國立臺灣大學電機資訊學院電機工程學系

碩士論文

Department of Electrical Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

利用檢索增強推理解決註冊會計師考試

RAR: Tackling Reasoning-Intensive CPA Exams with Retrieval Augmented Reasoning

賴宥辰

Yu-Chen Lai

指導教授：陳銘憲 博士

Advisor: Ming-Syan Chen, Ph.D.

中華民國 113 年 7 月

July, 2024

doi:10.6342/NTU202402573

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

利用檢索增強推理解決註冊會計師考試

RAR: Tackling Reasoning-Intensive CPA Exams with Retrieval Augmented Reasoning

本論文係 賴宥辰 R11921097 在國立臺灣大學電機工程學系完成之碩士學位論文，於民國 113 年 7 月 24 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Electrical Engineering on 24/7/2024 have examined a Master's Thesis entitled above presented by Yu-Chen Lai R11921097 candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

陳銘憲　　葉彌妍　　賴冠廷

（指導教授 Advisor）

系主任 Director: 李建模

doi:10.6342/NTU202402573

誌謝

能夠順利完成這篇論文，我要感謝許多人的支持與幫助。首先，我要感謝我的指導教授陳銘憲老師，在研究過程中，陳老師細心地審查我的研究是否具有可行性，並且及時糾正我偏離正軌的研究方向，在我遇到研究瓶頸時，陳老師總能以不同的角度分析問題，並提供寶貴的建議，指引我走向正確的研究方向，讓我能夠順利進行實驗。眞的非常謝謝老師讓我有機會加入NetDB的大家庭，跟著有相同目標的同學一起努力做研究眞的很開心。

接著，我要感謝口試委員葉彌妍博士、吳齊人教授以及賴冠廷教授，他們從不同的角度審視我的研究，指出我未曾察覺的不足之處，並提供了許多寶貴的改進意見，我深感榮幸並心懷感激。

此外，我還要感謝老師的助理雅惠姐，在我心情低落時，她總是陪我聊天、帶我去農場散心，給了我很多心靈上的支持。我也要感謝博士班的璽文學長、方睿學長和津源學長經常給予我許多實驗上的建議，毫不保留地分享知識，在我遇到實驗瓶頸時總能適時提供詳細的建議，這對我完成研究有著極大的幫助。還有庭安學姊與世弦學長，謝謝他們在提供建議之餘也常常與我聊天舒壓，讓討論研究變得不再苦悶。

也感謝實驗室的每一位同學，與我同樣研究大型語言模型的立憲、朝棋和順貴，你們是我日常討論研究、互相激勵的好夥伴，還有來自不同研究領域的盈樺、子仙、依庭和于萱，雖然我們常常互相調侃彼此的研究，但你們的歡笑聲始終是我度過艱辛研究時光的動力源泉。碩一的學弟妹們，莉亞、怜均、昱宏、紋慈、佳儀與沛蓉，你們每一個人的支持、鼓勵與陪伴，都讓我在這段旅程中不再孤單。

另外，我還要特別感謝我的高中學妹靖芸，由於我的研究涉及一些會計相關知識，這對我來說是個陌生的領域，感謝你在繁忙的工作之餘，依然抽出時間陪我討論，並提供了許多寶貴的解題點子，對我的研究幫助良多。

最後，我要向我的家人致以深深的感謝，正是他們的包容與支持成就了今天的我。同時，我也要感謝自己，感謝那個在面對挫折時從未放棄、堅持到底的自己。

doi:10.6342/NTU202402573

摘要

人工智慧中大型語言模型（LLM）的使用激增凸顯了它們在文字處理和生成方面的先進能力。然而，它們在會計和金融等專業領域的熟練程度仍然受到審查，特別是在註冊會計師（CPA）考試等複雜任務方面。在美國，CPA考試由美國註冊會計師協會（AICPA）監督，包括四個部分：審計和簽證（AUD）、商業環境與理論（BEC）、財務會計和報告（FAR）以及法規（REG）。過往研究表明，包括ChatGPT在內的LLM在CPA考試中遇到了複雜的問題解決場景和多樣化的問題類型，這表明需要進一步改進才能有效地處理此類特定領域的任務。

爲了解決CPA考試的挑戰，引入了一種稱爲檢索增強推理（RAR）的新方法，將平均通過率從0.5提高到0.62。RAR使用任務路由器將問題分爲知識密集和推理密集型類別。對於知識密集型問題，它使用檢索增強生成（RAG）從外部資料庫中提取相關信息，以提高答案準確性。對於推理密集問題，RAR採用推理行動（ReAct）、代理人（Agent）和思想鏈（CoT）方法，並整合會計Python庫等外部工具，模仿眞實考試環境，有效解決複雜問題。

關鍵字: 註冊會計師考試、檢索增強推理、檢索增強生成、推理行動、代理人、思想鏈。

doi:10.6342/NTU202402573

Abstract

The surge in the use of Large Language Models (LLMs) in artificial intelligence highlights their advanced capabilities in text processing and generation. However, their proficiency in specialized fields, such as accounting and finance, remains under scrutiny, particularly regarding complex tasks like the Certified Public Accountant (CPA) examination. The CPA exam, overseen by the American Institute of CPAs, encompasses four sections: Auditing and Attestation (AUD), Business Environment and Concepts (BEC), Financial Accounting and Reporting (FAR), and Regulation (REG). Research indicates that LLMs, including ChatGPT, struggle with the exam's complex problem-solving scenarios and varied question types, demonstrating the need for further improvement to handle such domain-specific tasks effectively.

