EPFL MISTRAL MIND CHATBOT

Mohamed CHARFI | 311171 | mohamed.charfi@epfl.ch
Yacine CHAOUCH | 300418 | yassine.chaouch@epfl.ch
Aziz LAADHAR | 315196 | aziz.laadhar@epfl.ch
LASMER

Abstract

This paper introduces the EPFL Chatbot, a specialized language model designed to support the academic needs of students at École Polytechnique Fédérale de Lausanne (EPFL). Utilizing fine-tuning and Retrieval-Augmented Generation (RAG) techniques, the model aligns closely with the curriculum, enabling effective interaction in educational contexts. Our primary focus was on enhancing the model's capabilities in multiple-choice question answering and context-aware responses by integrating data from EPFL-specific resources and advanced prompt engineering. Experimental evaluations revealed that while the base model achieved notable performance, the incorporation of DPO (Direct Preference Optimization) and RAG provided modest improvements. These results highlight the potential of tailored LLMs in educational settings, while emphasizing the importance of domain-specific adaptation in enhancing the learning experience.

1 Introduction

With the recent rise of large language models (LLMs), there has been significant progress in natural language processing and understanding. These models showed impressive capabilities across various domains, showcasing an ability to generate coherent and contextually relevant text, translate languages, summarize content, and answer questions. However, despite their broad utility, these general-purpose LLMs do not guarantee their impressive performance with respect to domain-specific expertise, limiting their effectiveness and reliability in fields such as healthcare, finance, law, and education. Recognizing this limitation, researchers have developed specialized training methodologies and techniques to adapt language models for specific use cases.

Building on this, we present the EPFL Chatbot, a specialized language model tailored to the curriculum of École Polytechnique Fédérale de Lausanne (EPFL). Our model uses recent advancement of fine-tuning and Retrieval-Augmented Generation (RAG) (Gao et al., 2023), to leverage predominantly relevant data sources, ensuring it aligns closely with the academic requirements. This approach represents a novel experiment in the application of language models in education, offering a customized tool that addresses the specific needs of EPFL students.

The rest of the paper is organized as follows. Section 2 reviews existing literature on specialized language models and their applications in various domains, highlighting the gap our model aims to fill. Section 3 describes our methodology for data collection, training, and specialization of the EPFL Chatbot. Section 4 outlines the experiments conducted to evaluate the performance of our model, including the datasets used and the metrics for assessment. Section 5 presents a detailed analysis of our model's performance. Section 6 reflects on the broader impact of our work and any ethical considerations associated with the development and deployment of the EPFL Chatbot. Finally, Section 7 summarizes our findings, highlights the achievements of our project, and suggests avenues for future research.

2 Related Work

While general-purpose language models like ChatGPT (Brown et al., 2020) , Llama (Touvron et al., 2023), and Mistral (Jiang et al., 2023) exhibit impressive capabilities, they often lack the subject-matter expertise required for specialized contexts. To bridge this gap, researchers have developed domain-specific models such as ChatDoctor (Li et al., 2023) for healthcare, HuaTuoGPT (Zhang et al., 2023) for medical consultations, FinGPT (Yang et al., 2023) for financial analysis, and Chat-

Law (Cui et al., 2024) for legal assistance. Similarly, educational AI models like COURSEGPT-ZH (Qu et al., 2024), are trained on educational datasets to enhance their applicability in academic settings .

Our AI chatbot tutor, tailored for the EPFL Computer Science (IC) curriculum, integrates several advanced techniques to address the specific needs of computer science education. Direct Preference Optimization (DPO) (Rafailov et al., 2023) was employed to ensure that the Chatbot's responses maintain a coherent chain of thought and adhere to ethical reasoning. We then experimented with Supervised Fine-Tuning (SFT) (Alijani and Rahtu, 2021) to specialize the model's capabilities in handling multiple-choice questions (MCQs) format, optimizing it to generate concise, correct answers aligned with educational assessments. To enhance context-awareness, we incorporated Retrieval-Augmented Generation (RAG), which combines generative capabilities with dynamic information retrieval from a curated dataset, thus enriching the chatbot's responses with contextually relevant information.

