyping tools such as Streamlit[1] and Chainlit[2] that enable the creation of user interfaces for chatbots, and integrate them with LangChain and Ollama that we used in this project. We will also investigate other parameters within the RAG that can contribute to better retrieval capability of the RAG-powered LLM.

5.4 Ethical considerations

In this project, no human subjects were involved. We conducted experiments in generating synthetic data from course material to evaluate different retrieval augmented systems. As such, there are no issues with privacy violation nor the use of personal information of anyone's subject. Care was taken to analyze the course material downloaded from MIT OpenCourseWare to check if any personal information was connected to it, and none was found. Thus, to the best of our knowledge, the project adheres to ethical research standards.

[1] `https://streamlit.io/`
[2] `https://chainlit.io/`

Bibliography

Abdin, Marah, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. (2024). "Phi-3 technical report: A highly capable language model locally on your phone". In: *arXiv preprint arXiv:2404.14219*.

Ahmed, Imtiaz, Ayon Roy, Mashrafi Kajol, Uzma Hasan, Partha Protim Datta, and Md Rokonuzzaman Reza (2023). "ChatGPT vs. Bard: a comparative study". In: *Authorea Preprints*.

AI@Meta (2024). "Llama 3 Model Card". In: URL: `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Alberti, Chris, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins (2019). "Synthetic QA corpora generation with roundtrip consistency". In: *arXiv preprint arXiv:1906.05416*.

Alexandra Farazouli Teresa Cerratto-Pargman, Klara Bolander-Laksov and Cormac McGrath (2024). "Hello GPT! Goodbye home examination? An exploratory study of AI chatbots impact on university teachers' assessment practices". In: *Assessment & Evaluation in Higher Education* 49.3, pp. 363–375. DOI: `10.1080/02602938.2023.2241676`. eprint: `https://doi.org/10.1080/02602938.2023.2241676`. URL: `https://doi.org/10.1080/02602938.2023.2241676`.

Alkhaled, Lama and Ng Yee Fei (2023). "Automated Invoice Processing System". In: *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, pp. 0188–0192.

Allen, James F (2003). "Natural language processing". In: *Encyclopedia of computer science*, pp. 1218–1222.

Almeida, Roberto G de and Gary Libben (2005). "Changing morphological structures: The effect of sentence context on the interpretation of structurally ambiguous English trimorphemic words". In: *Language and Cognitive Processes* 20.1-2, pp. 373–394.

Aydin, Ömer and Enis Karaarslan (2023). "Is ChatGPT Leading Generative AI? What is Beyond Expectations?" In: *Academic Platform Journal of Engineering and Smart Systems* 11.3, pp. 118–134.

Banko, Michele and Eric Brill (2001). "Scaling to very very large corpora for natural language disambiguation". In: *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pp. 26–33.

Biderman, Stella, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. (2023). "Pythia: A suite for analyzing large language models across training and scaling". In: *International Conference on Machine Learning*. PMLR, pp. 2397–2430.

Borji, Ali and Mehrdad Mohammadian (2023). "Battle of the Wordsmiths: Comparing ChatGPT, GPT-4, Claude, and Bard". In: *GPT-4, Claude, and Bard (June 12, 2023)*.

Chai, Hum Yan, Liang Kim Meng, Hamam Mohamed, Hon Hock Woon, and Khin Wee Lai (2016). "Elimination of character-resembling anomalies within a detected region using density-dependent reference point construction in an automated license plate recognition system". In: *Journal of Electronic Imaging* 25.6, pp. 061614–061614.

Chen, Zeming, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut (2023). *MEDITRON-70B: Scaling Medical Pretraining for Large Language Models*. arXiv: `2311.16079 [cs.CL]`.

Conover, Mike, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin (2023). *Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM*. URL: `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm` (visited on 06/30/2023).

Curran, James Richard (2004). "From distributional to semantic similarity". In: *Edinburgh Research Archive: Informatics thesis and dissertation collection*.

Devi, K Vimala, V Manjula, Tareek Pattewar, et al. (2023). *ChatGPT: Comprehensive Study On Generative AI Tool*. Academic Guru Publishing House.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: `1810.04805 [cs.CL]`.