To address the challenges of the CPA exam, a new method called Retrieval Augmented Reasoning (RAR) has been introduced, improving the average pass rate from 0.5 to 0.62. RAR employs a task router to classify questions into knowledge-intensive and reasoning-intensive categories. For knowledge-intensive questions, it uses Retrieval Augmented Generation (RAG) to extract relevant information from external databases, enhancing answer accuracy. For reasoning-intensive questions, RAR utilizes ReAct, Agent, and Chain of Thought (CoT) approach, and integrates external tools like the accounting Python library to solve complex problems effectively, mimicking the real exam environment.

Keywords: *CPA, RAR, RAG, ReAct, CoT*

doi:10.6342/NTU202402573

Contents

doi:10.6342/NTU202402573

doi:10.6342/NTU202402573

List of Tables

doi:10.6342/NTU202402573

List of Figures

doi:10.6342/NTU202402573

Chapter 1

Introduction

In artificial intelligence (AI), the use of Large Language Models (LLMs) has surged, becoming an integral part of modern technology. These advanced models, known for their human-like text processing and generation capabilities, have achieved significant success across diverse tasks and are increasingly being applied to specialized fields. Nonetheless, an essential question emerges in the specialized areas of accounting and finance: To what extent are these models proficient in executing complex, domain-specific tasks, such as those encountered in the Certified Public Accountant (CPA) examination?

The CPA exam is a rigorous and comprehensive test designed to assess whether candidates possess the knowledge and skills required for a career in public accounting. Administered by the American Institute of CPAs (AICPA), the exam consists of four sections (Table 1.1): Auditing and Attestation (AUD), Business Environment

Section	CPA Review
AUD	Auditing & Attestation
BEC	Business Environment & Concepts
FAR	Financial Accounting & Reporting
REG	Regulation

Table 1.1: Exam of AICPA.

doi:10.6342/NTU202402573

and Concepts (BEC), Financial Accounting and Reporting (FAR), and Regulation (REG). Each section covers distinct areas of accounting, including ethics, auditing, business law, and financial management.

Some researches [1, 2, 3, 4] have tested the capabilities of language models against the CPA exam, only to find that they fell short of expectations. These models struggled to navigate the complex problem-solving scenarios presented by the exam.

We test ChatGPT [5] (Table 2.1) for the CPA exam reveals it can only solve some questions. The model struggles with both theoretical and calculation questions, indicating significant difficulties across the full range of CPA exam question types. This suggests language model needs further improvement to effectively handle complex and varied exam questions.

doi:10.6342/NTU202402573

Chapter 2

Problems and Contribution

2.1 Problems

Previous studies [1, 2, 3, 4] have evaluated the performance of the language model within the context of the CPA exam. The results indicate that these models do not meet the anticipated standards, as they encounter significant difficulties in navigating the complex problem-solving scenarios presented by the exam. Despite their notable capabilities in general language processing tasks, these language models can not effectively address the specific and intricate challenges of the CPA exam. This highlights the limitations of those models in domain-specific applications and underscores the necessity for further advancements to enhance their proficiency in specialized fields.

In our analysis, we evaluate ChatGPT [5] (Table 2.1) for the CPA exam and find that it can only solve a subset of the questions. We observed that the model performed poorly not only on theoretical and conceptual questions but also encountered a significant performance decrease on calculation questions. This finding indicates that the language model encounters considerable difficulties when addressing the full range of question types present in the CPA exam. Such challenges suggest that while ChatGPT has considerable potential, it requires further enhancement and support

doi:10.6342/NTU202402573

to handle complex and varied exam questions effectively.

	AUD	BEC	FAR	REG	Average
Exam paper 1	0.58	0.36	0.38	0.48	0.45
Exam paper 2	0.56	0.56	0.52	0.6	0.56
Exam paper 3	0.56	0.59	0.44	0.45	0.51
Average	*0.57*	*0.51*	*0.43*	*0.5*	*0.5*

Table 2.1: Pass rate of zero-shot prompting from GPT-3.5.

2.2 Contribution

To solve these problems, we introduction a new method, **R**etrieval **A**ugmented **R**easoning (**RAR**), to solve these problems separately. Our method improves the average pass rate from 0.5 to 0.62.

1. We first use a task router to classify the questions into two primary categories: knowledge-intensive and reasoning-intensive. This distinction is crucial because each category necessitates a unique approach for effective resolution. By recognizing these two categories from the outset, we can apply suitable methods to handle each type efficiently.

    - **Knowledge-intensive questions** encompass numerous areas such as auditing, tax law, and corporate law. This requires a deep comprehension of the material. Each audit procedure offers a multitude of options, necessitating a comprehensive integration of knowledge. This process thoroughly tests one's understanding of the subject matter.

    - **Reasoning-intensive questions** involve complex calculations and task-based simulations requiring extensive knowledge. Candidates must deeply understand and apply auditing, business concepts, financial accounting,

doi:10.6342/NTU202402573

tax law, and current regulations. These questions also demand proficiency in using numerous accounting formulas to solve intricate problems.

2. In addressing **knowledge-intensive questions**, we employ Retrieval Augmented Generation (RAG) [6, 7, 8] to extract relevant information from external databases. We utilize accounting textbooks to build databases. RAG significantly enhances the accuracy of answers by integrating external knowledge, effectively reducing the risk of misinformation. This process not only makes the responses more precise and credible but also ensures that the solutions reflect the complexity and demands of real CPA exam scenarios.

