AI-Enabled Tax Assistance for Low/Moderate Income Taxpayers: An Evaluation of RAG-based LLMs for VITA Volunteer Support

Sina Gogani-Khiabani, University of Illinois Chicago, USA
Rohan Sai Buddhi, University of Illinois Chicago, USA
Yogesh Dabral, University of Illinois Chicago, USA
ShinPing Chyi, CPA, El Paso, USA
Ashutosh Trivedi, University of Colorado Boulder, USA
Saeid Tizpaz-Niari, University of Illinois Chicago, USA (Contact Author: saeid@uic.edu)

Abstract

The complexity of U.S. tax law presents significant challenges for Volunteer Income Tax Assistance (VITA) volunteers who assist low- and moderate-income taxpayers. This paper investigates the accuracy and reliability of AI systems in aiding VITA volunteers to prepare tax returns. In particular, the research presents a system design and a tool that incorporates Large Language Models (LLMs) and retrieval-augmented generation (RAG) to provide real-time, accurate, and context-aware support. We evaluate the efficacy and efficiency of our RAG-based AI system using the VITA certification tests (a dataset of 130 questions from IRS Form 6744). Our findings reveal key insights into RAG's effectiveness in this domain: it considerably improves accuracy by augmenting relevant IRS tax documents in generating responses, and crucially, it enables smaller LLMs to address complex open-ended numerical questions that are entirely intractable for the baseline models. Our evaluation demonstrates that while RAG boosts overall accuracy, performance on advanced scenarios with complex numerical calculations requires further investigation. We hope that our RAG-enabled AI-software solution enhances the quality, efficiency, and accessibility of tax services for US taxpayers, including those from underserved communities.

1 Introduction

Advancements in AI continue to push boundaries of AI-human collaborative problem-solving in a wide range of applications and tasks, ranging from finance and manufacturing to healthcare. The capabilities in solving challenging socio-technological problems in high-dimensions regardless of input data type (e.g., code, images, text, audio) have enabled widespread adoption of applications in the medical domain [11], planning [17], tax preparation [19, 1, 3, 15, 13], and finance/banking [2].

While large language models (LLMs) have made substantial impacts across different domains, the applicability of this technology to aid tax-related questions requires further investigation [14]. In particular, the IRS Volunteer Income Tax Assistance (VITA) program aims to provide free tax preparation services to low and moderate income taxpayers annually. Despite extensive training, VITA volunteers often struggle to handle the complexities of tax law, particularly with nuanced regulations and frequent updates. Existing resources are static and lack the dynamic adaptability required to address the full spectrum of taxpayer scenarios. Hence, there is an urgent need for intelligent, adaptive tools to assist VITA volunteers.

15th Annual IRS/TPC Joint Research Conference on Tax Administration (IRS-TPC 2025).

We posit that advances in LLMs, particularly retrieval-augmented generation (RAG), present an opportunity to address these challenges and enhance tax assistance for low/moderate income taxpayers within the VITA program. While RAG-based LLMs have shown promise in general-purpose question answering, their potential to support the domain-specific tax preparation remains underexplored.

Recently, the authors have investigated the accuracy of LLMs in answering tax-related questions [4] taken from VITA certification tests [8], employed by the U.S. Internal Revenue Service (IRS) to certify tax volunteers who assist low/moderate income taxpayers. However, the complexity of tax law combined with the requirements for sophisticated numerical reasoning present challenges for LLMs in this domain. **This paper aims to design an AI-enabled tax assistance software through retrieval-augmented generation to assist VITA volunteers in resolving tax returns of eligible taxpayers within the VITA program.** The primary goals are to evaluate the baseline LLMs vs. RAG-enabled LLMs for their ability in responding accurately to tax-related queries, as well as to explore prompting strategies in improving the reasoning and accuracy of our RAG-based LLM system.

To achieve these goals, this paper details the design and evaluation of a RAG-enhanced LLM system tailored for VITA scenarios. We first establish a performance baseline using a selection of smaller, accessible LLMs (including Meta's Llama 3.1 8B/70B, OpenAI's GPT-4o mini, and Anthropic's Claude 3.5 Haiku) evaluated directly on the VITA certification test questions [8]. We then introduce our RAG system, which leverages the IRS documents and guidance (Publication 4491 [7]) to provide relevant, up-to-date context to these LLMs. Our approach employs a multi-step prompting strategy, dissecting the question-answering process into distinct tasks handled by separate LLM instances, both with and without RAG augmentation. The subsequent sections will elaborate on the specific data sources, the RAG architecture, the prompting techniques, and the comparative evaluation results derived from the standardized VITA test cases.