The application of AI tutors in educational settings addresses critical challenges such as personalized learning, immediate feedback, and scalable instruction, providing substantial benefits over traditional methods. For instance, AI Tutor (Cao, 2023) has demonstrated its potential in addressing educational disparities among disadvantaged and under-resourced children in China by providing accessible, personalized support. NewtBot (Lieb and Goel, 2024), designed for secondary physics education, showcases how AI can enhance student engagement and learning outcomes through personalized tutoring. Furthermore, our use case involves individuals with a foundational knowledge of AI, which contrasts with findings by (Zamfirescu-Pereira et al., 2023) regarding non-AI experts. Yao et al. highlight the complexities that non-AI experts face in prompt engineering, revealing that they often struggle with effective AI interaction design due to a lack of systematic approach and overgeneralization from human-to-human instructional experiences. Our model, designed for students proficient in AI, benefits from their ability to engage more effectively with the AI tutor, leveraging their background to refine interactions and enhance learning outcomes. This distinction highlights the importance of aligning AI educational tools with the user's technical proficiency to maximize their effectiveness and user satisfaction.

3 Approach

Our approach involved structured steps in data collection, training, and specialization, tailored to meet diverse educational needs. We employed Direct Preference Optimization (DPO) and curated a variety of datasets to enhance the model's capabilities in different dimensions. For generating ethically sound responses, we utilized the Anthropic HH RLHF dataset, which provided prompts and responses based on Helpfulness-Honesty-Harmlessness (HHH) (Huang et al., 2023) criteria. This dataset was instrumental in training the model to produce responsible, and ethically appropriate outputs by learning from interactions that emphasized ethical considerations and usefulness. It also ensured that the sole purpose of this Chatbot was providing help as a tutor and not reply to any other potential inappropriate request.

To improve error detection and correction, we incorporated the MMLU-STEM (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021) datasets. The MMLU-STEM dataset focused on breadth, providing a wide range of STEM-related questions that required correct, helpful responses. This means that given a correct answer, we generate, using a GPT wrapper, another response following a wide range of criterias including but not limited to 'not relevant','misleading','too general' or even 'containing hate content'. In contrast, GSM8K was used to generate deliberately incorrect answers through methods like deletion, substitution, and insertion. The purpose was to guarantee that the model does not do careless mistakes that may go unnoticed by students. This dual approach allowed the model to learn to distinguish correct answers from incorrect ones, enhancing its future ability to identify and correct errors effectively.

We also used the EPFL Preference Dataset, which was curated by EPFL students. This dataset contained course-specific questions and responses, tailored by the students themselves. Although this dataset in general did not improve our model reward, we estimated that it was essential for the purpose of reflecting the actual academic expectations and preferences of EPFL students when it comes to good/bad answers. It was crucial not only for aligning the model's responses with real-world academic standards but also to meet the specific needs of the EPFL student body in particular. For that, we curated only questions carefully made by students. Our criteria for filtering was that more responsible students took put extra effort to provide more optional criteria in comparing the chosen/rejectedf pairs.

Additionally, we included the SetFit (Ultron, 2020) dataset to improve the model's general knowledge and response quality. This dataset provided examples across various topics, following the HHH criteria, thus broadening the model's ability to generate clear, contextually appropriate answers in diverse scenarios. The integration of these datasets created a comprehensive training set that balanced breadth, depth, and real-world relevance, ensuring the model could generate helpful and responsible answers.

For model training, we initially applied DPO to the Tiny Llama model but later transitioned to the Mistral 7B model (Jiang et al., 2023) for better performance. To further enhance the model's capabilities, we integrated a Retrieval-Augmented Generation (RAG) system by collecting textbooks and course materials from EPFL, creating embeddings, and employing a similarity search mechanism. This allowed the model to dynamically fetch relevant information from sources provided by the professors themselves, improving context richness and relevance of responses.

For multiple-choice question (MCQ) specialization, we designed a specific dataset and applied Supervised Fine-Tuning (SFT) optimized by LoRA (Hayou et al., 2024) adapters and quantization. We also crafted very carefully prompts that guided the model to adapt its answer format. This setup enabled the model to deliver precise single-letter answers, ensuring accurate and efficient performance tailored to the needs of STEM students at EPFL.

4 Experiments

4.1 Training the model to output a single letter

After training the DPO model, the goal was to improve the model's accuracy the MCQ style. To enhance the model's accuracy in answering MCQs, we prepared a dataset of 6,000 samples derived primarily from the SciQ dataset (Johannes Welbl, 2017). This dataset included questions with three distractors and one correct answer, formatted with alphabet-letter options to facilitate single-letter responses. We ensured the correct answer was randomized among the options to prevent the model from learning positional biases and overfitting to specific patterns. In other words, we randomized the placement of correct answers across all options ('A', 'B', etc.) to ensure an even distribution.