3. For **reasoning-intensive questions**, RAR is designed to closely replicate the exam environment. The zero-shot prompting method was utilized to assess the language models' capacity to address questions without prior exposure to examples. In contrast, the Chain of Thought (CoT) [9] approach facilitated the model's ability to decompose intricate problems into simpler, more manageable steps. Moreover, we integrated external tools by Agent from ReAct [10], including the accounting Python library [11] with PythonREPLTool from LangChain [12], which can write and execute code to solve math-related problems. It can dynamically generate and execute Python code on demand adds a significant dimension to problem-solving by seamlessly integrating theoretical knowledge with practical application.

Chapter 3

Related Works

3.1 Use ChatGPT to take the CPA exam

Zero-shot prompting [13] evaluates the language model's capability to address questions without the provision of preceding examples. The findings[2, 3] reveal that the models underperform relative to the anticipated benchmarks, facing substantial difficulties when confronted with the complex problem-solving demands of the exam. While these models exhibit strong capabilities in general language processing tasks, they prove inadequate in addressing the unique and intricate challenges posed by the CPA exam. This outcome illuminates the constraints of current models in domain-specific applications, highlighting a critical need for further advancements to enhance their effectiveness in specialized contexts.

Chain of Thought (CoT) [9] is a reasoning process that breaks down complex problems into simpler steps, enhancing problem-solving accuracy and improving decision-making and output quality. This approach involves presenting the model with several examples that explicitly delineate the step-by-step reasoning process.

3.1.1 Retrieval Augmented Generation (RAG)

RAG [6, 7, 8] (Figure 4.4) has emerged as a promising solution by incorporating knowledge from external databases. This approach not only enhances the accuracy and reliability of generated content but is also particularly beneficial for tasks that require extensive knowledge. By continuously updating knowledge and integrating domain-specific information, RAG significantly improves the accuracy and credibility of content generation. This method addresses the limitations of traditional generative models by providing a dynamic and adaptable framework that can evolve with new information and specialized requirements.

3.1.2 ReAct

ReAct [10], which stands for Reasoning and Acting, enhances the accuracy of responses from language models by integrating reasoning and action. This method diverges from traditional language models that primarily focus on text generation, by embedding a mechanism that allows the model to interact with external tools and databases. This interaction enables the model to gather real-time data and apply logical reasoning to generate more precise and contextually relevant responses. Furthermore, ReAct's architecture includes a feedback loop where the model's actions influence subsequent reasoning processes. This iterative approach not only improves response accuracy but also enhances the model's capability to handle complex queries that require multi-step reasoning.

3.1.3 Agent

Agent [14, 15, 16, 17] refers to a model equipped with reasoning and decision-making capabilities, capable of receiving environmental variables (such as user inputs, external data, etc.) as input and generating responses (such as dialogue replies, decision recommendations, etc.) as output. It is designed to adhere to the concept

doi:10.6342/NTU202402573

of cyclical learning, continually learning and improving through interaction with its environment.

It can function as a general task model or as an advanced integrated model incorporating a language model. By integrating with various data sources such as databases, API interfaces, sensors, and more, the agent can be applied to a wide range of scenarios, including dialogue systems, knowledge queries, autonomous driving, and other applications.

doi:10.6342/NTU202402573

Chapter 4

Methodology

The methodology pipeline of our approach involves an initial classification step based on GPT-3.5 to differentiate between various questions. Depending on the question type, we apply an appropriate solution strategy. For knowledge-intensive questions, we employ the RAG [6, 7, 8] technique to supplement the knowledge gaps of the language model. For reasoning-intensive questions, we utilize the RAR in conjunction with the PythonREPLTool and CoT methodology to accurately execute and solve the necessary calculation steps. Please refer to our workflow in Figure 4.1 for more details.

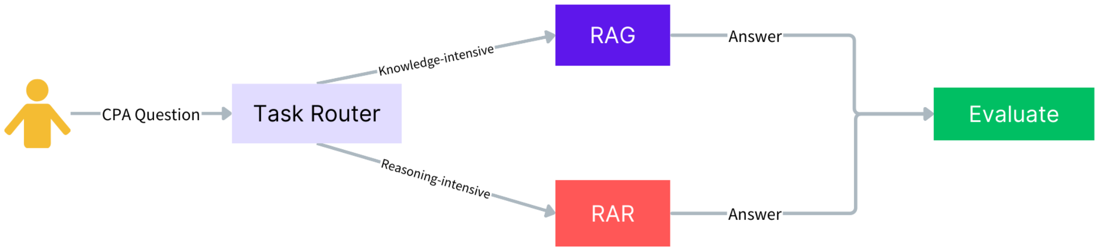
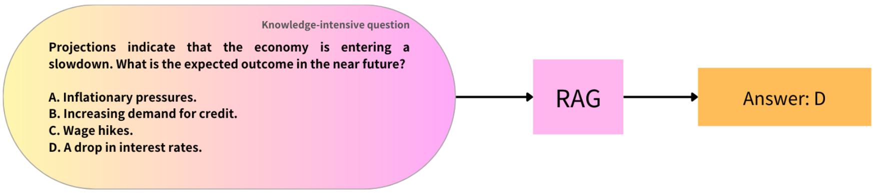


Figure 4.1: Problem solving steps.

