Shaping the Future of Tax Advisory Using Artificial Intelligence

Conor Kelly

Postgraduate, Department of Engineering, School of Management
University College London, London, United Kingdom

Abstract: ***This research harnessed the capabilities of GPT-4 using Langchain and Vector Search to develop an automated Tax Advisory Service, termed TaxBot. In light of recent advancements in Large Language Models, the project aimed to automate facets of the tax advisory domain, an industry traditionally reliant on skilled labour. TaxBot exhibited notable accuracy, achieving 70% on Charted Accountant Proficiency Level 2 Examination questions, and even up to 80% in specific tax areas like VAT. Efforts were made to enhance its reliability by labelling chunk embeddings, although challenges related to consistency persist. The project underscores the potential of AI in reshaping the tax advisory landscape, indicating a promising trajectory for the integration of AI technologies in professional advisory services. Future work will focus on refining the system for real-world commercial applications.***

Keywords: Artificial Intelligence in Tax Advisory, GPT-4 Application, Retrieval Augmented Generation, Vector Search, Large Language Models, Automated Tax Services, Accountant Proficiency Examination, Value-Added Tax (VAT) Automation, Chunk Embeddings in AI

1. Introduction

The rapid evolution of artificial intelligence (AI) and the advent of advanced models like GPT-4 have opened new vistas of opportunity across diverse industries. Tax advisory, a domain that has historically relied on skilled labour and specialised knowledge, stands poised to be disrupted in this new wave of technological transformation.

Within this paper, we embark on an exploration of the potential of harnessing these AI advancements by using GPT-4 and Retrieval Augmented Generation to automate areas of Tax Advisory using Irish Tax Law. This system is dubbed TaxBot.

Assessing the commercial application of this system, we look at how similar technology stacks are currently being used to create value today and identify four core objectives for the project, which aim for the system to be accurate, reliable, truthful and user-friendly.

The literature review offers a foundational understanding. We delve into the intricacies of GPT-4, unpack the nuances of Vector Databases, explore the workings of Langchain, and survey the alternative methods that exist in the contemporary AI landscape.

Digging deeper into how TaxBot would be brought to market, we take a snapshot of the Irish Tax Advisory Industry and discuss the business model for real world application, in which the case for a business to business distribution is deemed preferable.

Our methodology, dissects the technical backbone of TaxBot. From system architecture to data handling, model selection, and the intricacies of evaluation. This provides a granular understanding of the underpinnings of the system.

Project experimentations include prompt engineering, narrow versus broad tax advisory systems and elucidate the strategies behind labelling chunks for improved model reliability.

Our findings offer an evaluative perspective on TaxBot's capabilities. The next steps sketch a trajectory for the future, shedding light on potential expansions and real world application.

In conclusion, this paper aims to be a comprehensive examination of the intersection between state-of-the-art AI models and the tax advisory domain to make a compelling glimpse into the future of tax advisory using artificial intelligence.

1.1 Objectives

The objective of this project is to use Retrieval Augmented Generation (RAG) with high preforming Large Language Models to replicate the capabilities of a Tax Advisor which specialises in Irish Tax Law.

The requirements necessary to build this system closely resemble those required to bring it to market. Therefore, the core objectives of the product serve a simultaneous purpose; build a system that replicates the capabilities of an Irish Tax Advisor and make it fit for commercial purposes.

The core objectives established to attain this are as follows:

1) Accuracy

The system's responses must match or surpass the accuracy of a professional tax advisor. This entails performing thorough analyses based on a user's context, ensuring the system comprehends the query, and subsequently retrieving, interpreting, and delivering precise, beneficial, and actionable information.

2) Reliability

The system must consistently offer correct information relevant to the user's context. For commercial adoption, the system must maintain a high response rate with minimal

Paper ID: ES24115141827 DOI: https://dx.doi.org/10.21275/ES24115141827

errors. Consistency in both the structure and content of the output is the crucial metric for this.

3) Truthful

Interactions with the system must be transparent. In the realm of Tax Advisory, dishonesty equates to negligence. Since LLMs can sometimes generate fabricated data, a major project goal is to limit this risk.

4) User Friendly

The system should mimic the approachable demeanour of a human tax advisor. Chatbots are an ideal foundation, given their Q&A format. Enhancing this foundation, the system should elucidate its findings in a way that's comprehensible to all users, irrespective of their legal expertise or command of language.

2. Literature Review

2.1 Overview

In Nakajima's 'Task-driven Autonomous Agent Utilizing GPT-4, Pinecone, and LangChain for Diverse Applications', a novel task-driven autonomous agent that leverages OpenAI's GPT-4 language model, Pinecone vector search, and the LangChain framework to perform a wide range of tasks across diverse domains is proposed (Nakajima, 2023). The main objectives are for the system to complete tasks, generate new tasks and prioritize tasks (Nakajima, 2023).

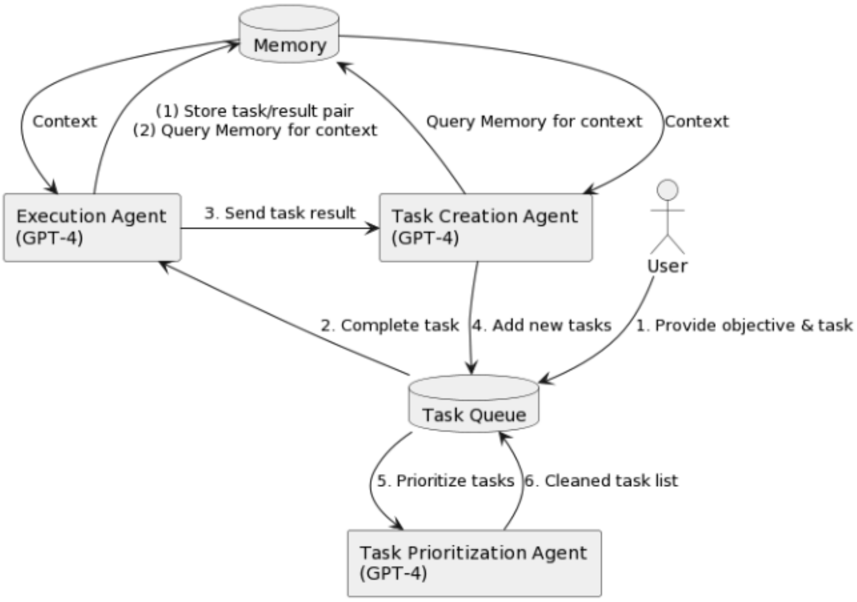


Figure 1: Task Driven Autonomous Agent Workflow (Source: Yohei Nakajima)

This approach has inspired the basis of Taxbot. We will look specifically at one objective of the system, which is its ability to complete tasks.