Ding, Yujuan, Wenqi Fan, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li (2024). *A Survey on RAG Meets LLMs: Towards Retrieval-Augmented Large Language Models*. arXiv: `2405.06211 [cs.CL]`.

Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou (2024). *The Faiss library*. arXiv: 2401.08281 [cs.LG].

Es, Shahul, Jithin James, Luis Espinosa-Anke, and Steven Schockaert (2023). "Ragas: Automated evaluation of retrieval augmented generation". In: *arXiv preprint arXiv:2309.15217*.

Farah, Juan Carlos, Sandy Ingram, Basile Spaenlehauer, Fanny Kim-Lan Lasne, and Denis Gillet (2023). "Prompting Large Language Models to Power Educational Chatbots". In: *International Conference on Web-Based Learning*. Springer, pp. 169–188.

Faruqui, Syed Hasib Akhter, Nazia Tasnim, Iftekhar Ibne Basith, Suleiman Obeidat, and Faruk Yildiz (2024). *Integrating A.I. in Higher Education: Protocol for a Pilot Study with 'SAMCares: An Adaptive Learning Hub'*. arXiv: 2405.00330 [cs.CY].

Ferber, Dyke, Isabella C Wiest, Georg Wölflein, Matthias P Ebert, Gernot Beutel, Jan-Niklas Eckardt, Daniel Truhn, Christoph Springfeld, Dirk Jäger, and Jakob Nikolas Kather (2024). "GPT-4 for Information Retrieval and Comparison of Medical Oncology Guidelines". In: *NEJM AI*, AIcs2300235.

Fu, Xue-Yong, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Bhushan TN (2024). *Tiny Titans: Can Smaller Large Language Models Punch Above Their Weight in the Real World for Meeting Summarization?* arXiv: 2402.00841 [cs.CL].

Gale, William A, Kenneth Church, and David Yarowsky (1992). "One sense per discourse". In: *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Gao, Leo, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy (2020). "The Pile: An 800GB Dataset of Diverse Text for Language Modeling". In: *arXiv preprint arXiv:2101.00027*.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv: 2312.10997 [cs.CL].

Gašević, Dragan, George Siemens, and Shazia Sadiq (2023). *Empowering learners for the age of artificial intelligence*.

Greshake, Kai, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz (2023). "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection". In: *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pp. 79–90.

Gunes, Yasin Celal and Turay Cesur (2024). "A Comparative Study: Diagnostic Performance of ChatGPT 3.5, Google Bard, Microsoft Bing, and Radiologists in Thoracic Radiology Cases". In: *medRxiv*, pp. 2024–01.

Gwon, Yong Nam, Jae Heon Kim, Hyun Soo Chung, Eun Jee Jung, Joey Chun, Serin Lee, and Sung Ryul Shim (2024). "The Use of Generative AI for Scientific Literature Searches for Systematic Reviews: ChatGPT and Microsoft Bing AI Performance Evaluation". In: *JMIR Medical Informatics* 12, e51187.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hum, Yan Chai, Yee Kai Tee, Wun-She Yap, Hamam Mokayed, Tian Swee Tan, Maheza Irna Mohamad Salim, and Khin Wee Lai (2022). "A contrast enhancement framework under uncontrolled environments based on just noticeable difference". In: *Signal Processing: Image Communication* 103, p. 116657.

Ide, Nancy and Jean Véronis (1998). "Introduction to the special issue on word sense disambiguation: the state of the art". In: *Computational linguistics* 24.1, pp. 1–40.

Javed, Saleha, Fredrik Sandin, Hamam Mokayed, Jerker Delsing, and Marcus Liwicki (2022). "Deep Ontology Alignment with BERT_INT: Improvements and Industrial Internet of Things (IIoT) Case Study". In.

Javed, Saleha, Muhammad Usman, Fredrik Sandin, Marcus Liwicki, and Hamam Mokayed (2023a). "Deep Ontology Alignment Using a Natural Language Processing Approach for Automatic M2M Translation in IIoT". In: *Sensors* 23.20, p. 8427.

Javed, Salman, Aparajita Tripathy, Jan van Deventer, Hamam Mokayed, Cristina Paniagua, and Jerker Delsing (2023b). "An approach towards demand response optimization at the edge in smart energy systems using local clouds". In: *Smart Energy* 12, p. 100123.