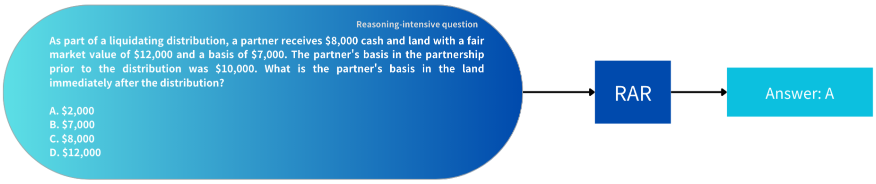
doi:10.6342/NTU202402573



Projections indicate that the economy is entering a slowdown. What is the expected outcome in the near future?

A. Inflationary pressures.
B. Increasing demand for credit.
C. Wage hikes.
D. A drop in interest rates.

Figure 4.2: Example question of knowledge-intensive questions.



As part of a liquidating distribution, a partner receives $8,000 cash and land with a fair market value of $12,000 and a basis of $7,000. The partner's basis in the partnership prior to the distribution was $10,000. What is the partner's basis in the land immediately after the distribution?

A. $2,000
B. $7,000
C. $8,000
D. $12,000

Figure 4.3: Example question of knowledge-intensive questions.

4.1 Task router for classification

Initially, we classify the questions into two primary categories: **knowledge-intensive questions** (Figure 4.2) and **reasoning-intensive questions** (Figure 4.3). This differentiation is crucial as each category necessitates a distinct approach for effective resolution. We use GPT-3.5 as a task router, assigning the appropriate process for each question type. It accurately discerns the category of each question, ensuring the application of the correct method for effective resolution. Knowledge-intensive questions necessitate a profound understanding of specific information, while reasoning-intensive questions require a systematic logical process to determine the correct answer. By identifying these two categories at the outset, we can apply suitable methods to address each type effectively.

doi:10.6342/NTU202402573

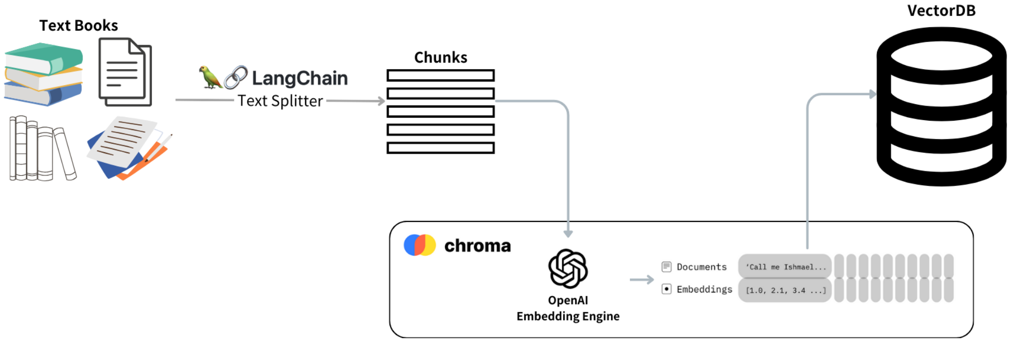


Figure 4.4: Pipeline of RAG.

4.2 RAG for knowledge-intensive questions

To solve the knowledge limitations of the language model, particularly for CPA exam questions, we design a framework incorporating the RAG technique. As Figure 4.4, the detailed process of RAG involves using the PyMuPDFLoader module of Langchain [12] to load eight compulsory accounting textbooks and merge all books' content into a single file set. Since directly processing the entire text may be too large and difficult to manage effectively, especially since some books can exceed two thousand pages, we use Langchain's RecursiveCharacterTextSplitter module to divide these textbooks into smaller blocks. Each block contains approximately two hundred words, with a ten-word overlap between chunks to ensure contextual coherence.

Next, we utilize OpenAI's embedding model, OpenAIEmbeddings [18], to generate embedded representations of these text chunks. This embedding model converts the PDF textbook content into numerical vectors, representing the semantic content of the text. We use Chroma [19], an open-source embedded database that is lightweight and easy to use, to store these embeddings. Chroma facilitates the integration of knowledge, facts, and skills, making it ideal for building LLM applications. It can run in memory, save to disk, or function as a database server,

doi:10.6342/NTU202402573

providing versatile options for data management. By storing the embeddings on our computer's hard drive using the Chroma library, we enable quick loading of the embeddings for subsequent retrieval and reuse. This approach significantly reduces both the time and cost associated with these experiments.

By integrating RAG, we enable the model to retrieve relevant information from external sources, thereby augmenting its responses with accurate and comprehensive data. This approach not only compensates for the inherent knowledge gaps in language model but also enhances its ability to provide precise and contextually relevant answers. This is particularly beneficial for the CPA exam, where extensive domain-specific knowledge is crucial for solving complex and nuanced questions.

4.3 RAR for reasoning-intensive questions

In addressing reasoning-intensive questions, the RAG approach alone is inadequate, as these questions necessitate both cognitive and computational steps. Additionally, certain problems remain unresolved through these methods due to the language model's lack of knowledge regarding the appropriate formulas to apply, leading to inaccuracies in its responses. Due to the limitations in addressing reasoning-intensive questions with knowledge gaps, it is necessary to provide the language model with the missing formulas to ensure it operates correctly. However, we have observed that when provided with text-based formulas, the language model sometimes fails to understand or apply them properly.