We will assess its core components GPT-4, vector databases and the LLM framework LangChain which the goal of single task completion (eg: Ask a question and get an answer).

The methodology acts a deep dive into the technology required to build a system that can support TaxBot's objectives.

3. Methodology

3.1 System Structure

The structure of this project is partly inspired by Nakajima's "Task-driven Autonomous Agent Utilizing GPT-4, Pinecone, and LangChain for Diverse Applications," in which a retrieval system is created that relies on direction provided by GPT-4. In this project, GPT-4 plays three distinct agent roles: Execution, Task Creation, and Task Prioritisation, which combine to create a loop of decision-making and execution, allowing it to act 'autonomously.'

The shortcoming of this model is that the error rate of GPT-4 (which varies depending on the task) compounds with the number of execution cycles that it performs. For example, if the Execution Model hallucinates, this will pass false information to the Task Creation and Prioritisation Agents, who will act on this information as they possess no ability to verify its truthfulness.

Taxbot aims to minimize returning false information. Thus, it follows the base structure of Nakajima's project while leaving out components prone to error.

Namely, interlinking GPT-4 agents are not included in the structure. Instead, there are two independent directional operations performed by GPT-4. One to direct retrieval and one to respond to query.

For TaxBot, the retrieval structure remains in place, and the vector database remains the core of the system functionality.

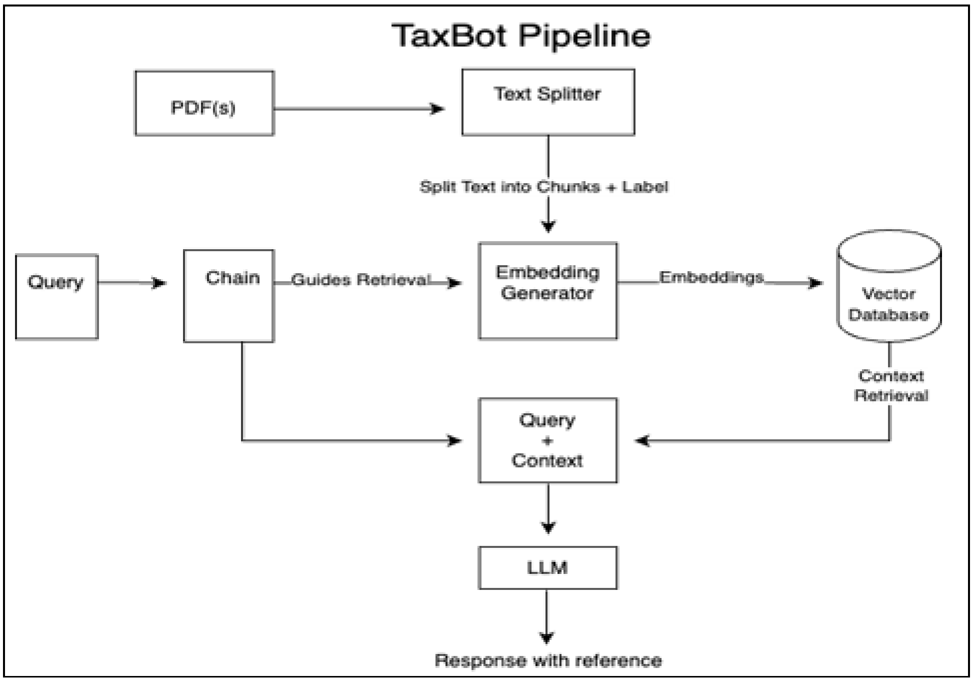


Figure 2: TaxBot Pipeline

3.2 Data Inputs

We will be using Chartered Accountants Ireland as a resource for both tax legislation data and examination questions/answers for evaluations (CAI, 2023).

To keep members up to date with legislation, CAI maintains a directory of 'Key Irish Tax Acts' and any amendments that have been made to them. This provides a comprehensive data source for tax legislation that is relevant to tax professionals and fit for practice (CAI, 2023).

The following acts have been scraped for data inputs:

- Taxes Consolidation Act, 1997 (as amended up to and including Finance Act 2022)
- Stamp Duties Consolidation Act, 1999 (as amended up to and including Finance Act 2022)
- Capital Acquisitions Tax Consolidation Act 2003 (as amended up to and including Finance Act 2022)
- Value-Added Tax Consolidation Act 2010 (as amended up to and including Finance Act 2022)
- Local Property Tax Act 2012 (as amended up to and including Finance Tax Appeals Act 2021)

3.3 Data Cleaning & Pre-Processing

To effectively store the scraped data, HTML characters were removed, such as inequality signs used for tags, and the remaining text was labeled using the name of the legislation, which was derived from and stored on a pdf.

These files were then pre-processed by removing metadata, creating chunks, and labeling each chunk based on its source. This is discussed in further detail under 'Experimentation.'

The Recursive Text Splitter module was applied to chunk the data. This is a Langchain module that splits documents recursively by different characters - starting with "\n\n," then "\n," then. " Recursive splitting involves breaking down a text into smaller pieces, often sentences or words, and then further breaking down those pieces if necessary. This keeps semantically relevant content in place while removing irrelevant text from the document (Langchain, 2023).

Parameters passed through this module are chunkSize and chunkOverlap. These determine the number of characters in each split and the crossover of characters between each split.

This presents a weakness in the system as these numbers are arbitrarily chosen, creating a likelihood that the document will be split in places where semantic context is necessary. For Taxbot, a chunk size of 3000 and an overlap of 100 was selected.

As this was arbitrarily chosen, there's more room for experimentation to see how this improves the model's accuracy.

3.4 Operations

A Langchain module named RetrievalQA is the backbone of the system's operations,

This provides a fundamental capability to the system so that it can retrieve relevant documentation from the vector database when the user provides input and provides it to the LLM as context to generate output. This process is referred to as "Retrieval Augmented Generation" (Langchain, 2023).