Our comparative evaluation highlights the benefits of the RAG approach. The results demonstrate that augmenting LLMs with targeted retrieval from the IRS documents leads to great improvements (12% on average) in accuracy compared to the baseline LLMs' performance across various question types. Notably, the RAG system enables the models to tackle open-ended numerical questions that have been found to be challenging for the LLMs. These results provide quantitative evidence for RAG's potential to overcome key limitations of LLMs in the complex tax preparation domain.

2 Methodology

This section details the methodology employed to evaluate the efficacy of a Retrieval-Augmented Generation (RAG) system in enhancing the performance of selected smaller Large Language Models (LLMs) on U.S. tax law questions. We first describe the standardized VITA certification test dataset used for evaluation and the process for establishing ground truth answers. Subsequently, we introduce the baseline LLMs chosen for this study. The core of our methodology involves the detailed design of our RAG system, including knowledge source preparation, indexing, and a multi-stage retrieval process. Finally, we outline the multi-step prompting strategies and experimental workflows used to assess both the baseline LLM performance and the performance of the RAG-enabled LLM system.

15th Annual IRS/TPC Joint Research Conference on Tax Administration (IRS-TPC 2025).

2.1 Evaluation Dataset and Ground Truth

Our analysis is derived from the Internal Revenue Service (IRS) Volunteer Income Tax Assistance (VITA) certification tests, as outlined in the IRS Form 6744 for the 2022 tax year [8]. Specifically, we utilize the complete dataset of 130 questions included in this publication, which are divided into two main categories: **60 basic scenario questions** and **70 advanced scenario questions**. The basic questions comprise 32 True/False (T/F), 24 Multiple Choice (MC), and 4 Open-Ended (OE) items. The advanced questions consist of 29 T/F, 29 MC, and 12 OE items. These questions are designed to assess the comprehension and application of tax law for volunteers who wish to assist taxpayers, particularly those with low to moderate income.

It is important to note that the answers to these questions are not publicly available, adding a layer of complexity to the task of evaluating the performance of large language models in this domain. To overcome this challenge, two of the authors (a Certified Public Accountant (CPA) and an IRS-certified tax preparer) independently answered all 130 questions, and then they met for 3 hours and used a consultation to reach agreements on all the answers [4].

The chosen scenarios are deliberately focused on taxpayers under specific circumstances, emphasizing scenarios tailored towards low-income taxpayers. This approach not only aligns with the practical applications of these models in assisting vulnerable communities but also presents a unique challenge in assessing the models' ability to navigate the intricacies of tax law.

2.2 Large Language Models

This study evaluates a selection of smaller Large Language Models (LLMs), both open-source and closed-source options. This deliberate focus on more compact models is driven by two key considerations. Firstly, it aligns with our overarching goal of developing a freely accessible AI-powered tax assistant for low- and middle-income individuals, where the resource efficiency of smaller models and confidentiality of open-source LLMs are crucial. Secondly, smaller LLMs, with their typically more limited inherent knowledge, provide a clearer baseline against which the performance improvements afforded by our Retrieval-Augmented Generation (RAG) system can be distinctly observed and quantified. This allows for a more targeted assessment of RAG's efficacy in augmenting model capabilities within the specialized tax domain.

Llama 3.1 8B (Meta): This is an open-source LLM developed by Meta. It offers a smaller parameter size (8 billion) compared to larger models like GPT-4, making it potentially more accessible for research and deployment in resource-constrained environments. Its performance on complex tasks like tax-related question answering is a key aspect of our evaluation.

Llama 3.1 70B (Meta): Also developed by Meta, Llama 3.1 70B offers a larger parameter size (70 billion) compared to the 8B version. This increased scale often translates to improved performance on complex tasks. Our study investigates whether this holds true for tax-related questions and how its performance compares to both closed-source and smaller open-source models.

GPT-4o mini (OpenAI): Developed by OpenAI, This smaller closed-source model allows us to explore the trade-off between model size and performance in the tax domain. Its reduced computational requirements make it potentially suitable for deployment on devices with limited resources.

Claude 3.5 Haiku (Anthropic): Developed by Anthropic, Claude 3.5 Haiku is another smaller closed-source model included in our evaluation to further investigate the impact of model size on performance in the context of tax-related questions.