Our method, **Retrieval Augmented Reasoning (RAR)** (Figure 4.5), not only integrates a textbook database but also addresses this issue by using Python-code-based information [11]. Initially, we extract knowledge from textbooks to build a knowledge base tailored to specific questions. Besides, the language model also searches the Python library for similar formulas and scripts based on questions that can be applied. The formulas are written in Python code and de-identified, enabling

doi:10.6342/NTU202402573

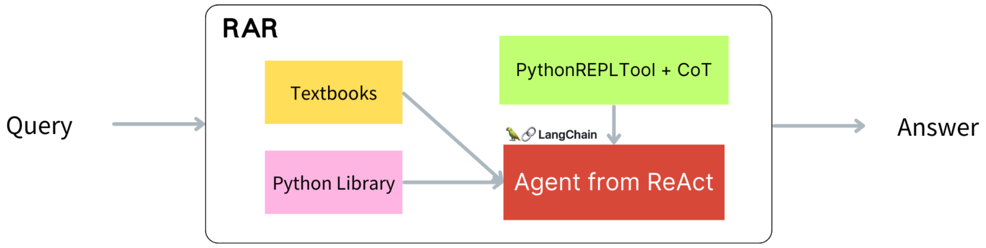


Figure 4.5: Detail of RAR.

the language model to process and utilize them more effectively. Supplying these essential formulas enhances the agent's ability to generate accurate and reliable solutions, improving our methodology's effectiveness and precision.

To further facilitate the step-by-step reasoning process within the CoT, we propose a more complex method that integrates an external tool, Agent from ReAct [10] with PythonREPLTool from LangChain [12]. This approach allows the language model to first retrieve relevant and useful additional knowledge from the vector database of the textbook and related Python script from the Python library after analyzing the question. The retrieved information is then combined with the question and input into the language model. This comprehensive method ensures that the language model can accurately address reasoning-intensive questions by leveraging both external knowledge and Python-based computational tools.

4.3.1 Agent from ReAct

ReAct [10], which stands for Reasoning and Acting, enhances the accuracy of responses from language models by integrating reasoning and action. This methodology enables an agent to instruct the language model on the necessary actions, specify the tools to be utilized, and ultimately collect the answers generated by these tools. The tool we use is PythonREPLTool, an auxiliary tool provided by Langchain, enables the language model to dynamically execute code within the

Python programming environment. It allows Agent to obtain and return the execution results for further evaluation. Agent then assesses these results and determines the subsequent course of action. This iterative process enables Agent to adapt and respond to the evolving problem-solving landscape, ensuring that each step is informed by the most current data and computational outcomes.

4.3.2 Chain of Thought (CoT)

The goal of CoT [9] is to break down complex problems into simpler and more manageable sub-problems, allowing the language model to consider each step sequentially. By combining PythonREPLTool and CoT, language model is compelled to decompose the knowledge obtained from RAG into sub-step calculations through Python code during the problem-solving process. It can debug and rapidly iterate on the results obtained each time, thereby accelerating the research process and improving the reliability of the outcomes. This approach not only compensates for the inherent knowledge gaps in language model but also enhances its ability to provide precise and contextually relevant answers, particularly beneficial for the CPA exam, where extensive domain-specific knowledge is crucial for solving complex and nuanced questions.

doi:10.6342/NTU202402573

Chapter 5

Experiments

5.1 Datasets and experiment setting

For our evaluation, we collect a dataset of 380 multiple-choice questions from three trusted CPA exam references. The dataset includes 101 AUD questions, 82 BEC questions, 109 FAR questions, and 88 REG questions. Each question is converted into a JSON object, encompassing vital metadata such as the main question, options, answers, and other pertinent information. This structured format enable us to easily evaluate the LLM's performance across various domains of the CPA exam.

We use GPT-3.5 as the language model tool for this evaluation, leveraging its advanced capabilities to analyze and respond to the questions.

5.2 Zero-shot prompting

Initially, as observed in Table 2.1, the average pass rates of the language model's responses are 0.57, 0.51, 0.43, and 0.5 for AUD, BEC, FAR, and REG, respectively, with an overall average of 0.5. These results indicate that the language model does not achieve optimal performance across the various types of questions. This finding suggests that, when used alone, the language model struggles to produce consistently accurate results. Such findings underscore the necessity of enhancing the model's

doi:10.6342/NTU202402573

capabilities through supplementary methods and tools.

Except for the AUD type, which consists solely of knowledge-intensive questions, the other types encompass both knowledge-intensive questions and reasoning-intensive questions.

5.3 Results

5.3.1 Comparison of zero-shot prompting, RAG, and RAR

The first row of Table 5.2 and the green bar in Figure 5.1 present the zero-shot prompting results, which yield an average pass rate of 0.5. This outcome highlights the language model's suboptimal performance in answering questions without any external assistance across various types of questions. The consistently low pass rate underscores the limitations of the language model when operating in isolation, revealing its struggles with the complexity and diversity of the questions. This emphasizes the need for supplementary methods to enhance its accuracy and reliability. These improvements are essential to bridge the gap between the model's current performance and the high standards required for effective problem-solving in real-world applications.

The second row of Table 5.2 and the blue bar in Figure 5.1 demonstrate how the use of Retrieval-Augmented Generation (RAG) can improve the accuracy of the language model's responses. However, upon examining the correct responses distinct from zero-shot prompting and RAG, it is evident that most of these pertain to knowledge-intensive questions, indicating that RAG does not effectively address reasoning-intensive questions. Notably, for the AUD type, which consists solely of knowledge-intensive questions, RAG contributes to a significant improvement, with an average 12% increase in the correct answer rate. Specifically, RAG achieves improvements of 7%, 12%, and 19% on each exam paper, as shown in Table 5.1. This substantial enhancement underscores the effectiveness of RAG in augmenting the

doi:10.6342/NTU202402573

language model's capabilities by providing relevant external information, particularly for questions that demand extensive domain-specific knowledge.