Jerison, David (2006). *Single Variable Calculus*. MIT OpenCourseWare: Massachusetts Institute of Technology. 18.01 (Fall 2006). Licensed under CC BY-NC-SA 4.0. Available at `https://ocw.mit.edu/courses/18-01-single-variable-calculus-fall-2006/`.

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed (2023). *Mistral 7B*. arXiv: `2310.06825 [cs.CL]`.

Jiang, Albert Q, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. (2024). "Mixtral of experts". In: *arXiv preprint arXiv:2401.04088*.

Jiao, Xiaoqi, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu (2019). "Tinybert: Distilling bert for natural language understanding". In: *arXiv preprint arXiv:1909.10351*.

Jurafsky, Daniel and James H Martin (n.d.). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.

Khalid, Marzuki, Rubiyah Yusof, and Hamam Mokayed (2011). "Fusion of multi-classifiers for online signature verification using fuzzy logic inference". In: *International Journal of Innovative Computing* 7.5, pp. 2709–2726.

Klebahn, Perry and Sebastian Krakowski (2023). *How You Can Use ChatGPT to Increase Your Creative Output*. `https://online.stanford.edu/how-you-can-use-chatgpt-increase-your-creative-output`. Accessed: 2024-04-10.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. arXiv: `1909.11942 [cs.CL]`.

Latif, Ehsan, Luyang Fang, Ping Ma, and Xiaoming Zhai (2024). *Knowledge Distillation of LLM for Automatic Scoring of Science Education Assessments*. arXiv: `2312.15842 [cs.CL]`.

Leung[1], Rosanna and Iris Sheungting Lo (2024). "Check for Can ChatGPT Inspire Me? Evaluate Students' Questioning Techniques on AI Tool for Overcoming Fixation Rosanna Leung[1] and Iris Sheungting Lo[2]". In: *Information and Communication Technologies in Tourism 2024: ENTER 2024 International eTourism Conference, Izmir, Türkiye, January 17–19*. Springer Nature, p. 75.

Li, Xianming and Jing Li (2023). "AnglE-optimized Text Embeddings". In: *arXiv preprint arXiv:2309.12871*.

Li, Yuhua, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett (2006). "Sentence similarity based on semantic nets and corpus statistics". In: *IEEE transactions on knowledge and data engineering* 18.8, pp. 1138–1150.

Lieb, Anna and Toshali Goel (2024). "Student Interaction with NewtBot: An LLM-as-tutor Chatbot for Secondary Physics Education". In: *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*. CHI EA '24. ¡conf-loc¿ ¡city¿Honolulu¡/city¿ ¡state¿HI¡/state¿ ¡country¿USA¡/country¿ ¡/conf-loc¿: Association for Computing Machinery. ISBN: 9798400703317. DOI: `10.1145/3613905.3647957`. URL: `https://doi.org/10.1145/3613905.3647957`.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: `1907.11692 [cs.CL]`.

Luo, Ziyang, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang (2023). "Wizardcoder: Empowering code large language models with evol-instruct". In: *arXiv preprint arXiv:2306.08568*.

Maryamah, Maryamah, Muhammad Maula Irfani, Edric Boby Tri Raharjo, Netri Alia Rahmi, Mohammad Ghani, and Indra Kharisma Raharjana (2024). "Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access". In: *2024 16th International Conference on Knowledge and Smart Technology (KST)*, pp. 259–264. DOI: 10.1109/KST61284.2024.10499652.

Meta Platforms, Inc. (2023). *Llama 2 License Agreement*. https://github.com/meta-llama/llama/blob/main/LICENSE. Version Release Date: July 18, 2023. Ireland and USA: Meta Platforms.

Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao (2024). "Large language models: A survey". In: *arXiv preprint arXiv:2402.06196*.

Mokayed, Hamam, Liang Kim Meng, Hon Hock Woon, and Ng Hooi Sin (2014). "Car plate detection engine based on conventional edge detection technique". In: *The International Conference on Computer Graphics, Multimedia and Image Processing (CGMIP2014). The Society of Digital Information and Wireless Communication*.

Mokayed, Hamam, Amirhossein Nayebiastaneh, Lama Alkhaled, Stergios Sozos, Olle Hagner, and Björn Backe (2024). "Challenging YOLO and Faster RCNN in Snowy Conditions: UAV Nordic Vehicle Dataset (NVD) as an Example". In: *2024 2nd International Conference on Unmanned Vehicle Systems-Oman (UVS)*. IEEE, pp. 1–6.