In this case, retrieval is done using semantic search, which vectorizes the incoming queries and then retrieves the most similar vector embeddings from the database to match with the query for the LLM to generate output.

RetrievalQA passes three parameters: LLM, Chain (chain_type), and Retriever.

LLM: This point to the model we're using to both guide the retriever and generate output. In this case, GPT-4.

Chain: This refers to the method of retrieval Langchain will use. In this case, it is 'Stuff'. This chain takes a list of documents, inserts them all into a prompt, and passes that prompt to the LLM.

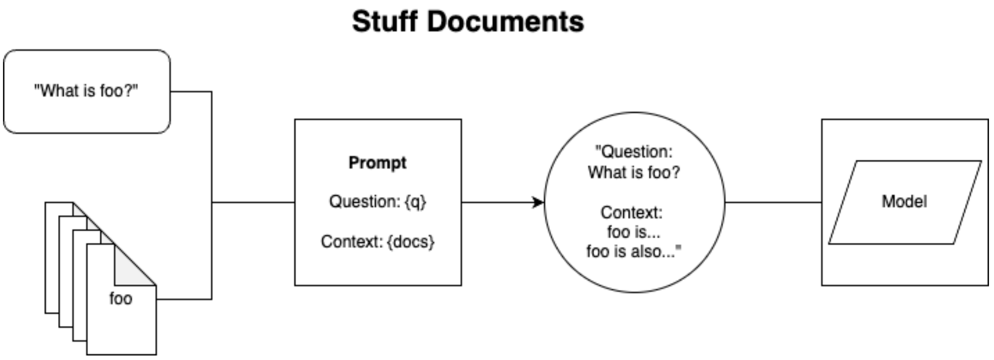


Figure 3: 'Stuff' Chai (Source: Langchain)

Retriever: This points to the database where information is to be retrieved from. Which, in this case, is a vector database.

RetrievalQA is the component of Langchain, which provides the infrastructure for the system's operations. By passing parameters that determine the model and chain type, there is some flexibility to tweak components in the chain to optimize for performance. It is also simple to redirect the database as it is an independent component. These flexible characteristics, along with a rigid structure, make Langchain an ideal framework for building the system.

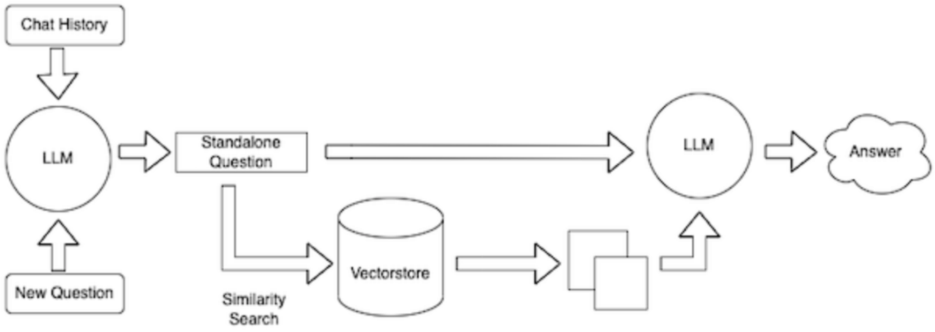


Figure 4: Diagram of Retrieval Step (Source: Langchain)

3.5 Database

ChromaDB provides a more suitable database structure than Pinecone for this project as it is open-source and provides temporary local ephemeral storage, which is ideal for experimentation and iterating (Campos, 2023).

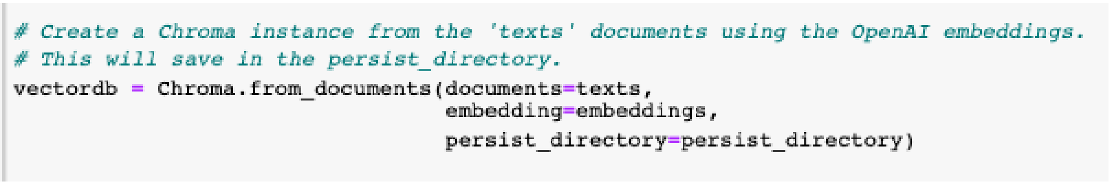
However, to launch a full-scale application, Pinecone would be a better alternative as it provides cloud-based database persistence with no limitation on storage (Campos, 2023).

ChromaDB takes three parameters: documents, embedding, and persistent directory.

Documents: Points to the pre-processed documentation prepared for vectorization.

Embedding: The embedding model used on the documentation.

Persist Directory: Local storage for the vector embeddings.



```
# Create a Chroma instance from the 'texts' documents using the OpenAI embeddings.
# This will save in the persist_directory.
vectordb = Chroma.from_documents(documents=texts,
                                 embedding=embeddings,
                                 persist_directory=persist_directory)
```

Figure 5: Passing Parameters Through ChromaDB

Passing these parameters gives the developer control over key elements of the project, such as storage and the embedding model. The straightforward structure also makes it ideal for experimentation and iteration.

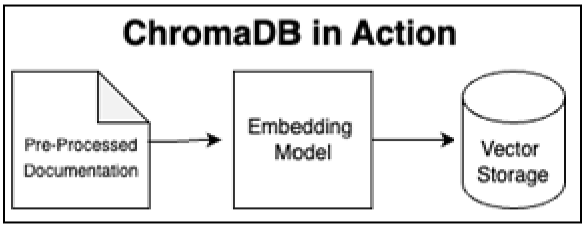


Figure 6: ChromaDB in Action

3.6 Model Selection

Embedding Model:
Embeddings are numerical representations of concepts converted to number sequences, which make it easy for computers to understand the relationships between those concepts (Greene et al., 2022).

Microsoft (2023) mentions how typically embedding functions are based on methods such as machine learning models, word embeddings, and feature extraction algorithms. However, in the case of applications using advanced LLMs, embeddings are typically generated using other LLMs that have a large number of attributes or features and vectorize in highly highly dimensional space (Schwaber-Cohen, 2023).