Starting with our 7B DPO model, we applied quantization to the pre-trained model. This process reduced the model's memory footprint and computational requirements, making any future potential deployment more cost-effective and efficient. This has significantly decreased the memory usage and sped up the inference process. This step was crucial to balance performance with resource constraints, not only for training but especially for educational applications where affordability and accessibility are important.

Following quantization, we conducted Supervised Fine-Tuning (SFT) using the PEFT LoRA model. The objective was to refine the answer format without altering the model's core knowledge. Low-Rank Adaptation enables efficient fine-tuning by injecting trainable rank decomposition matrices into the Transformer layers, thereby reducing the number of parameters that need adjustment. The key challenge was setting the hyper-parameters for LoRA—specifically the rank r, alpha, and dropout rate. These parameters influence the model's ability to adapt to new tasks without excessive training or loss of existing knowledge.

Our initial attempt used r=256, with alpha set to half that value and a dropout rate of 0.05 over one epoch. In LoRA, r represents the rank of the decomposition, controlling the dimension of the subspace where adaptation occurs. A high rank like 256 allows more complex adaptation but can lead to over fitting if the adaptation capacity exceeds the needs of the task. Alpha scales the adaptation, ensuring it's proportionate to the pre-trained weights, and a lower value mitigates drastic changes, maintaining model stability. Dropout introduces randomness during training to prevent over fitting by deactivating a portion of neurons. With these settings, the model's performance degraded, likely due to over fitting on the formatting task and insufficient adaptation focus.

To address this, we doubled the dropout rate to 0.1 to introduce more regularization and prevent over fitting, while decreasing the rank to minimize new learning and focus on formatting adjustments. We fixed alpha at 16 to keep the weight changes proportionate and experimented with different values for r. Good results were observed with r=128 over two epochs, achieving a weight scaling ratio of 16/128, meaning approximately 12% of the new weights were used for the adjustment. This setup balanced adaptation and stability, allowing the model to effectively learn the formatting without significant knowledge alteration. Further improvement was noted with r=32 (2 * alpha) over one epoch, indicating that a lower rank, close to the alpha value, along with less exposure to training dataset (only one epoch), could provide more focused and efficient adaptation.

These findings highlight the need for careful hyper-parameter tuning when using SFT for specific tasks like answer formatting. The initial suboptimal results suggest that our dataset might have been too extensive or not ideally suited for purely formatting purposes. This inadequacy was evident as the model occasionally produced the same answer regardless of the prompt. In this context, to ensure the model retained its understanding capabilities and did not merely output letters randomly, we tested it with tricky prompts designed to request a specific letter, such as asking for answer "B" when it was positioned in option "A" and rotating to see the model's adjustments . This approach validated that the model could still correctly interpret prompts and generate appropriate responses, rather than just guessing or following learned patterns. Another point worth establishing is that the model did not learn to only output one followed by the end of sequence token but rather the option letter repeated (e.g: Answer= CCCCC..). This did not constitute an issue as it suffices to set maximum number of tokens generated to 1. However, we noticed with some of the earlier models that the model is not always categorical, meaning it can output 'CCCACBACCC'. After careful examination, we found that the next letters did not have a clear pattern. In other words, the letters order did not showcase the models ordering of the answers. Simple tests like rotating the answers displacements sufficed to rejected this hypothesis. However, it might be beneficial in the future to experiment other optimisations that leads almost surely to the format '[Correct Answer]<EOS>'.

4.2 Prompt Engineering

Although we managed to fine-tune the model's behaviour to output the requested MCQ answer format, we estimated that our results were not conclusive and deicded to further attempt to guide the LLM into the desired output by Prompt Engineering. To do so we did the following : Initially, we provide the model with the question and its options and allow it to generate a response in any format. We then collected the initial response and re-inputted the question among with its choices and the model's initial response. Then we asked the model to generate the response in a specific format from which we could easily extract the chosen letter. here is an example to demonstrate our approach:

first prompt

"You are an AI tutor specialized in Multiple-Choice Question Answering (MCQA). Make sure to give the letter choice.
"Question: Statement 1| Binary trees are used to implement binary search trees. Statement 2| A complete binary tree is also a full binary tree.
Options:
A. True, True
B. False, False
C. True, False
D. False, True
Answer:"

Intial response

"C. True, False
Explanation: Statement 1 is true because binary trees can be used to implement binary search trees. Statement 2 is false because a complete binary tree is not necessarily a full binary tree. A complete binary tree is one in which all levels except possibly the last are completely filled, and the last level is filled left to right. A full binary tree is one in which each internal node has exactly two children. "

Second Prompt

"You will be having a question and a response. I want you to extract the MCQ choice letter. Respond with the following format: [The chosen letter is _] where _ is the correct option (A,B,C,D). Don't focus on the question itself and just extract the answer. Please respect the format that I provided.