	Zero-shot	RAG
Exam paper 1	0.58	0.65
Exam paper 3	0.56	0.68
Exam paper 3	0.56	0.75
Average	*0.57*	*0.69*

Table 5.1: Use RAG on type AUD.

Except for the AUD type, which consists solely of knowledge-intensive questions, other types include both knowledge-intensive and reasoning-intensive questions. The third row of Table 5.2 and the yellow bar from Figure 5.1 illustrate that RAR can further enhance performance. Specifically, RAR improves the accuracy for BEC, FAR, and REG by 3%, 8%, and 2%, respectively, providing an additional 5% improvement for all questions overall.

	AUD	BEC	FAR	REG	Average
Zero-shot	0.57	0.51	0.43	0.5	0.5
RAG	0.69	0.62	0.47	0.51	0.57
RAR	0.69	0.65	0.55	0.53	0.62

Table 5.2: Comparison of zero-shot and my method for whole questions.

5.3.2 RAG on knowledge-intensive questions

We separated the knowledge-intensive questions from each type for this experiment. Table 5.3 shows the proportion of knowledge-intensive questions through whole exam papers. Table 5.4 and Figure 5.2 present the results using accounting textbooks as the database. The grey bar shows that RAG can help improve

doi:10.6342/NTU202402573

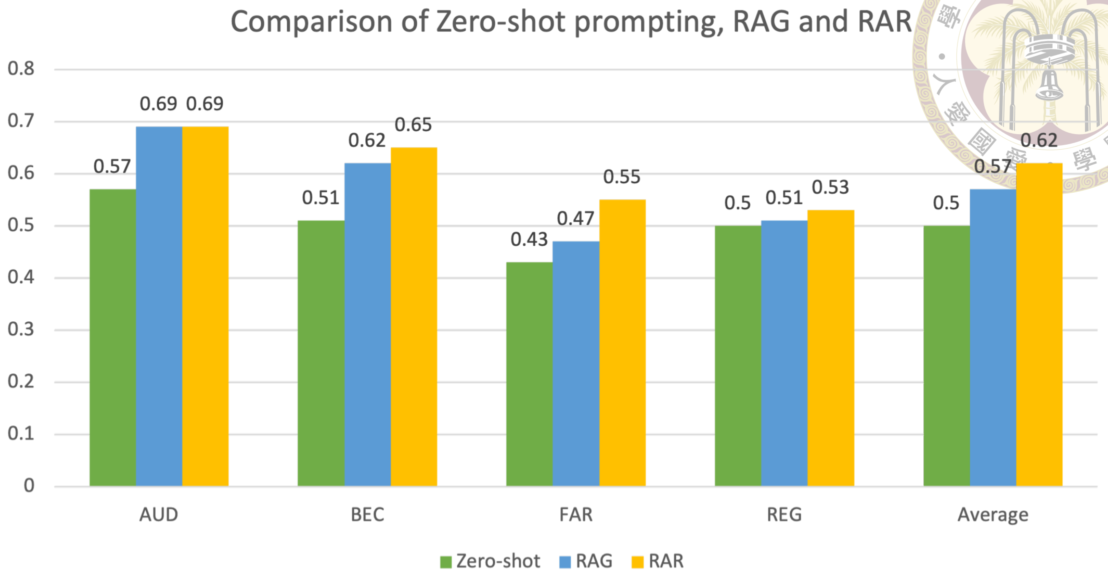


Figure 5.1: Bar chart of the comparison.

accuracy slightly for each question type, yielding an average improvement of 6%. Specifically, RAG achieves 12%, 1%, 5%, and 4% improvements for the AUD, BEC, FAR, and REG question types, respectively. This demonstrates the effectiveness of RAG in leveraging external knowledge sources to enhance the performance of the language model, particularly in handling knowledge-intensive queries. The consistent improvement across different question types highlights RAG's capability to provide relevant and precise information, thereby augmenting the language model's ability to generate accurate and reliable responses.

	AUD	BEC	FAR	REG
Exam paper 1	1	0.36	0.5	0.48
Exam paper 2	1	0.92	0.56	0.56
Exam paper 3	1	0.81	0.68	0.53

Table 5.3: Proportion of knowledge-intensive questions.

doi:10.6342/NTU202402573

	AUD	BEC	FAR	REG	Average
Zero-shot	0.57	0.67	0.48	0.65	0.59
RAG	0.69	0.68	0.53	0.69	0.65

Table 5.4: Use RAG for knowledge-intensive questions.