Mokayed, Hamam, Amirhossein Nayebiastaneh, Kanjar De, Stergios Sozos, Olle Hagner, and Björn Backe (2023). "Nordic Vehicle Dataset (NVD): Performance of vehicle detectors using newly captured NVD from UAV in different snowy weather conditions." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5313–5321.

Mokayed, Hamam, Shivakumara Palaiahnakote, Lama Alkhaled, and Ahmed N AL-Masri (2022). "License plate number detection in drone images". In: *Artificial Intelligence and Applications*.

Mokayed, Hamam, Palaiahnakote Shivakumara, Marcus Liwicki, and Umapada Pal (2020). "A new defect detection method for improving text detection and Recognition performances in natural scene images". In: *2020 Swedish Workshop on Data Science (SweDS)*. IEEE, pp. 1–7.

Mokayed, Hamam, Palaiahnakote Shivakumara, Rajkumar Saini, Marcus Liwicki, Loo Chee Hin, and Umapada Pal (2021). "Anomaly detection in natural scene images based on enhanced fine-grained saliency and fuzzy logic". In: *IEEE Access* 9, pp. 129102–129109.

Nasukawa, Tetsuya and Jeonghee Yi (2003). "Sentiment analysis: Capturing favorability using natural language processing". In: *Proceedings of the 2nd international conference on Knowledge capture*, pp. 70–77.

Navigli, Roberto (2009). "Word sense disambiguation: A survey". In: *ACM computing surveys (CSUR)* 41.2, pp. 1–69.

Nechakhin, Vladyslav, Jennifer D'Souza, and Steffen Eger (2024). *Evaluating Large Language Models for Structured Science Summarization in the Open Research Knowledge Graph*. arXiv: 2405.02105 [cs.AI].

Neupane, Subash, Elias Hossain, Jason Keith, Himanshu Tripathi, Farbod Ghiasi, Noorbakhsh Amiri Golilarz, Amin Amirlatifi, Sudip Mittal, and Shahram Rahimi (2024). *From Questions to Insightful Answers: Building an Informed Chatbot for University Resources*. arXiv: 2405.08120 [cs.ET].

Nikolaidou, Konstantina, George Retsinas, Vincent Christlein, Mathias Seuret, Giorgos Sfikas, Elisa Barney Smith, Hamam Mokayed, and Marcus Liwicki (2023). "Wordstylist: styled verbatim handwritten text generation with latent diffusion models". In: *International Conference on Document Analysis and Recognition*. Springer Nature Switzerland Cham, pp. 384–401.

OpenAI et al. (2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL].

Peng, Wenjun, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen (2024). "Large language model based long-tail query rewriting in taobao search". In: *Companion Proceedings of the ACM on Web Conference 2024*, pp. 20–28.

Pickering, Martin J and Roger PG Van Gompel (2006). "Syntactic parsing". In: *Handbook of psycholinguistics*. Elsevier, pp. 455–503.

Puri, Raul, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro (2020). "Training question answering models from synthetic data". In: *arXiv preprint arXiv:2002.09599*.

Radev, Dragomir, Eduard Hovy, and Kathleen McKeown (2002). "Introduction to the special issue on summarization". In: *Computational linguistics* 28.4, pp. 399–408.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training". In.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang (2016). "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250*.

Ram, Bal and Pratima Verma (2023). "Artificial intelligence AI-based Chatbot study of ChatGPT, Google AI Bard and Baidu AI". In: *World Journal of Advanced Engineering Technology and Sciences* 8.01, pp. 258–261.

Resnik, Philip (1999). "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language". In: *Journal of artificial intelligence research* 11, pp. 95–130.

Riabi, Arij, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano (2020). "Synthetic data augmentation for zero-shot cross-lingual question answering". In: *arXiv preprint arXiv:2010.12643*.

Rosenfeld, Ronald (2000). "Incorporating linguistic structure into statistical language models". In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 358.1769, pp. 1311–1324.

Roucher, Aymeric (n.d.). *RAG Evaluation*. `https://huggingface.co/learn/cookbook/en/rag_evaluation`. Accessed: 2024-04-04.

