Solution Vision - POC

Tax BASE

Version: 1.0

Date: 10 December 2023

TABLE OF CONTENTS

DOCUMENT DETAILS

Version Management

Version	Date	Author	Nature of amendment
1.0		Richie Lee	Draft

Reviewer

Name	Version	Date

References

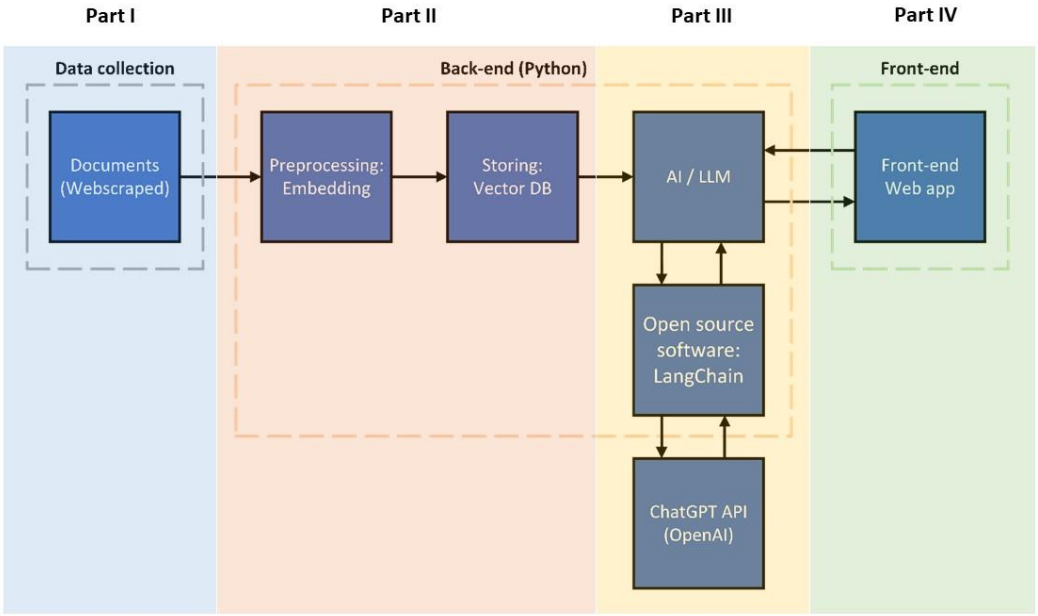
Document name	Version	Date

Intended audience

Target Audience

1 EXECUTIVE SUMMARY

Tax BASE harnesses AI to redefine tax advisory services. It aggregates a wide breadth of BDO's tax documents through automated webscraping, transforming them into a searchable database using **semantic embeddings** and stores them in a **Vector database** (ChromaDB) for efficient **retrieval**. Utilizing **OpenAI's ChatGPT API** and open-source software **LangChain**, the system intelligently processes queries, delivering precise, scalable results. A user-friendly web application facilitates seamless interaction, offering a solid platform ready for future AI advancements in natural language processing.



The BDO Tax BASE system can be structured into four distinct modules:

1. **Data Collection:** This module is focused on gathering tax documents via webscraping.
2. **Data Transformation:** It converts data into a uniform, AI-optimized format, laying the foundation for a centralized repository that supports efficient AI querying.
3. **AI Interaction:** This part of the system streamlines the complexities of interfacing with ChatGPT's API, abstracting away all AI development complexities such that developers can concentrate on crafting effective prompts.
4. **User Interface:** The front-end serves as the user's access point, interacting with the back end through API calls to retrieve information.