2.3 RAG System Architecture

To provide the LLMs with accurate, relevant, and up-to-date context for answering VITA-specific tax questions, we designed and implemented a Retrieval-Augmented Generation (RAG) [10] system. This system integrates external knowledge from IRS documents directly into the LLM's response generation process. The architecture involves careful preparation of the knowledge source, a robust indexing and vectorization strategy, and a multi-stage retrieval pipeline to ensure high-quality context is provided.

2.3.1 Knowledge Source and Preprocessing

The primary knowledge source for our RAG system is the **IRS Publication 4491, VITA/TCE Volunteer Assistor's Guide**, specifically for the tax year 2022 corresponding to our evaluation dataset [7]. This comprehensive guide provides detailed tax law explanations, procedures, and examples directly relevant to VITA volunteers.

Initial processing involved converting the official PDF document into Markdown format using the 'LlamaParse' library. Recognizing that raw document conversions often contain elements potentially confusing to LLMs (such as embedded exercises, sample interviews, unanswered questions, or complex tables not directly related to core tax rules), a crucial cleaning step was performed. We employed regular expressions to identify and remove these extraneous sections from the Markdown file. This cleaning process targeted specific patterns (e.g., "Exercise", "Taxpayer Interview", "Questions") and removed content associated with them until the next structural heading was encountered, thereby preserving the core informational text while improving its suitability for LLM consumption.

2.3.2 Vectorization and Indexing

Following preprocessing, the cleaned Markdown document was chunked for efficient vectorization. We adopted a **heading-based chunking strategy**, where the text content residing under each Markdown heading (denoted by '#', '##', etc.) constitutes a single chunk. This approach aims to maintain semantic coherence within chunks, assuming that content under a specific heading relates to a distinct topic or subtopic. The hierarchical heading structure itself was retained as metadata associated with each chunk.

Each text chunk was then converted into a high-dimensional vector representation using OpenAI's `text-embedding-large` embedding model. This model was chosen for its strong performance on various embedding benchmarks [6]. These vector embeddings were stored and indexed locally using **ChromaDB**[5], an open-source vector database optimized for efficient similarity searches.

2.3.3 Multi-Stage Retrieval Process

To retrieve the most relevant context for a given query we implemented a **two-stage retrieval process** designed to balance recall and precision:

1. **Stage 1: Dense Retrieval (Recall-focused):** Upon receiving a query, the system first performs a similarity search against the ChromaDB index using the same `text-embedding-large` model to embed the query. ChromaDB retrieves an initial, relatively large set of the **top 500** most similar document chunks based on cosine similarity. This broad initial retrieval aims to maximize the chance of capturing all potentially relevant information (high recall) [9].

2. **Stage 2: Reranking (Precision-focused):** The initial 500 candidate chunks are then passed to a more computationally intensive but fine-grained reranking step. We utilize a **Cross-Encoder model** (`ms-marco-MiniLM-L-6-v2`) [16]. Unlike embedding models that compare vector similarity, cross-encoders directly process pairs of (query, document chunk) to compute a more accurate relevance score. The cross-encoder scores the relevance of each of the 500 chunks to the specific query. The chunks are then reranked based on these scores, and the **top 5** highest-scoring chunks are selected as the final context. This reranking step enhances the precision of the retrieved context [12].

These final top 5 document chunks represent the specific, relevant context extracted from Publication 4491, which is then provided to the downstream LLM instances (the Information Checker and the Answerer, see Section 2.4) to augment their knowledge and inform the final response generation.

2.4 Prompting Strategy

Interacting with the LLMs in this study involved a structured prompting methodology designed to ensure fair evaluation and mimic realistic, step-by-step problem-solving. We employed a zero-shot prompting approach, meaning no examples of correctly answered questions were included within the prompts themselves. Furthermore, to enhance accuracy and gain insight into the model's reasoning process, we generally encouraged the models to articulate their reasoning or provide a step-by-step derivation before presenting the final answer, aligning with Chain-of-Thought (CoT) principles [18]. For each distinct tax scenario taken from the VITA/TCE certification tests [8], a new, independent chat session was initiated with the LLM to prevent context carry-over. A key aspect of our methodology was the implementation of distinct multi-step workflows using separate LLM instances for different sub-tasks, both for establishing the baseline and evaluating the RAG-enabled system:

– **Baseline LLM Evaluation Workflow:** To assess the models' inherent capabilities without external knowledge, a two-step chain was employed:

    1. *Information Checker:* The first LLM instance received the full VITA scenario text and the associated question. Its task was to determine if any specific supplementary information mentioned within the scenario description (e.g., details from a hypothetical W-2 or Form 1099 described in the text) was necessary to answer the question.