Roumeliotis, Konstantinos I, Nikolaos D Tselikas, and Dimitrios K Nasiopoulos (2023). "Llama 2: Early Adopters' Utilization of Meta's New Open-Source Pretrained Model". In.

Roy, Ayush, Palaiahnakote Shivakumara, Umapada Pal, Hamam Mokayed, and Marcus Liwicki (2023). "Fourier feature-based CBAM and vision transformer for text detection in drone images". In: *International Conference on Document Analysis and Recognition*. Springer Nature Switzerland Cham, pp. 257–271.

Salemi, Alireza and Hamed Zamani (2024). "Evaluating Retrieval Quality in Retrieval-Augmented Generation". In: *arXiv preprint arXiv:2404.13781*.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108*.

Sean Lee Aamir Shakir, Darius Koenig-Julius Lipp (2024). *Open Source Strikes Bread - New Fluffy Embeddings Model*. URL: `https://www.mixedbread.ai/blog/mxbai-embed-large-v1`.

Shakeri, Siamak, Cicero dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang (2020). "End-to-end synthetic data generation for domain adaptation of question answering systems". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5445–5460.

Shoufan, Abdulhadi (2023). "Exploring Students' Perceptions of ChatGPT: Thematic Analysis and Follow-Up Survey". In: *IEEE Access* 11, pp. 38805–38818. DOI: `10.1109/ACCESS.2023.3268224`.

Srishti, Richa (2024). "ChatGPT in Education: Augmenting Learning Experience or Dehumanizing Education?" In: *Educational Perspectives on Digital Technologies in Modeling and Management*. IGI Global, pp. 114–128.

Sutskever, Ilya (2013). *Training recurrent neural networks*. University of Toronto Toronto, ON, Canada.

Sutskever, Ilya, James Martens, and Geoffrey E Hinton (2011). "Generating text with recurrent neural networks". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 1017–1024.

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto (2023). "Alpaca: A strong, replicable instruction-following model". In: *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html* 3.6, p. 7.

Team, Gemma, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. (2024). "Gemma: Open models based on gemini research and technology". In: *arXiv preprint arXiv:2403.08295*.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. (2023). "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971*.

Vacalopoulou, Anna, Viktor Gardelli, Theodoris Karafyllidis, Foteini Liwicki, Hamam Mokayed, Marios Papaevripidou, George Paraskevopoulos, Spyridoula Stamouli, Athanasios Katsamanis, and Vassilis Katsouros (2024). "AI4EDU: An Innovative Conversational Ai Assistant For Teaching And Learning". In: *INTED2024 Proceedings*. IATED, pp. 7119–7127.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Wang, Jiayi, David Ifeoluwa Adelani, Sweta Agrawal, Ricardo Rei, Eleftheria Briakou, Marine Carpuat, Marek Masiak, Xuanli He, Sofia Bourhim, Andiswa Bukula, et al. (2023). "AfriMTE and AfriCOMET: Empowering COMET to Embrace Under-resourced African Languages". In: *arXiv preprint arXiv:2311.09828*.

Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi (2022). "Self-instruct: Aligning language models with self-generated instructions". In: *arXiv preprint arXiv:2212.10560*.

Williams, Mobolaji (2017). *Introduction to Oscillations and Waves*. MIT OpenCourseWare: Massachusetts Institute of Technology. RES.8-009 (Summer 2017). Licensed under CC BY-NC-SA 4.0. Available at `https://ocw.mit.edu/courses/res-8-009-introduction-to-oscillations-and-waves-summer-2017/`.

Xiao, Changrong, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia (July 2023). "Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase

of ChatGPT in Education Applications". In: *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. Ed. by Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anaïs Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch. Toronto, Canada: Association for Computational Linguistics, pp. 610–625. DOI: 10.18653/v1/2023.bea-1.52. URL: https://aclanthology.org/2023.bea-1.52.

Yen, An-Zi and Wei-Ling Hsu (Dec. 2023). "Three Questions Concerning the Use of Large Language Models to Facilitate Mathematics Learning". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 3055–3069. DOI: 10.18653/v1/2023.findings-emnlp.201. URL: https://aclanthology.org/2023.findings-emnlp.201.