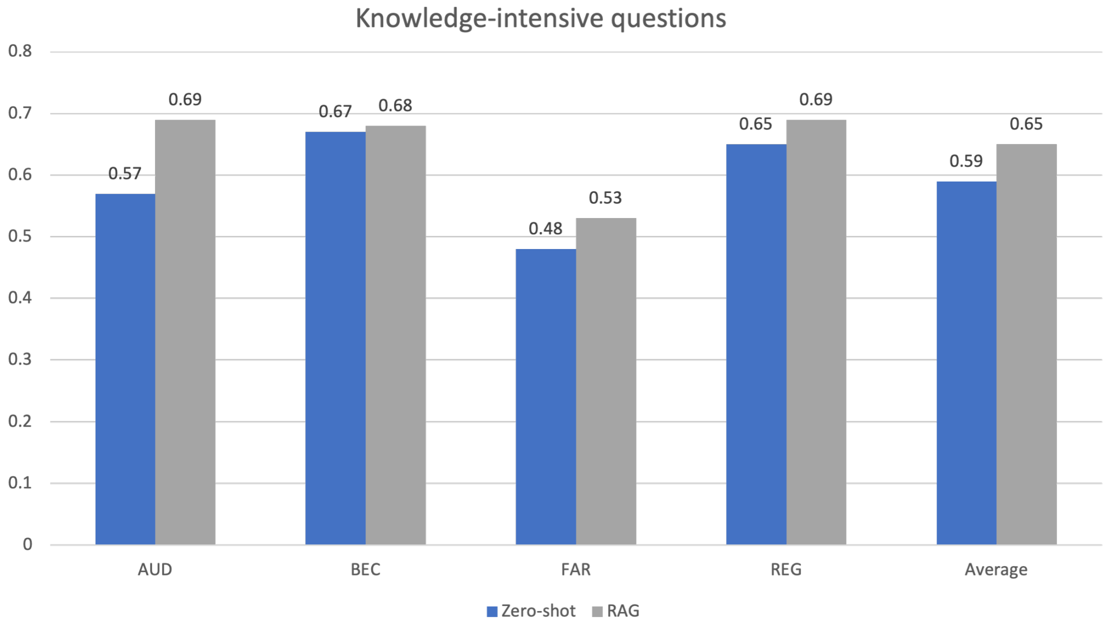


Figure 5.2: RAG can improve solving knowledge-intensive questions in each type.

5.3.3 RAR on reasoning-intensive questions

Agent from ReAct integrates the PythonREPLTool and CoT methodology, which can be thought of as decomposing a larger problem into several intermediate steps. Table 5.5 shows the proportion of reasoning-intensive questions through whole exam papers. By breaking down complex problems in this manner, the agent can effectively utilize additional information and Python formulas provided by RAR to arrive at the final answer.

As the results shown in Table 5.6 indicate, our method substantially improves performance, with improvements 24%, 31% and 10% for BEC, FAR, and REG respectively. On average, there is a 22% improvement per exam section. This sig-

doi:10.6342/NTU202402573

nificant enhancement demonstrates the effectiveness of our approach in addressing the challenges posed by reasoning-intensive questions.

Figure 5.3 illustrates that the language model performs better when using RAR on reasoning-intensive questions. The bar chart highlights the performance difference between the zero-shot prompting, RAG, and RAR, showcasing the advantages and effectiveness of our approach.

	AUD	BEC	FAR	REG
Exam paper 1	0	0.64	0.5	0.52
Exam paper 2	0	0.08	0.44	0.44
Exam paper 3	0	0.19	0.32	0.47

Table 5.5: Proportion of Reasoning-intensive questions.

	BEC	FAR	REG	Average
Zero-shot	0.38	0.32	0.37	0.35
RAG	0.49	0.29	0.38	0.37
RAR	0.62	0.63	0.47	0.57

Table 5.6: Use RAR for answering reasoning-intensive questions.

5.3.4 More analysis

Additionally, Figure 5.3 also illustrates the visualization comparison results of our experiment when using RAG and RAR on reasoning-intensive questions. As observed in the previous experiment, RAG provided significant enhancements. However, for the FAR type, experimental results unexpectedly showed a reduction in the pass rate. Table 5.7 presents the detailed experimental results for the FAR type from each exam paper, revealing that two of the papers experienced a decrease in the pass rate. This anomaly suggests that while RAG generally improves perfor-