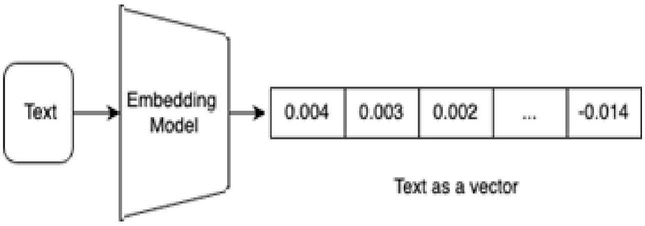


Figure 7: Embedding Model in Action (Source: OpenAI)

As this project is performing text similarity search. There are a number of models available which can support the required embeddings.

Factors such as context window, dimensional space cost, and efficiency should be considered for suitability to compare models.

There are open-source embedding models such as all-MiniLM-L6-v1, SPLADEv2, or multi-qa-mpnet-base-dot-v1, which can provide a low-cost, efficient and performant solution for embeddings (Reimers, 2021).

The default model being used is OpenAI's text-embedding-ada-002. This model has a wide range of applications (suitable for text-similarity, text-search-query, text-search-doc, code-search-text, and code-search-code), allowing it to be performant on a diverse set of text input (Greene et al., 2022).

It has a context window of 8192 tokens (6000 words) and embeddings of 1536 dimensions (Greene et al, 2022). This makes it both suitable for large documents and efficient.

The cost of running text-embedding-ada-002 is $0.0001 per 1,000 tokens (Wiggers, 2023), which makes the model ideal for experimentation and iteration.

Model	Performance
Text-embedding-ada-002	81.5
Text-similarity-davinci-001	80.3
Text-similarity-curie-001	80.1
Text-similarity-babbage-001	80.1
Text-similarity-ada-001	79.8

Figure 8: OpenAI Embedding Model Comparisons for Sentence Similarity (Source: OpenAI)

Based on context window, efficiency, and cost, text-embedding-ada-002 makes an ideal first candidate for embedding legal tax documentation. However, there is room for further exploration, substituting this model for open-source alternatives.

Large Language Model:
There are a number of LLMs available to use for this task. Open-source models and providers such as Claude, Anthropic, and Cohere have high performant models available which are suitable for this task. However, there are a number of reasons why GPT-4 is preferable for this task.

3.7 Technical Structure

Following the system structure, the components of TaxBot can be seen in Fig 31. These were selected as the most advanced components available, which had project suitability. The operations are supported by Langchain, the modeling by OpenAI, and the database by ChromaDB. These give the strongest likelihood of performance across Accuracy, Consistency, and Reliability.

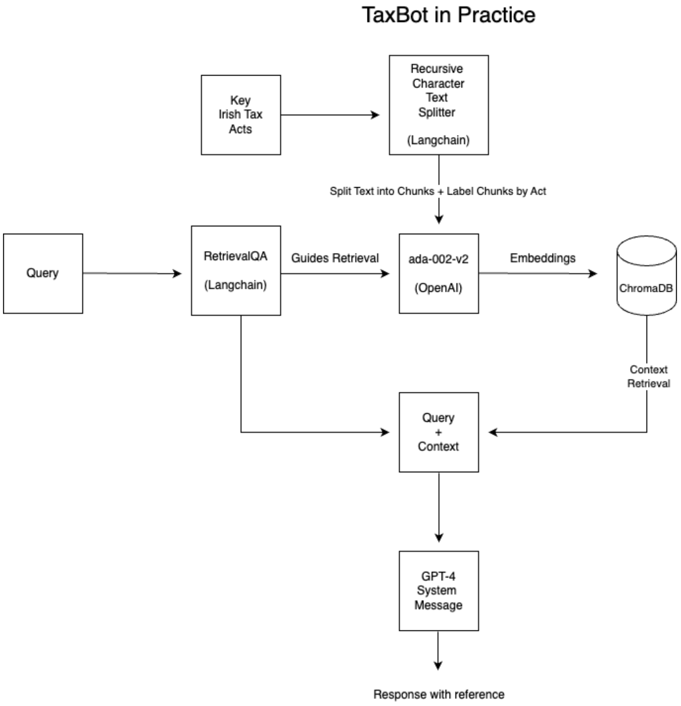


Figure 9: TaxBot in Practice

4. Evaluation

4.1 Evaluation Criteria

The objective of the project is to create a Legal Tax Advisor that is Accurate, Reliable, Truthful, and User Friendly.

These will be the core criteria governing our evaluation, with separate weighting attached to each.

Every response from the model will be scored out of 6. The breakdown will be as follows:

1) Accuracy

Accuracy is measured based on whether the answer provided by the model is correct according to the examiner's solutions.

This is the most important criterion as it determines the viability of the product. It will be marked as follows:

Incorrect	30-59%	60-90%	Correct
0	1	2	3

Hallucinations: In the context of LLMs, “hallucination” refers to a phenomenon where the model generates text that is incorrect, nonsensical, or not real. From a high level, hallucination is caused by limited contextual understanding since the model is obligated to transform the prompt and the training data into an abstraction, in which some information may be lost. Moreover, noise in the training data may also provide a skewed statistical pattern that leads the model to respond in a way you do not expect (Tam, 2023).

To minimize the tendency for the model to hallucinate, each output will be cross-referenced with the legal text from which the answer has been derived.

This is made possible by labeling the chunks pre-embedding so that when retrieved, the model has context from where they came from. The model is then prompted to reference where the data has come from in the output.

This adds a layer of security that minimizes hallucinations and gives additional confidence to the user. Further discussion on this is in the Experimentation section.

2) Contextual Understanding

Contextual understanding is what gives the system human-like capabilities. It allows the advisor to be flexible for each case that it deals with.

This will be assessed based on the evidence that the model understood the nuance of the question being asked. This is displayed in its answers. It will be marked as follows:

Context will be measured by providing the bot with information about the user and asking it questions, which rely on this information to provide a result. The answers will be marked as:

Misunderstood	Partially Understood (50%+)	Understood
0	0.5	1

3) Explainability

Legibility can be described as the model's ability to explain the answers it has provided. This is a crucial part of being an advisor, as the technical knowledge of your client base varies.

Evaluation here is subjective, but it is assessed on the conciseness, readability, and interpretability of responses from the model. The model is prompted to provide an answer as well as an answer that “can be understood by a person who is unfamiliar with legal terminology.” It will be marked as follows:

Unclear	Reasonably Explained	Very Well Explained
0	1	2

4) Consistency

Given the non-deterministic nature of LLMs, maintaining output consistency can be a challenge. To check for the consistency of our model, we will run the same query 3 times.