Yu, Hao (2023). "Reflection on whether Chat GPT should be banned by academia from the perspective of education and teaching". In: *Frontiers in Psychology* 14, p. 1181712.

Zhang, Peiyuan, Guangtao Zeng, Tianduo Wang, and Wei Lu (2024). "Tinyllama: An open-source small language model". In: *arXiv preprint arXiv:2401.02385*.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. (2024). "Judging llm-as-a-judge with mt-bench and chatbot arena". In: *Advances in Neural Information Processing Systems* 36.

Zimmermann, Lucien and Florian Rohrer (2024). *Study Buddy*. OST-Ostschweizer Fachhochschule. URL: https://eprints.ost.ch/id/eprint/1176/.

Appendices

A Appendices

A.1 RAG Scenarios

The table A.1 shows the full list of RA scenarios used in the experimentation. There are a total of 64 RAG combinations.

Scenario	Chunk Size	Overlap	Text Splitter	Vector Store	Embedding Model	Model
1	500	50	Character	Chroma	mxbai-embed-large	phi3
2	500	50	Character	Chroma	mxbai-embed-large	llama3
3	500	50	Character	Chroma	llama3	phi3
4	500	50	Character	Chroma	llama3	llama3
5	500	50	Character	FAISS	mxbai-embed-large	phi3
6	500	50	Character	FAISS	mxbai-embed-large	llama3
7	500	50	Character	FAISS	llama3	phi3
8	500	50	Character	FAISS	llama3	llama3
9	500	50	RecursiveCharacter	Chroma	mxbai-embed-large	phi3
10	500	50	RecursiveCharacter	Chroma	mxbai-embed-large	llama3
11	500	50	RecursiveCharacter	Chroma	llama3	phi3
12	500	50	RecursiveCharacter	Chroma	llama3	llama3
13	500	50	RecursiveCharacter	FAISS	mxbai-embed-large	phi3
14	500	50	RecursiveCharacter	FAISS	mxbai-embed-large	llama3
15	500	50	RecursiveCharacter	FAISS	llama3	phi3
16	500	50	RecursiveCharacter	FAISS	llama3	llama3
17	500	100	Character	Chroma	mxbai-embed-large	phi3
18	500	100	Character	Chroma	mxbai-embed-large	llama3
19	500	100	Character	Chroma	llama3	phi3
20	500	100	Character	Chroma	llama3	llama3
21	500	100	Character	FAISS	mxbai-embed-large	phi3
22	500	100	Character	FAISS	mxbai-embed-large	llama3
23	500	100	Character	FAISS	llama3	phi3
24	500	100	Character	FAISS	llama3	llama3

25	500	100	RecursiveCharacter	Chroma	mxbai-embed-large	phi3
26	500	100	RecursiveCharacter	Chroma	mxbai-embed-large	llama3
27	500	100	RecursiveCharacter	Chroma	llama3	phi3
28	500	100	RecursiveCharacter	Chroma	llama3	llama3
29	500	100	RecursiveCharacter	FAISS	mxbai-embed-large	phi3
30	500	100	RecursiveCharacter	FAISS	mxbai-embed-large	llama3
31	500	100	RecursiveCharacter	FAISS	llama3	phi3
32	500	100	RecursiveCharacter	FAISS	llama3	llama3
33	1000	50	Character	Chroma	mxbai-embed-large	phi3
34	1000	50	Character	Chroma	mxbai-embed-large	llama3
35	1000	50	Character	Chroma	llama3	phi3
36	1000	50	Character	Chroma	llama3	llama3
37	1000	50	Character	FAISS	mxbai-embed-large	phi3
38	1000	50	Character	FAISS	mxbai-embed-large	llama3
39	1000	50	Character	FAISS	llama3	phi3
40	1000	50	Character	FAISS	llama3	llama3
41	1000	50	RecursiveCharacter	Chroma	mxbai-embed-large	phi3
42	1000	50	RecursiveCharacter	Chroma	mxbai-embed-large	llama3
43	1000	50	RecursiveCharacter	Chroma	llama3	phi3
44	1000	50	RecursiveCharacter	Chroma	llama3	llama3
45	1000	50	RecursiveCharacter	FAISS	mxbai-embed-large	phi3
46	1000	50	RecursiveCharacter	FAISS	mxbai-embed-large	llama3
47	1000	50	RecursiveCharacter	FAISS	llama3	phi3
48	1000	50	RecursiveCharacter	FAISS	llama3	llama3
49	1000	100	Character	Chroma	mxbai-embed-large	phi3
50	1000	100	Character	Chroma	mxbai-embed-large	llama3
51	1000	100	Character	Chroma	llama3	phi3
52	1000	100	Character	Chroma	llama3	llama3
53	1000	100	Character	FAISS	mxbai-embed-large	phi3
54	1000	100	Character	FAISS	mxbai-embed-large	llama3
55	1000	100	Character	FAISS	llama3	phi3
56	1000	100	Character	FAISS	llama3	llama3
57	1000	100	RecursiveCharacter	Chroma	mxbai-embed-large	phi3
58	1000	100	RecursiveCharacter	Chroma	mxbai-embed-large	llama3
59	1000	100	RecursiveCharacter	Chroma	llama3	phi3
60	1000	100	RecursiveCharacter	Chroma	llama3	llama3
61	1000	100	RecursiveCharacter	FAISS	mxbai-embed-large	phi3
62	1000	100	RecursiveCharacter	FAISS	mxbai-embed-large	llama3
63	1000	100	RecursiveCharacter	FAISS	llama3	phi3
64	1000	100	RecursiveCharacter	FAISS	llama3	llama3