2. *Answerer:* The second LLM instance received the original scenario and question, along with additional information depending on the decision from the Information Checker. Based solely on this input and its internal knowledge, it generated the final answer to the question.

– **RAG-Enabled LLM Evaluation Workflow:** To evaluate the impact of retrieval augmentation, a three-step chain incorporating the RAG system was used:

1. *Query Generator:* The first LLM instance received the VITA scenario and question. Its sole task was to generate a concise and relevant search query suitable for querying the RAG system's knowledge base (derived from Publication 4491).
2. *RAG Retrieval:* The generated query was used by the RAG system (detailed in Section 2.3) to retrieve the top 5 most relevant document chunks from Publication 4491.
3. *Information Checker (RAG-informed):* The second LLM instance received the original scenario, the question, *and the retrieved RAG document chunks*. Its task was identical to the baseline's Information Checker – determining if supplementary information from the scenario description was needed – but now informed by the relevant context from the IRS publication.
4. *Answerer (RAG-informed):* The third LLM instance received the original scenario, the question, the decision from the RAG-informed Information Checker, *and the same retrieved RAG document chunks*. Its task was to synthesize all this information, leveraging the provided authoritative context, to generate the final answer.

This structured, multi-step approach allows for a more controlled evaluation by breaking down the complex question-answering process and clearly delineating how external context from the RAG system is integrated into the workflow.

3 Results

This section presents the empirical evaluation of the selected Large Language Models (LLMs) on the VITA certification test questions. We structure the results into two main parts. First, Section 3.1 establishes the baseline performance of the models (GPT-4o mini, Llama 3.1 8B, Llama 3.1 70B, and Claude 3.5 Haiku) operating without any retrieval augmentation. Second, Section 3.2 details the performance of these same models when enhanced with our RAG system. This allows us to conduct a direct comparison to assess the impact of retrieval augmentation on accuracy across different question types and scenarios.

3.1 Baseline LLMs

Our exploration of the baseline capabilities of selected Large Language Models (LLMs) without Retrieval-Augmented Generation (RAG) encompasses OpenAI's GPT-4o mini, Meta's Llama 3.1 8B and Llama 3.1 70B, and Anthropic's Claude 3.5 Haiku. We evaluated these models across the basic and advanced scenarios derived from the original 130 VITA certification questions [8]. Performance was measured across true/false, multiple-choice, and open-ended questions. Detailed baseline results for each model are presented in Table 1. This analysis provides a crucial

Table 1: Baseline LLM Performance (Without RAG) - Accuracy % (Correct/Total)

Model	Scenario	T/F % (C/T)	MC % (C/T)	OE % (C/T)	Overall % (C/T)
GPT-4o mini	Basic	62.50 (20/32)	62.50 (15/24)	0.00 (0/4)	58.33 (35/60)
	Advanced	65.52 (19/29)	55.17 (16/29)	0.00 (0/12)	50.00 (35/70)
Llama 3.1 8B	Basic	71.88 (23/32)	25.00 (6/24)	0.00 (0/4)	48.33 (29/60)
	Advanced	58.62 (17/29)	44.83 (13/29)	0.00 (0/12)	42.86 (30/70)
Llama 3.1 70B	Basic	68.75 (22/32)	45.83 (11/24)	0.00 (0/4)	55.00 (33/60)
	Advanced	44.83 (13/29)	68.97 (20/29)	0.00 (0/12)	47.14 (33/70)
Claude 3.5 Haiku	Basic	53.12 (17/32)	41.67 (10/24)	0.00 (0/4)	45.00 (27/60)
	Advanced	62.07 (18/29)	65.52 (19/29)	0.00 (0/12)	52.86 (37/70)

reference point for understanding the models' inherent limitations, particularly in numerical reasoning, before introducing RAG.

3.1.1 Basic Scenarios

For the basic scenarios, the performance of baseline LLMs varied (see Table 1, Basic Scenario rows). For true/false (T/F) questions, accuracy peaked with Llama 3.1 8B at 71.88%, while Claude 3.5 Haiku had the lowest score at 53.12%. Performance on multiple-choice (MC) questions showed some divergence: GPT-4o mini led with 62.50% accuracy, whereas Llama 3.1 8B struggled considerably, achieving only 25.00%. Claude 3.5 Haiku (41.67%) and Llama 3.1 70B (45.83%) performed moderately on MC questions.