Consistency is assessed based on how frequently the same answer is given from the model. It is marked as follows:

Once	Twice	Three Times
0	1	2

The most common answer is the answer that’s marked in 1, 2 & 3.

5. Experimentation

5.1 Labelling Chunks

Labeling chunks is a novel technique that could be a promising method for Retrieval systems to help minimize hallucinations.

The procedure undertaken to label the chunks in this system can be seen in Fig 41.

Paper ID: ES24115141827 DOI: https://dx.doi.org/10.21275/ES24115141827

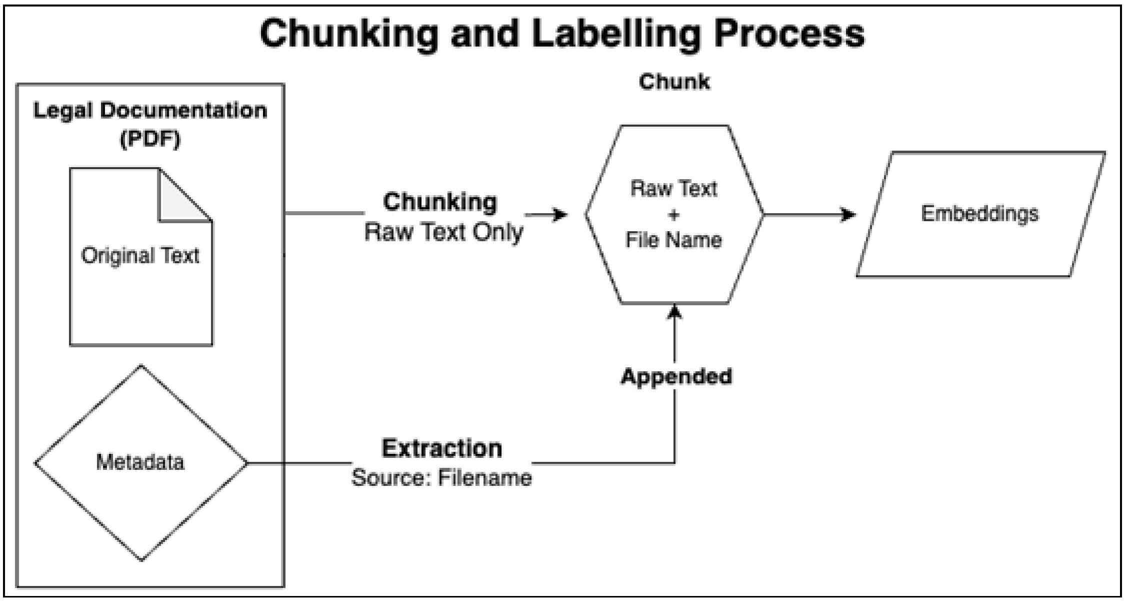


Figure 10: Chunking & Labelling Process

In Fig 41, the Legal Documentation is a pdf file that would contain the legislation for a specific area of taxation. As the name of this file was stored in the metadata, it is possible to use this as a reference point for each chunk.

However, the metadata is repetitive and contains no relevant information for the task other than the source filename. Thus, cleaning it from the chunks pre-embedding is important to optimize for semantic relevance. Therefore, the best approach was to extract the source from the metadata as a string and append it to each text chunk from the file. This way each chunk as a reference point which the model can access.

This created a backbone in the output from the model, as it could use the filename to reference where the data was coming from, allowing the end user to check the reference if necessary. It also added a layer of consistency, addressing the hallucination problem in a realistic and effective manner. There were no noticeable issues caused by this in the accuracy of the retrieval.

6. Results

6.1 Summary

Overall, TaxBot’s performance was highly impactful, scoring an incredible average of 70% on Accuracy. For context, the pass mark in the CAP2 Exams is 50%.

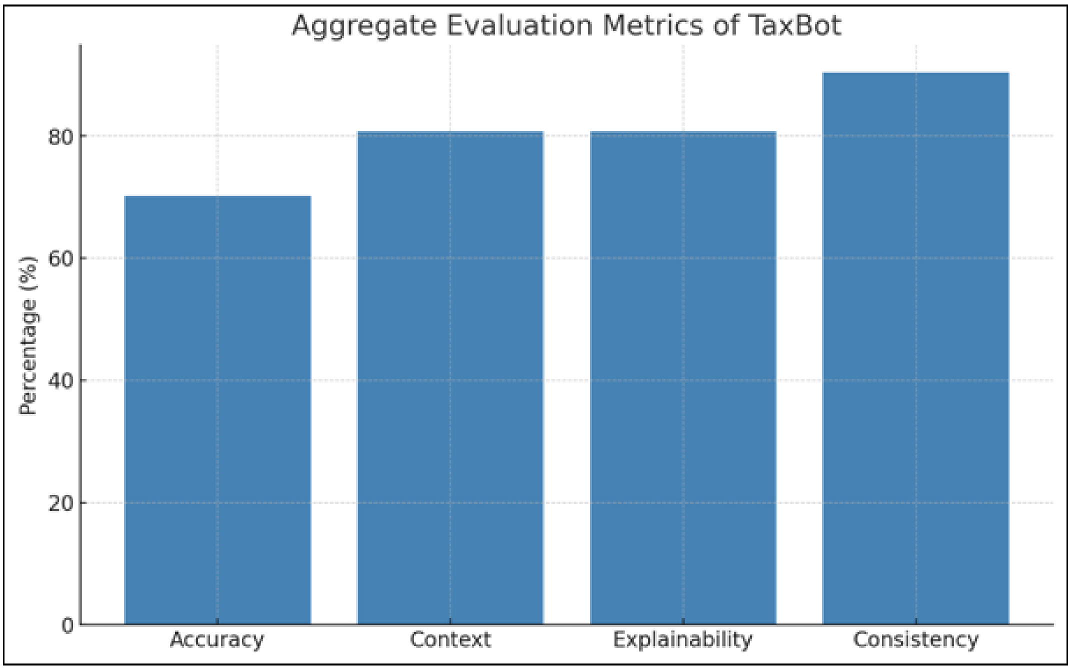


Figure 11: Aggregate Evaluation Metrics of TaxBot

The results varied based on the tax area, however performance was strong across Accuracy, Context and Explainability on each area apart from Corporate Tax. Capital Gains Tax performed the best overall.