Table A.1: Retrieval-augmented generation (RAG) scenarios used in the experimentation.

A.2 Short ground-truth answers in Maths

The table A.2 shows the top 20 synthetic question and ground-truth pairs and the answer generated by maths RAG having configuration 2 (the configuration that Physics

had highest accuracy).

No.	Question	Answer	Ground Truth
1	What does sin θ approach as θ approaches 0?	I don't know.	1
2	What is the surface area of a unit sphere?	I don't know.	4π
3	What is approximately equal to L?	The critical point.	h
4	What is the value of $f(0)$?	0	0
5	What is the value of A when $x = 0$?	I don't know.	3
6	What is the value of h when $r = 2$?	I don't know.	5
7	What is the maximum area that can be enclosed by two squares with sides of length x and $1 - x$?	$2x + 2(1 - x)$	4
8	What is the value of C_1 in the equation $\frac{1}{2}B_1x + C_1 = (x^2 + 1)(x - 1)$?	I don't know.	-2
9	What is the maximum power of x in a quadratic factor like $(Ax^2 + Bx + C)$?	2	2
10	What does sin θ approach as θ approaches 0?	I don't know.	1
11	What is the area of the triangle formed by the tangent line and the x- and y-axes?	2	2
12	What is the value of x that Newton's method converges to?	I don't know.	$\sqrt{3}$
13	What is the maximum area that can be enclosed by two squares with sides of length x and $1 - x$?	$2x + 2(1 - x)$	4
14	What is the linear approximation of $\sin x$?	I don't know.	x
15	What is the value of the limit as x approaches $15 - 1$?	I don't know.	5
16	What is the value of $f(0)$?	0	0
17	What is the surface area of a unit sphere?	I don't know.	4π
18	What is the value of the limit as x approaches $15 - 1$?	I don't know.	5
19	What is the value of $\frac{d(\sin(x))}{dx}$ when $x = 0$?	I don't know.	1
20	What is the base point used to expand the Taylor series for $\sqrt{x}$?	The base point used to expand the Taylor series for $\sqrt{x}$ is $b = 1$.	1

Table A.2: Questions, Answers, and Ground Truths

B Extra figures

B.1 Maths RAG on high performing physics RAG

The figure B.1 shows the character count for the RAG configuration #2 for mathematics that physics had the highest accuracy.