Open-ended questions requiring numerical answers proved insurmountable for all models in the baseline assessment. Accuracy was uniformly **0.00%** across all four models (GPT-4o mini, Llama 3.1 8B, Llama 3.1 70B, and Claude 3.5 Haiku). This universal failure highlights a critical inherent limitation in the precise numerical application of basic tax rules for these models operating without external knowledge augmentation.

Overall, in the basic scenario, GPT-4o mini showed the highest baseline accuracy (58.33%), while Claude 3.5 Haiku had the lowest (45.00%). The results clearly indicate that while some models handle basic T/F and MC questions reasonably well, the inability to perform numerical calculations presents a considerable gap.

3.1.2 Advanced Scenario

The advanced scenarios tested more nuanced tax comprehension. On true/false questions, GPT-4o mini performed best among the baseline models at 65.52%, followed closely by Claude 3.5 Haiku (62.07%). Llama 3.1 70B showed notably lower accuracy on T/F questions in this scenario (44.83%). Multiple-choice performance was led by Llama 3.1 70B (68.97%) and Claude 3.5 Haiku (65.52%), while Llama 3.1 8B again lagged (44.83%).

Consistent with the basic scenario, **open-ended questions remained impossible** for all baseline models, with every model scoring 0.00% accuracy on the 12 advanced open-ended questions. This reinforces the observation that even with more complex scenarios, the fundamental difficulty in precise numerical calculation and rule application persists for these models in their baseline state.

Table 2: RAG-Enabled LLM Performance - Accuracy % (Correct/Total)

Model	Scenario	T/F % (C/T)	MC % (C/T)	OE % (C/T)	Overall % (C/T)
GPT-4o mini	Basic	68.75 (22/32)	83.33 (20/24)	25.00 (1/4)	71.67 (43/60)
	Advanced	72.41 (21/29)	65.52 (19/29)	16.67 (2/12)	60.00 (42/70)
Llama 3.1 8B	Basic	75.00 (24/32)	41.67 (10/24)	0.00 (0/4)	56.67 (34/60)
	Advanced	75.86 (22/29)	55.17 (16/29)	0.00 (0/12)	54.29 (38/70)
Llama 3.1 70B	Basic	75.00 (24/32)	62.50 (15/24)	25.00 (1/4)	66.67 (40/60)
	Advanced	75.86 (22/29)	68.97 (20/29)	8.33 (1/12)	61.43 (43/70)
Claude 3.5 Haiku	Basic	78.12 (25/32)	58.33 (14/24)	0.00 (0/4)	65.00 (39/60)
	Advanced	68.97 (20/29)	68.97 (20/29)	16.67 (2/12)	60.00 (42/70)

Overall accuracy in the advanced scenario was led by Claude 3.5 Haiku (52.86%), followed by GPT-4o mini (50.00%). Llama 3.1 8B had the lowest overall accuracy at 42.86%. In essence, the results confirm that the baseline models can handle some inferential reasoning for T/F and MC questions, but they failed on open-ended numerical tasks. This result underscores a critical performance gap that RAG systems aim to address.

3.2 RAG-Based LLMs

This section evaluates the performance of the same LLMs (GPT-4o mini, Llama 3.1 8B, Llama 3.1 70B, and Claude 3.5 Haiku) when augmented with the RAG system described in Section 2.3. The objective is to assess the extent to which providing relevant context from IRS Publication 4491 improves accuracy compared to the baseline results presented in Section 3.1. Detailed results for the RAG-enabled models are presented in Table 2, and comparative visualizations are shown in Figures 1 and 2.

3.2.1 Impact on Overall Accuracy

As illustrated in Figure 1, the introduction of the RAG system generally led to notable improvements in overall accuracy across most models and scenarios compared to the baseline (Table 1). For instance, GPT-4o mini's overall accuracy in the Basic scenario increased substantially from 58.33% (Baseline) to 71.67% (RAG), representing a gain of over 13 percentage points. Similarly, Llama 3.1 70B's overall accuracy improved from 55.00% to 66.67% in the Basic scenario and from 47.14% to 61.43% in the Advanced scenario. Llama 3.1 8B also saw consistent gains, moving from 48.33% to 56.67% (Basic) and 42.86% to 54.29% (Advanced). Claude 3.5 Haiku showed a great improvement in the Basic scenario (45.00% to 65.00%) and also improved in the Advanced scenario based on the updated results (52.86% to 60.00%). These results suggest that providing targeted context via RAG is broadly beneficial for enhancing the general question-answering capability of these LLMs in the tax domain.