Digging deeper into each section, we find that Accuracy was consistently high among all areas except for Corporate Tax. This is likely due to the nature of the Corporate Tax Questions being outside of the scope of the legislation provided to the model, indicated by the fact that GPT-4 used retrieved information in just 50% of its answers. Most of which were incorrect. This performance could likely be improved by applying the correct legislation.

Excluding Corporate Tax, this puts average accuracy on the evaluation set at 82.42%. An exceptionally high rate which indicates promising capabilities for commercial use.

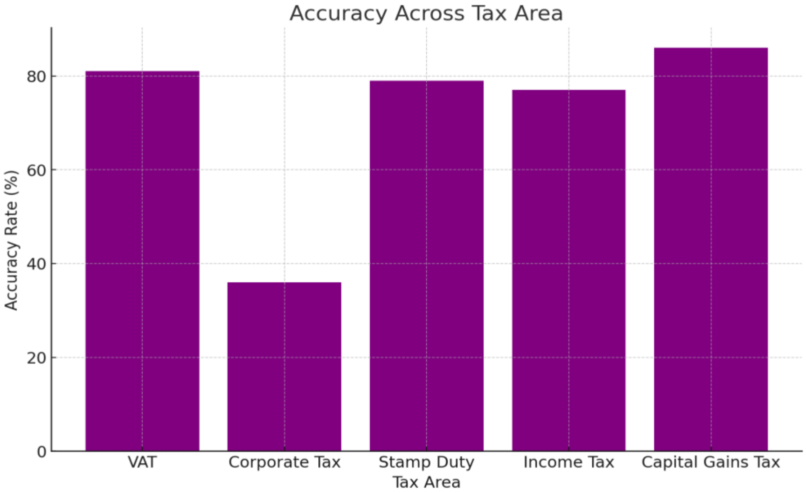


Figure 12: Accuracy Across Tax Area

The system’s capabilities around Context, Explainability and Consistency performed at an impressively high level. The relatively low consistency on Capital Gains Tax (67%), despite its high performance, is likely an area for re-evaluation.

The model’s ability to understand context is consistently performant, which is expected from GPT-4. Explainability is also strong.

The total performance is based on a score of 6 from each question, which is broken down by; Accuracy (3), Context (1) and Explainability (2).

On this basis we can see that VAT and Capital Gains Tax perform the highest overall.

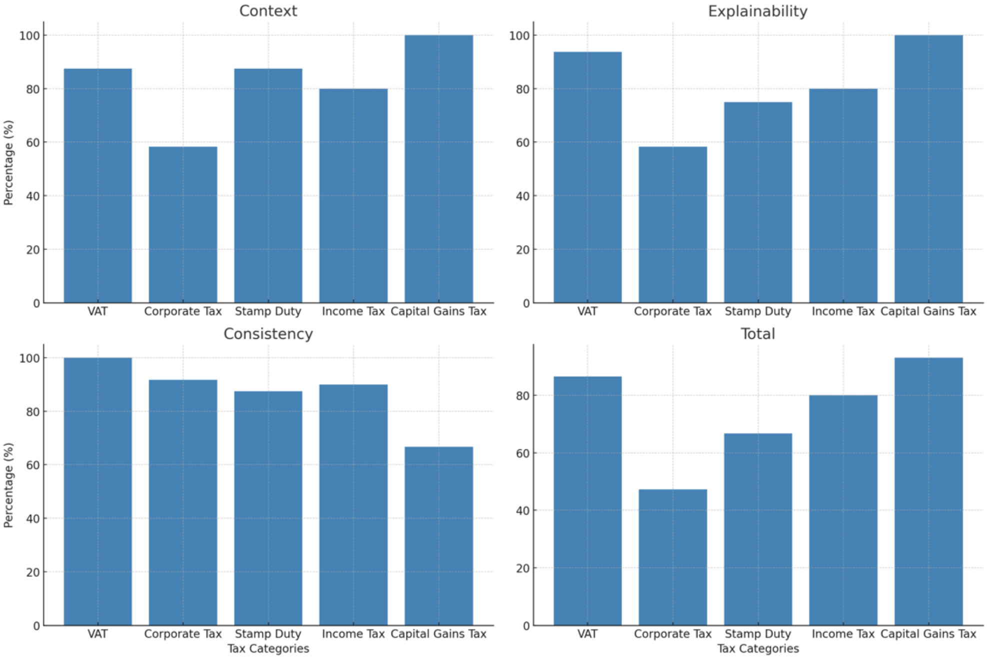


Figure 13: Comparison of Performance in Context, Explainability, Consistency and Total Across Tax Areas

6.2 Project Evaluation

To commercialize Taxbot, reliability is imperative. In a professional setting, users need to have the assurance that the information they're receiving is accurate and that they can consistently rely on the system to deliver this.

Thus, in evaluating the readiness of this system for production, we need to look closely at Accuracy and Consistency scores.

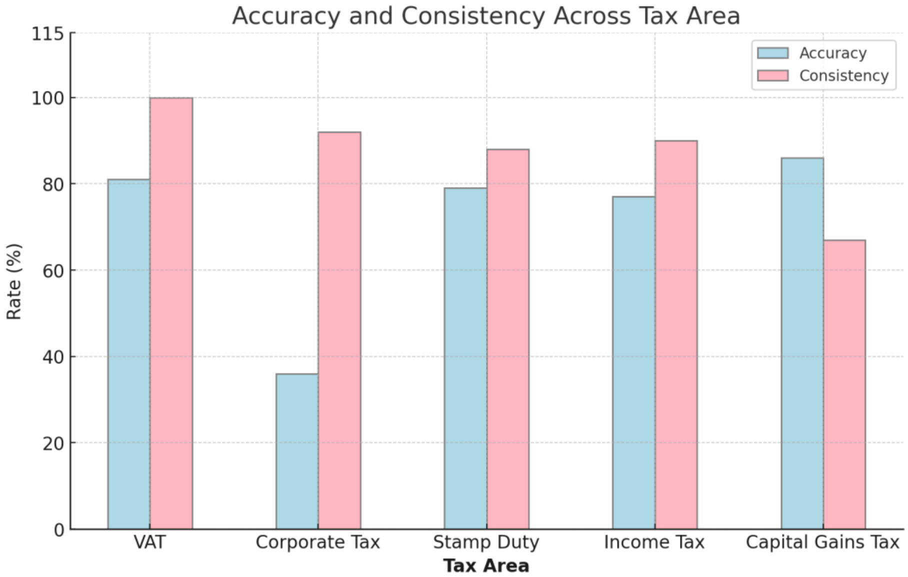


Figure 14: Accuracy and Consistency Across Tax Area

At a first look, consistency doesn't seem to be linked with Accuracy. For example, Capital Gains Tax performances the highest in Accuracy but is the least consistent and Corporate Tax is the worst performing but maintains a high consistency score.