doi:10.6342/NTU202402573

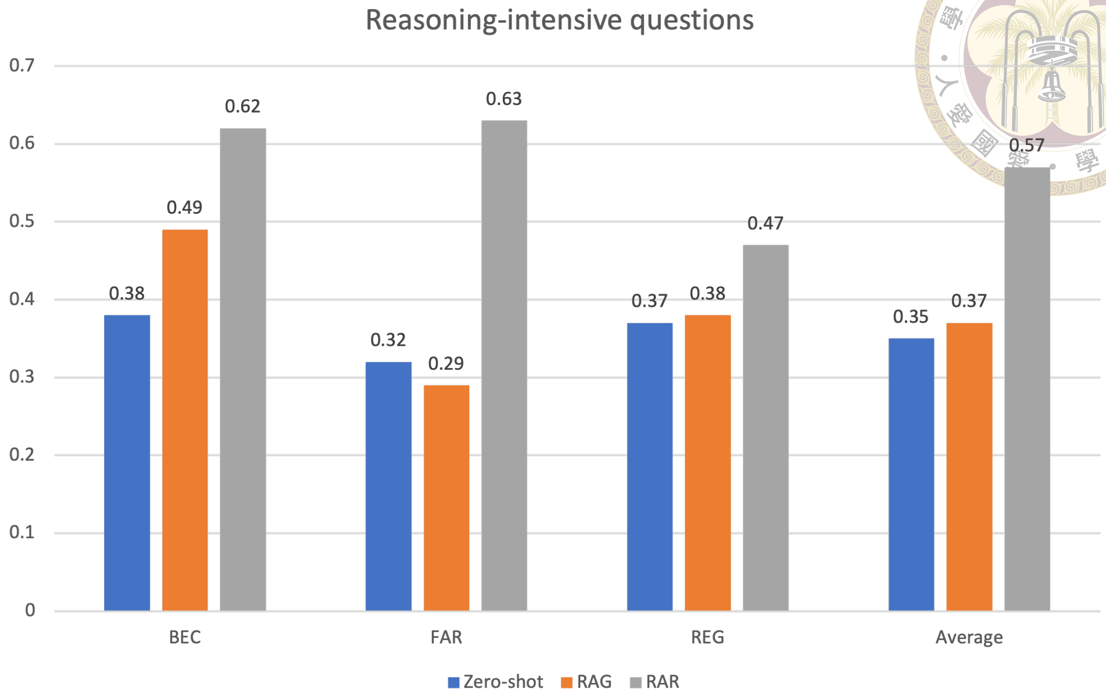


Figure 5.3: RAR can help improve the accuracy on reasoning-intensive questions.

mance, it may not be universally effective across all reasoning-intensive questions, highlighting the significant benefits of our RAR method in improving response accuracy.

We consider that textbooks often organize content in a linear and chapter-wise manner, where information can be very detailed and scattered. This organization can cause the model to have difficulty retrieving relevant information, as it may be spread across multiple chapters or sections. Additionally, textbooks cover a wide range of topics, but not necessarily in enough depth to address specific questions. Some questions may require more in-depth or specific information, and the content provided by a textbook may be too general or insufficiently detailed. Furthermore, the language model may extract numbers from other topics in the textbook, leading to calculation misunderstandings.

Moreover, textbooks are updated less frequently and may not reflect the latest research findings or the current body of knowledge. If a question involves the lat-

doi:10.6342/NTU202402573

est scientific developments or dynamic knowledge, the information provided by the textbook may lag behind current advancements. These challenges highlight areas for potential improvement in our approach, and addressing these issues could be a focus for future work.

	zero-shot	RAG
exam 1	0.28	0.38
exam 2	0.45	*0.2*
exam 3	0.27	*0.18*
Average	0.32	0.29

Table 5.7: RAG result on type FAR reasoning-intensive questions.

Chapter 6

Conclusion

We separated the CPA exam questions into two categories: knowledge-intensive questions and reasoning-intensive questions, and addressed each category individually. To decrease the knowledge gap of the language model and improve the pass rate for knowledge-intensive questions, we utilized RAG. For reasoning-intensive questions, we incorporated our proposed method, RAR, along with the Agent from ReAct to assist PythonREPLTool and CoT in solving these complex problems. This approach allows the decomposition of knowledge into sub-step calculations through Python code during the problem-solving process, ensuring that each intermediate step is addressed with relevant and precise data. This method enhances the overall accuracy and reliability of the solution.

doi:10.6342/NTU202402573

Chapter 7

Future Works

In future work, we plan to use this pipeline to solve questions from different domains. By leveraging different textbooks and related Python libraries, we can easily adapt the pipeline to address various subject areas. This flexibility will allow us to extend the current methodology beyond the domain of accounting, applying it to fields such as physics or chemistry. By integrating domain-specific knowledge bases and computational tools, we aim to enhance the language model's ability to generate accurate and contextually relevant responses across a broad spectrum of disciplines. This adaptability demonstrates the potential of our approach to be a versatile tool for addressing a wide range of complex, domain-specific problems.

Besides PythonREPLTool, we can integrate additional external tools for the Agent's use. For example, LangChain can be employed to add WolframAlpha, an answer engine developed by Wolfram Research. This tool can address factual queries by computing answers from externally sourced data.

doi:10.6342/NTU202402573

References

[1] Ahmedhelow. Exploring llms' performance on accounting exams. Published in Sage Ai, March 27 2024.

[2] Marc Eulerich, Aida Sanatizadeh, Hamid Vakilzadeh, and David A. Wood. Is it all hype? chatgpt's performance and disruptive potential in the accounting and auditing industries, November 17 2023. Available at SSRN: `https://ssrn.com/abstract=4452175` or `http://dx.doi.org/10.2139/ssrn.4452175`.

[3] Chris Gaetano. We had chatgpt take the cpa exam — and it failed. Accounting Today, May 08 2023.

[4] Marc Eulerich, Aida Sanatizadeh, Hamid Vakilzadeh, and David Wood. Can artificial intelligence pass accounting certification exams? chatgpt: Cpa, cma, cia, and ea? *SSRN Electronic Journal*, 01 2023.

[5] OpenAI. Gpt-3.5 technical report, 2022.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[7] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 13–18 Jul 2020.

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[10] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[11] ekmungai. Python accounting, 2024.

[12] LangChain Contributors. Langchain. `https://github.com/hwchase17/langchain`, 2024.

[13] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2152–2161, Lille, France, 07–09 Jul 2015. PMLR.

[14] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995.

doi:10.6342/NTU202402573

[15] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. arxiv 2022. *arXiv preprint arXiv:2205.06175*, pages 1–40.

[16] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

[17] Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, et al. Agentgym: Evolving large language model-based agents across diverse environments. *arXiv preprint arXiv:2406.04151*, 2024.

[18] OpenAI. Openai embedding model. `https://platform.openai.com/docs/guides/embeddings`, 2024.

[19] Anton Troynikov Suvansh Sanjeev. Chroma technical reprot. `https://github.com/chroma-core/chroma`, `https://www.trychroma.com/`, 2024.

doi:10.6342/NTU202402573