3.2.2 Impact by Question Type

Figure 2 provides a more granular view, revealing how RAG affected performance on different types of questions.

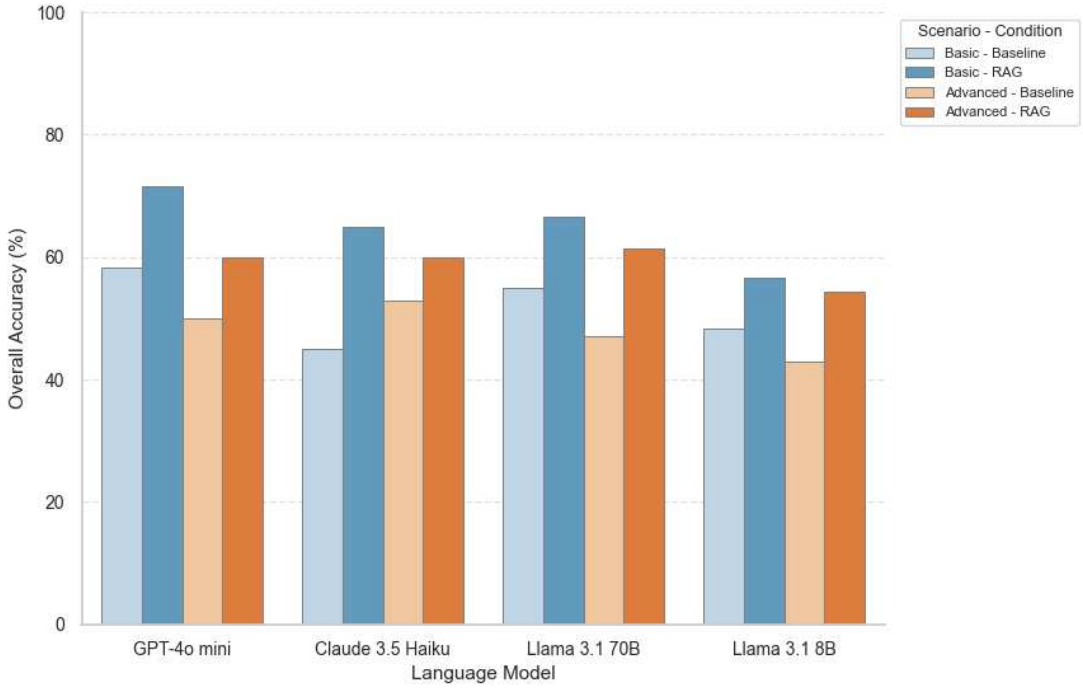

Figure 1: Overall Accuracy Comparison: Baseline vs. RAG-Enabled LLMs across Basic and Advanced Scenarios.

- **True/False (T/F):** RAG generally improved or maintained T/F accuracy. For example, Claude 3.5 Haiku's Basic T/F score jumped significantly from 53.12% to 78.12%. Llama 3.1 70B also saw a large gain in the Advanced scenario (44.83% to 75.86%). Other gains were more modest, suggesting the baseline models already had some capability here, but RAG helped solidify or slightly enhance it.
- **Multiple Choice (MC):** MC questions also saw benefits from RAG. The most dramatic improvement was for GPT-4o mini in the Basic scenario, rising from 62.50% to 83.33%. Llama 3.1 70B (Basic: 45.83% to 62.50%) and Llama 3.1 8B (Advanced: 44.83% to 55.17%) also showed noticeable gains.
- **Open Ended (OE):** The impact of RAG was most striking, albeit from a low starting point, on open-ended questions requiring numerical calculation. Baseline models uniformly scored 0% on all OE questions. With RAG, GPT-4o mini (Basic: 25.00%, Advanced: 16.67%), Llama 3.1 70B (Basic: 25.00%, Advanced: 8.33%), and Claude 3.5 Haiku (Advanced: 16.67%) were able to answer *some* OE questions correctly. This demonstrates that RAG can make these challenging numerical reasoning tasks tractable, even if absolute accuracy remains low. However, challenges persist: Llama 3.1 8B remained unable to answer any OE questions correctly even with RAG, and Claude 3.5 Haiku also failed the Basic OE questions despite the RAG context.