To further understand how to control for consistency, a 0 or 1 to each question to denote whether Specific Knowledge or World Knowledge was used to answer the question.

'Used Specific Knowledge' refers to retrieval and 'Used World Knowledge' refers to GPT-4 Training data. In some cases both were used to answer a question as there were a multiple elements for the model to work through.

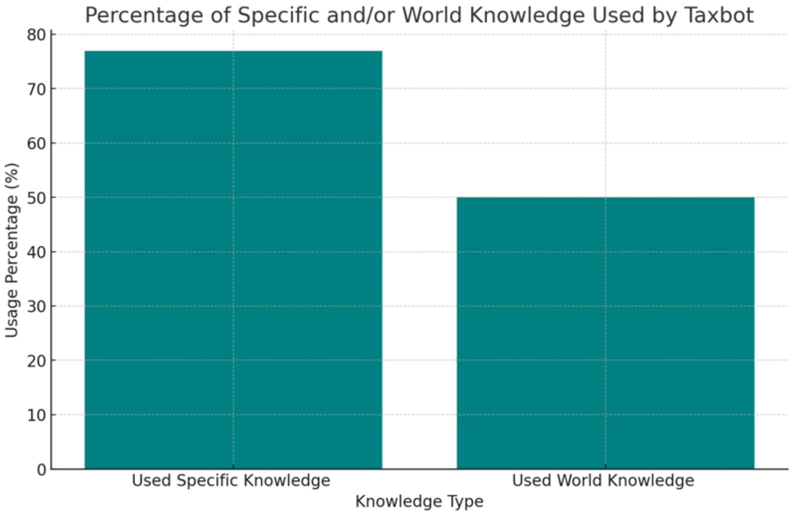


Figure 15: Pecentage of Specific and/or World Knowledge Used by TaxBot

Specific Knowledge was used over 75% of the time in total. However, if break this down by tax category we find that the model used less World Knowledge in areas where accuracy was highest. In Fig 47 we can see a strong link between Accuracy and Specific Knowledge Used, in all areas with the exception of Income Tax.

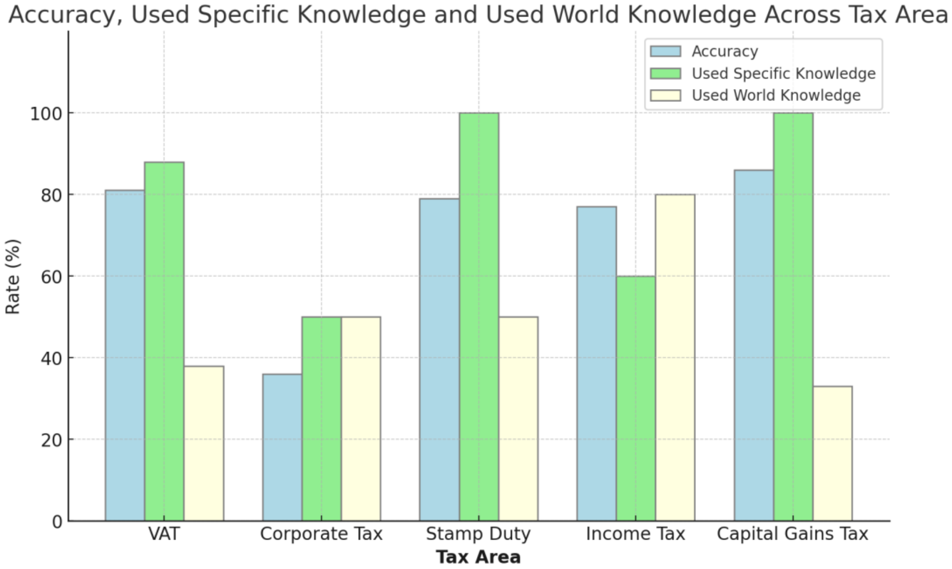


Figure 16: Accuracy, Used Specific Knowledge and Used World Knowledge Across Tax Area

Given that Income Tax performed highly in Accuracy while using more World Knowledge than Specific Knowledge, it indicates that this information may have already been in its training data. This is a sign that larger models like GPT-5 and beyond will be able to do these tasks with raw capabilities alone.

There doesn't seem to be a direct link between Consistency and World or Specific Knowledge used, as can be seen from Fig 48. This indicates that maintaining consistency could require more sophisticated evaluation techniques.

The degree to which consistency is dependent on retrieval data is unclear. However, it is unlikely to be entirely independent. Experimenting further on consistency is likely a qualitative task with multiple components for control, which goes beyond the current scope of the project.

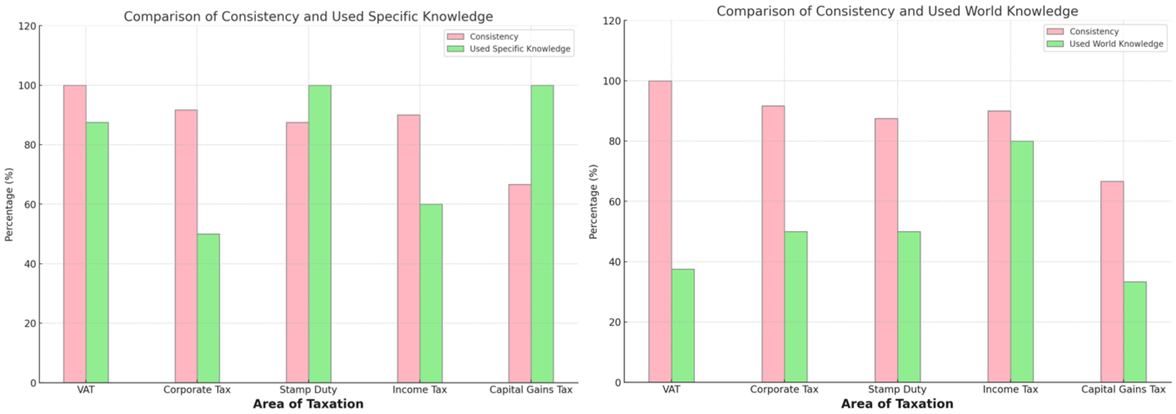


Figure 17: Consistency Across Specific Knowledge and World Knowledge

Overall, as the system returned high levels of Accuracy, World Knowledge Used and Consistency. This appears to be a strong case for further exploration toward production and commercialization.