Question: [Previous Question]
Intial response : [First model response]"

Final response

The chosen letter is [C]

Although this approach yielded acceptable decent results, it's important to note it requires more than twice the computational resources, increasing significantly financial expenses and processing delays.

4.3 Adding RAG to the model

To add RAG to the model, we started by collecting the dataset. We decided to use the EPFL books used in the courses of the IC curriculum as our primary source of data. The data collection part involved manually downloading all the available books from the IC drive. For courses that are taught in French, we decided to find equivalent books in English to ensure that all the courses are covered.

Once we gathered all the books, we meticulously removed their starting pages. This approach was used to delete unnecessary content such as the book covers, outlines, and prefaces which are not useful dor our RAG implementation. In addition to that, we downloaded some student summaries and notes that we found on the drive. We believed that these additional data points might be interesting since these student contributions might give more detailed insights in the course material, which could be beneficial for our final model.

After completing the data collection phase, we moved on to implementing the retriever. We first began by extracting all the text from each PDF file. Given the big amount of text, the limited context length of our initial model (8192 tokens) and the limited length of our embedding model (384 words), we decided to break the text into manageable chunks. We created chunks of 200 words each, with a 50-word overlap between consecutive chunks as suggested in the Advanced RAG Output Grading paper (Eibich et al., 2024). This overlap helps maintain context between chunks.

These text chunks were then fed into the embedding model "all-mpnet-base-v2" (Endait et al., 2024) from sentence-transformers. This model effectively captures the semantic essence of the text and maps each chunk to a 768-dimensional dense vector space. We stored these embedding inside the model so that they are ready for retrieval during the next steps.

To implement the RAG functionality inside the response generation of the model, we did the following: For every prompt received, we embed the prompt using the same embedding model to ensure consistency and then we conduct a semantic search across our stored embedding to identify the chunks with the highest similarity to the prompt. These chunks are then concatenated with the prompt to give context. Here is an example on how our retriever works with only the top 2 chunks:

Prompt

How to delete an element from a linked list?

chunk 1 with Similarity: 0.7841

The “first” element. Deleting a Node from a Linked List After adding a node to a linked list, deleting a node from a list is the next most important operation.To delete a node from a linked list, use list_del(): list_del(struct list_head *entry) This function removes the element entry from the list. Note that it does not free any memory
...
To delete a node from a linked list and reinitialize it, the kernel provides list_del_init(): list_del_init(struct list_head *entry) This function

chunk 2 with Similarity: 0.6422

_list_del(struct list_head *prev, struct list_head *next) next->prev = prev; prev->next = next; static inline void list_del(struct list_head *entry)
...
To check whether a list is empty list_empty(struct list_head *head) This returns nonzero if the given list is empty; otherwise, it returns zero. To splice two unconnected lists together list_splice(struct list_head *list, struct list_head *head) This function splices together two lists

5 Analysis

5.1 Evaluation Dataset

Since the main goal of our model is to support EPFL students with their homework, the best dataset to evaluate it are the previous exams that we had access to using the EPFL-IC drive. First we manually picked 2 solved exams from each course, we then used Chat-GPT 4-o to generate 10 answered MCQAs questions per course. To check the correctness of the generated questions we manually went through each question and kept only the ones that we were able to verify their answers. We than chose the most pertinent and important 300 questions as our evaluation dataset.

5.2 Evaluation Results

The evaluation of our model was conducted using the selected dataset, consisting of 300 questions curated from previous exams. The performance of different variations of our model was measured in terms of accuracy. The results are summarized in Table 1.

Model	Accuracy
MISTRAL	55.1%
MISTRAL + DPO	54.1%
MISTRAL + RAG	**55.5**%
MISTRAL + DPO + RAG	55.45%

Table 1: Evaluation Results of the Model on the Evaluation Dataset

These results indicate that while the base model performed well on the dataset, the additions of DPO and RAG did not significantly enhance the performance on the evaluation dataset.