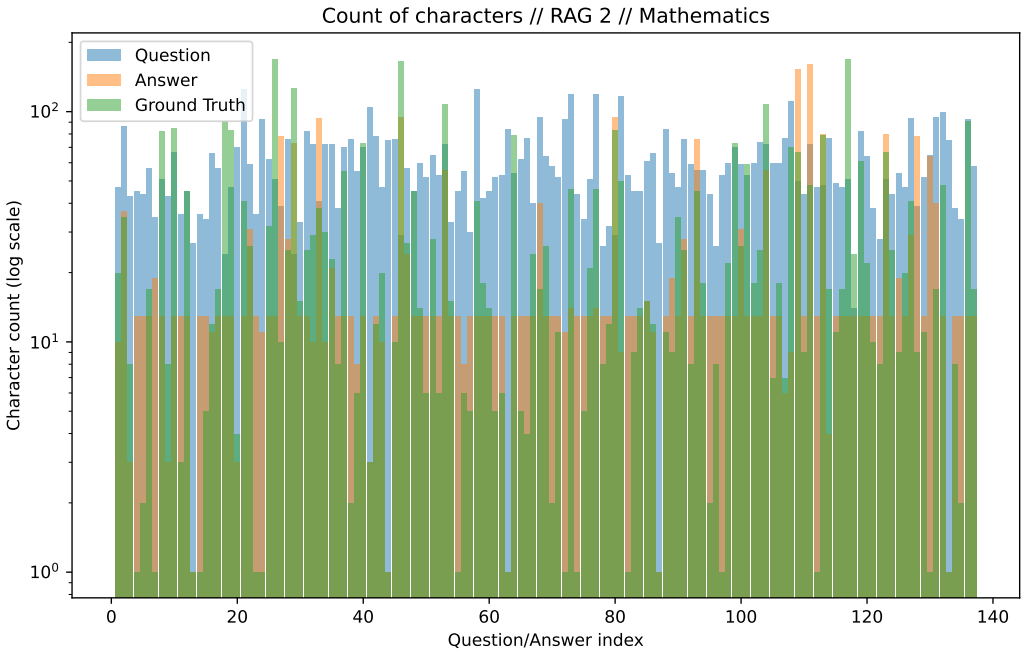

Figure B.1: Count of characters (log-scale) of question, answer (RAG) and ground-truth answer for Mathematics RAG (#2).

B.2 Physics RAG on high performing maths RAG

The figure B.2 shows the character count for the RAG configuration #57 for physics that maths had the highest accuracy.

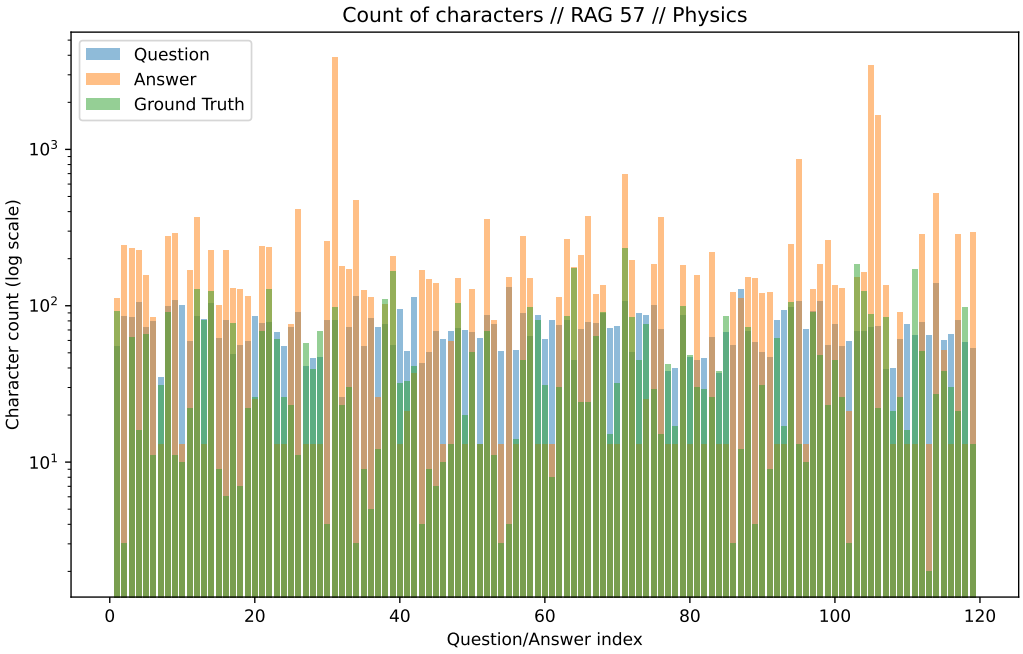

Figure B.2: Count of characters (log-scale) of question, answer (RAG) and ground-truth answer for Physics RAG (#57).