In summary, the RAG system demonstrated a clear positive impact, enhancing accuracy across T/F and MC questions and, crucially, enabling the models to achieve non-zero accu-

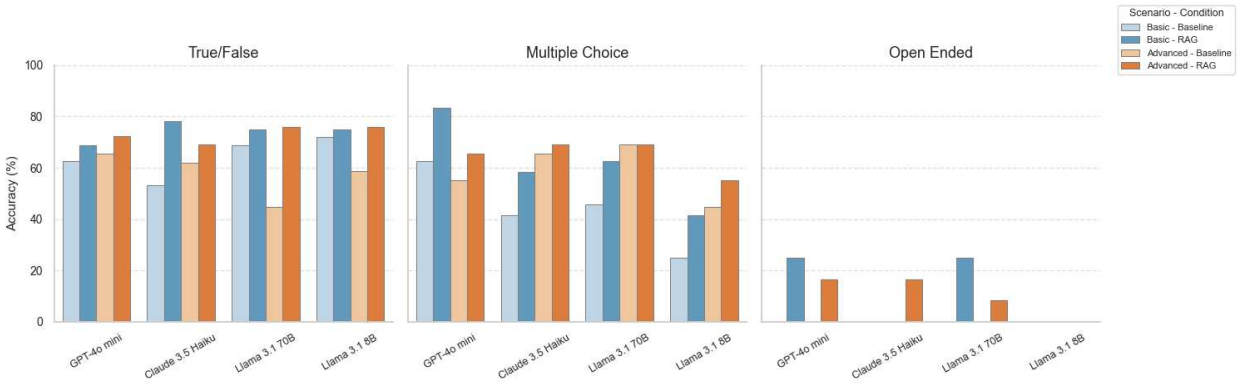

Figure 2: Accuracy Breakdown by Question Type (True/False, Multiple Choice, Open Ended): Baseline vs. RAG-Enabled LLMs.

racy on previously insurmountable open-ended numerical questions for most models. While OE performance is still far from perfect, RAG represents a much needed step towards addressing this key limitation identified in the baseline evaluation.

4 Discussion

Our findings demonstrate that incorporating Retrieval-Augmented Generation (RAG) enhances the ability of smaller LLMs to answer VITA tax questions compared to their baseline performance (Tables 1 and 2). The most crucial improvement was enabling models like GPT-4o mini and Llama 3.1 70B to achieve non-zero accuracy on open-ended numerical questions, a task impossible for all baseline models (Figure 2). RAG also yielded general improvements in overall accuracy (Figure 1), driven by better performance on True/False and Multiple Choice questions, suggesting the provided context aids both information recall and reasoning.

However, substantial challenges remain. Even with RAG, accuracy on open-ended questions remains low, and some models (Llama 3.1 8B) failed to show any improvement on this front, indicating that contextual information alone doesn't fully solve the inherent numerical reasoning limitations of these LLMs. This highlights the importance of the base model's capabilities.

Our RAG system design involved specific choices balancing effectiveness and feasibility. We used IRS Publication 4491 [7] as a focused knowledge source, employed heading-based chunking for semantic coherence, and implemented a two-stage retrieval process for precision. These choices, while effective, introduce limitations. The system's knowledge is confined to the scope of Pub 4491, potentially failing on out-of-scope topics. Furthermore, the chunking strategy can create variable lengths, and the two-stage retrieval adds latency.

Looking ahead, overcoming the remaining limitations, particularly in complex numerical reasoning, requires further innovation. We plan to investigate methods beyond standard RAG to improve the framework's performance, focusing on grounding responses not just in reliable sources but also in verifiable calculations. **Indeed, our preliminary experiments integrating this RAG system with dedicated small reasoning models within a multi-agent framework have shown great promise, achieving overall accuracies exceeding 95% on the VITA test dataset.** By expand-

ing on such hybrid frameworks, combining reliable RAG for factual grounding with specialized components for calculation and reasoning, we hope to create an AI assistant that can answer tax-related questions with significantly higher reliability and accuracy.

5 Conclusion

This paper investigated the potential of Retrieval-Augmented Generation (RAG) to enhance the capabilities of smaller, accessible Large Language Models (LLMs) for assisting Volunteer Income Tax Assistance (VITA) volunteers. By evaluating models like GPT-4o mini, Llama 3.1 8B/70B, and Claude 3.5 Haiku on standardized VITA certification test questions, we established baseline performance metrics and subsequently measured the impact of our RAG system, which leverages IRS Publication 4491 as a knowledge source.