6 Ethical considerations

The development of our chatbot involves several ethical considerations, including data privacy, bias and fairness, transparency, and the impact on student learning and well-being. We ensure data privacy by restricting access to the chatbot exclusively to EPFL students, as the data retrieved from the EPFL-IC drive is not publicly accessible. This measure safeguards sensitive information and ensures that only authorized users can interact with the chatbot.

Ensuring the ethical and responsible use of the EPFL Chatbot is also crucial. The chatbot is designed to assist students with their academic work and provide educational support, rather than to replace human instructors or facilitate cheating. We are exploring different mechanisms to detect and prevent misuse, promoting academic integrity.

Transparency in the functioning and limitations of the EPFL Chatbot is essential. We have meticulously documented the development process, the sources of training data, and the methodologies employed. By informing users about the chatbot's capabilities and limitations, we help them set realistic expectations and understand the tool's scope.

Accessibility is a critical concern, and ensuring that the EPFL Chatbot serves all EPFL students, including those with disabilities, remains a priority. While current capabilities do not encompass interaction through sign languages, multiple approaches could be implemented to further enhance accessibility. These approaches may involve integrating the chatbot with sign language recognition and generation technologies. For instance, leveraging video input to interpret sign language gestures would enable the chatbot to process and respond appropriately. Additionally, converting the chatbot's text-based responses into sign language using avatar-based systems represents a promising direction. Such advancements would support our commitment to inclusivity, thereby enhancing the learning experience for all students.

7 Conclusion

The development and evaluation of the EPFL Chatbot have demonstrated the potential of specialized language models in enhancing educational tools tailored to specific curricula. By leveraging fine-tuning and Retrieval-Augmented Generation (RAG) techniques, we successfully adapted a large language model to address the unique needs of EPFL students. Our results indicate that while the base MISTRAL model performed well, the integration of DPO and RAG led to small improvements, particularly in handling multiple-choice questions.

Despite these advancements, our work reveals that optimizing the model for a highly specialized academic environment presents challenges. The modest performance gains from DPO and RAG highlight the need for further refinement of these techniques to achieve substantial improvements. Future work should focus on enhancing the model's ability to process diverse academic queries, integrating more robust context-awareness, and exploring additional relevant data sources to improve response accuracy.

This work contributes to the broader field of educational AI, providing insights into the application of LLMs in academic settings and underscoring the importance of domain-specific adaptation. The lessons learned from this project will hopefully inform future developments in educational technology, helping to create more effective and accessible learning tools for all students.

Team Contributions

The development of the EPFL Chatbot was a collaborative effort with each team member contributing significantly to various aspects of the project:

- **Yacine Chaouch:**
  - Development of Direct Preference Optimization (DPO) datasets.
  - Training with DPO datasets.
  - Development of Supervised Fine-Tuning (SFT) datasets.
  - Training SFT using LoRA adapters and model quantization.
- **Mohamed Charfi:**
  - Implementation of Retrieval-Augmented Generation (RAG).
  - Extensive Evaluation of RAG's retrieval effectiveness and enhancements.
  - Comparison of different models.
  - Experimentation with prompt engineering techniques.
- **Aziz Laadhar:**
  - Development of Direct Preference Optimization (DPO) training.
  - Training and Evaluating DPO.
  - Development of the RAG dataset.
  - Generation of the evaluation dataset.

References

Farid Alijani and Esa Rahtu. 2021. Supervised fine-tuning evaluation for long-term visual place recognition. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bochun Cao. 2023. Ai tutor: Solution for chinas disadvantaged and under-resourced children. *Lecture Notes in Education Psychology and Public Media*, 32:133–141.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model.

Matouš Eibich, Shivay Nagpal, and Alexander Fred-Ojala. 2024. Aragog: Advanced rag output grading.

Sharvi Endait, Srushti Sonavane, Ridhima Sinare, Pritika Rohera, Advait Naik, and Dipali Kadam. 2024. Multilingual evaluation of semantic textual relatedness.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Shijia Huang, Jianqiao Zhao, Yanyang Li, and Liwei Wang. 2023. Learning preference model for LLMs via automatic preference data generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9187–9199, Singapore. Association for Computational Linguistics.

Albert Q. Jiang, 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.

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge.

Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. pages 1–8.

Zheyan Qu, Lu Yin, Zitong Yu, Wenbo Wang, and Xing zhang. 2024. Coursegpt-zh: an educational large language model based on knowledge distillation incorporating prompt optimization.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Hugo Touvron, 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. *arXiv preprint arXiv:2302.13971*.

Ultron. 2020. Iitjee neet aiims students questions data.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Huatuogpt, towards taming language model to be a doctor.