If the model can continue to perform at this rate when given a larger and more diverse dataset of tax questions, the next natural step would be to experiment with it in an industry setting.

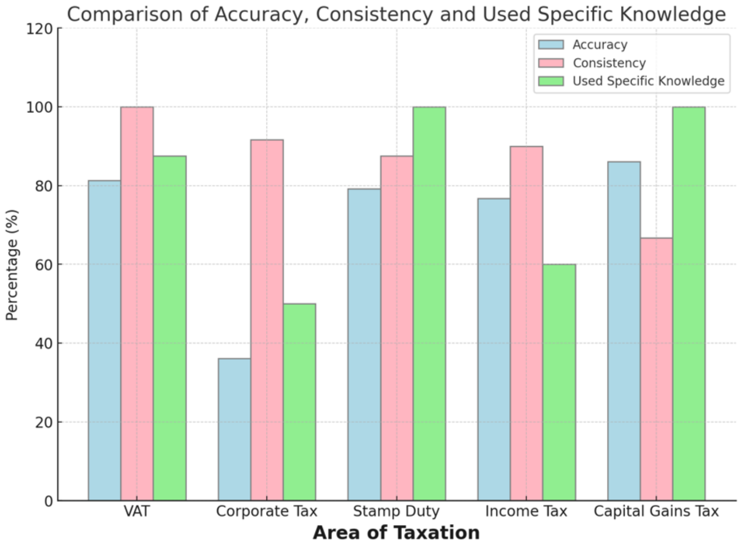


Figure 18: Comparison of Accuracy, Consistency and Used Specific Knowledge

7. Conclusion

In this project, we employed Langchain and Vector Search to harness the capabilities of GPT-4 in constructing an automated Tax Advisory Service.

The recent advancements in Large Language Models and their supporting infrastructure have paved the way for automating aspects of professional advisory services, provided the models have access to the requisite data. The significant potential market opportunity for this product is underscored by the multi-billion-euro tax advisory industry, which has historically been dependent on skilled labour and specialised knowledge.

The design of the system was driven by the goal of developing a minimum viable product for the market. Thus, our core objectives were to create a product that is not only accurate and reliable but also truthful and user-friendly.

Assessing the Irish Tax Advisory Industry, our indicates that a business-to-business model offers favourable distribution and optimises the commercial effectiveness of the system.

TaxBot's capabilities were underscored during its evaluation against qualitative Charted Accountant Proficiency Level 2 Examination questions, where it achieved an average accuracy of 70%. In specific tax domains like VAT and Capital Gains Tax, the model's accuracy soared to 80% or above.

One of the central objectives of this project was to minimize hallucinations. By labelling chunk embeddings based on the originating legislation, the model could reference its information sources. This added layer enhances the system's robustness and reliability.

While the system's flexibility is bolstered by its utilisation of a mix of retrieved information and training data, this approach also introduces a vulnerability to errors.

It is imperative to acknowledge the limitations of our approach. To holistically understand TaxBot's potential, a wider range of qualified questions and a more extensive data set reflecting the true complexities of the tax advisory industry is required.

Future endeavours will focus on enriching & expanding the data for retrieval, diversifying the examination questions, and integrating more advanced evaluation techniques. Pending promising outcomes, the next logical step would be testing in real-world commercial environments.

The initial results are promising. TaxBot could potentially signify a landmark development in tax advisory and set the tone for the evolution of professional advisory services in the AI era. This research demonstrates the current possibilities when leveraging existing & accessible technology within the tax advisory vertical.

Standing at the threshold of this technological integration, TaxBot's accomplishments offer a compelling glimpse into how artificial intelligence might shape the future of tax advisory.

References

[1] CAI (2023) “Overview: CAP2”. Chartered Accounts Ireland Website (retrieved 02/08/2023)
[2] CAI (2023) “Overview: FAE”. Chartered Accountants Ireland Website (retrieved 02/02/2023)
[3] Chai, W. (2022) “Definition: Software as a Service (SaaS).” Could Computing, TechTarget.
[4] Greene at al., (2023) “New and improved embedding model”. Product Announcements, OpenAI.
[5] LangCahin (2023) “LangChain + Chroma”. LangChain Blog. (retrieved 02/07/2023)
[6] LangChain (2023) “RecursiveCharacterTextSplitter.” Indexes, Text Splitters, Langchain Website Documentation. (retrieved 16th of July 2023)
[7] Law Society of Ireland (2023) “Annual Report and Accounts 2021/2022”. Law Society of Ireland.
[8] Microsoft (2023) “What is a vector database?”. Microsoft Semantic Kernel. (retrieved 14/07/2023)
[9] Nakajima, Y. (2023) “Task-driven Autonomous Agent Utilizing GPT-4, Pinecone, and LangChain for Diverse

Applications". Yoheinakajima.com. (retrieved 01/06/2023)

[10] OpenAI (2023) "GPT-4 Technical Report". OpenAI.

[11] Reimers, N. (2022) "OpenAI GPT-3 Text Embeddings - Really a new state-of-the-art in dense text embeddings?". Medium. (retrieved 10/07/2023).

[12] Schwaber-Cohen, R. (2023) "What is a Vector Database?". Pinecone.

[13] Segal, T. (2022) "Freemium: Definition, Examples, Pros & Cons for Business." Investopedia. (retrieved 14/07/2023)

[14] Tam, A. (2023) "A Gentle Introduction to Hallucinations in Large Language Models." Machine Learning Mastery.

[15] Wiggers, K. (2023) "OpenAI intros new generative text features while reducing pricing." TechCrunch.