Our findings clearly demonstrate that RAG improves the accuracy of these LLMs in the tax domain compared to their standalone performance. The augmented models showed enhanced capabilities across true/false and multiple-choice questions. Most critically, RAG enabled several models to achieve non-zero accuracy on open-ended numerical questions, a task where all baseline models failed completely, highlighting RAG's potential to address the core challenge of numerical reasoning in tax law by providing relevant context. However, performance on these numerical tasks remains limited, indicating that context alone is not a complete solution.

Looking ahead, our future work aims to build upon this framework to create a more robust and reliable AI tax assistant. We plan to explore the integration of more sophisticated reasoning models and investigate alternative methods beyond RAG to further improve performance. A key focus will be on enhancing the grounding of the model's responses, not only in reliable RAG-retrieved sources but also through verifiable and reliable calculations. We will develop mechanisms where the AI assistant can potentially utilize or generate relevant calculation code to produce numerical responses with higher fidelity. By combining reliable RAG for factual grounding with trustworthy calculation capabilities, we hope to create an advanced AI assistant capable of answering complex tax-related questions with much greater reliability and accuracy, ultimately aiding volunteers to assist low/moderate income taxpayers.

Acknowledgments. The authors thank Nina E. Olson, the Executive Director of the Center for Taxpayer Right, for the consultant and guidance on this project. This project has been partially supported by the NSF DASS program under grants CCF-2317206, CCF-2317207, and CCF-2532965.

References

[1] Embed intelligent tax software. https://www.getapril.com, 2023. Online.

[2] Gary Drenik. Ai-powered tax system is creating a new paradigm: Will banks and fintechs adopt the technology to help their customers save on their tax bill?, February 2023. Forbes, Accessed: 2025-04-18.

[3] Sina Gogani-Khiabani, Varsha Dewangan, Nina Olson, Ashutosh Trivedi, and Saeid Tizpaz-Niari. Technical challenges in maintaining tax prep software with large language models, 202. https://arxiv.org/abs/2504.18693.

[4] Sina Gogani-Khiabani, Ashutosh Trivedi, ShinPing Chyi, and Saeid Tizpaz-Niari. Performance of LLMs on VITA test: Potential for AI-assisted tax returns for low income taxpayers. *Artificial Intelligence and Law*, 2025. Accepted for publication. Forthcoming.

[5] Jeff Huber and Chroma Core Developers. Chroma: The ai-native open-source embedding database. Apache 2.0 License. Accessed 2025-05-12.

[6] HuggingFace. Massive text embedding benchmark (mteb) leaderboard. https://huggingface.co/spaces/mteb/leaderboard, Accessed 2025-05-12. Check leaderboard for latest information and potential citation guidance if available.

[7] Internal Revenue Service. Publication 4491, vita/tce volunteer assistor's guide. Tax Year 2022 revision, 2022.

[8] Internal Revenue Service. *VITA/TCE Volunteer Assistor's Test/Retest*, 2022.

[9] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics, 2020.

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Myle Ott, Wen-tau Chen, Alexis Conneau, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[11] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine, 2023.

[12] Gabriel Rosa, Luiz Bonifacio, Victor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Rodrigo Lotufo, and Rodrigo Nogueira. In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121*, 2022.

[13] Dananjay Srinivas, Rohan Das, Saeid Tizpaz-Niari, Ashutosh Trivedi, and Maria Leonor Pacheco. On the potential and limitations of few-shot in-context learning to generate metamorphic specifications for tax preparation software. In Daniel Preoțiuc-Pietro, Catalina Goanta, Ilias Chalkidis, Leslie Barrett, Gerasimos Spanakis, and Nikolaos Aletras, editors, *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 230–243, Singapore, December 2023. Association for Computational Linguistics.

[14] Saeid Tizpaz-Niari, Shiva Darian, and Ashutosh Trivedi. Metamorphic debugging for accountable software, 2024.

[15] Saeid Tizpaz-Niari, Verya Monjezi, Morgan Wagner, Shiva Darian, Krystia Reed, and Ashutosh Trivedi. Metamorphic testing and debugging of tax preparation software. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, pages 138–149, 2023.

[16] Sentence Transformers. Cross-encoder for ms marco passage ranking. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2, Accessed 2025-05-12. Model card for cross-encoder/ms-marco-MiniLM-L-6-v2.

[17] Taylor Webb, Shanka Subhra Mondal, Chi Wang, Brian Krabach, and Ida Momennejad. A prefrontal cortex-inspired architecture for planning in large language models, 2024.

[18] 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.

[19] Jay Yu, Kevin McCluskey, and Saikat Mukherjee. Tax knowledge graph for a smarter and more personalized turbotax, 2020.