2 TECHNICAL DESIGN

2.1 Technical architecture:

2.1.1 System architecture diagram

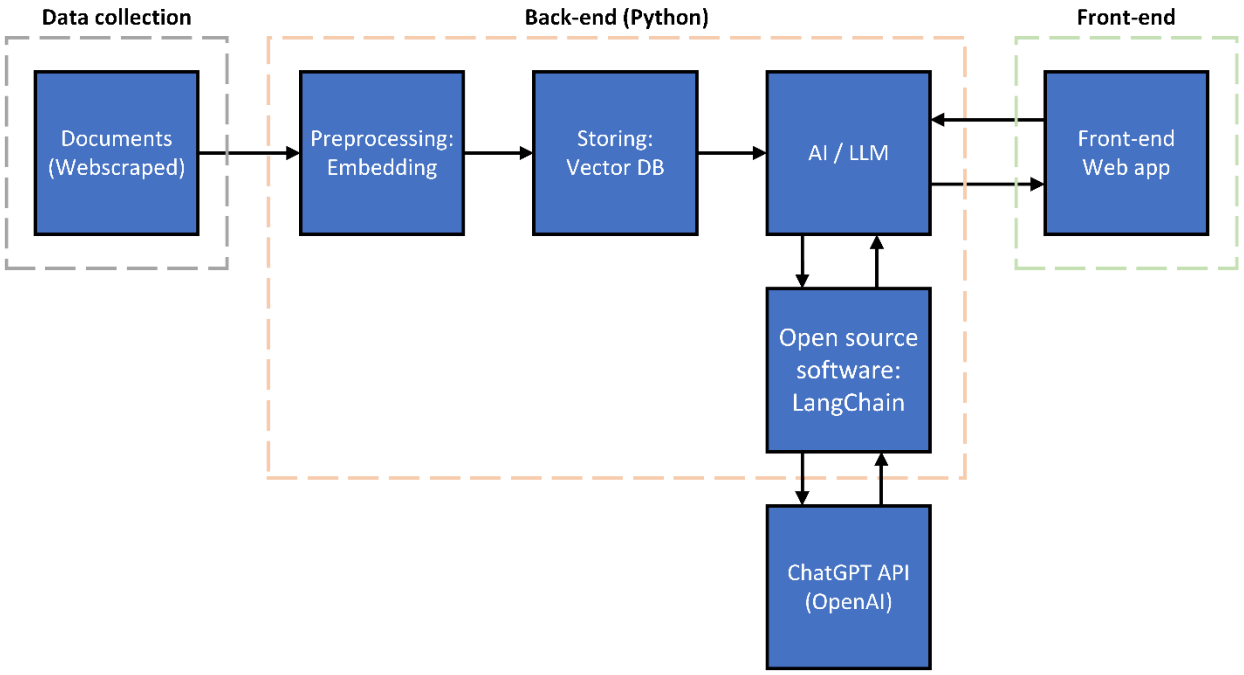


Figure 1 High level overview - technical design Tax BASE

2.1.2 Component descriptions

Documents (Webscraped)

Description: The starting point is the acquisition of tax-related documents via webscraping from BDO websites. Webscraping is the automated method of collecting data from the web, which ensures that the Tax BASE system has a continuous influx of current and comprehensive tax information to analyse and provide to users.

Preprocessing: Embedding

Description: The system preprocesses the raw data to create embeddings, which are numerical representations of text that capture semantic meaning—essentially, the context and underlying ideas within the text. This conversion is crucial for the system to 'understand' and process the text at a conceptual level, making it ready for semantic comparison and retrieval —the process of finding the most contextually relevant documents from the vector database in response to the user's query.

Storing: Vector Database

Description: The embeddings are stored in a vector database, a specialized database optimized for storing and querying high-dimensional data like embeddings. A vector database enables quick and semantically-rich searches, allowing the system to find the most relevant documents efficiently when presented with a user query.

AI / LLM (Large Language Model)

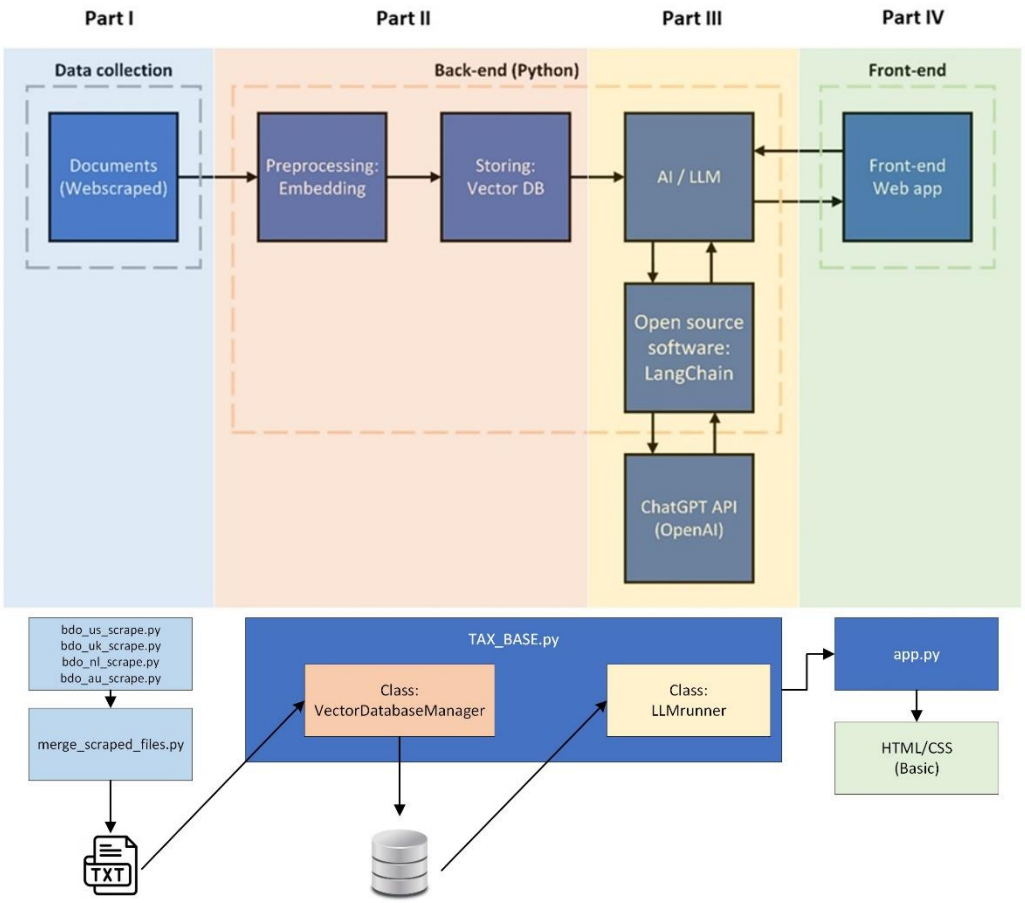
Description: The large language model (LLM), an AI capable of understanding and generating human-like text, is at the heart of the system's processing capability. It employs the ChatGPT API provided by OpenAI, which facilitates the interaction with OpenAI's models to interpret user queries

and generate appropriate responses. LangChain is integrated here as a supporting framework, providing tools and abstractions that make it easier to build applications on top of these language models. The LLM uses the context provided by the embeddings to perform retrieval.

Front-end Web app

Description: The front-end web app is where users interact with the Tax BASE system, inputting their queries and receiving responses. It serves as the user-friendly face of the system, translating the complex processes of data retrieval and language understanding into a simple and intuitive experience for the user.

2.2 Code structure



The BDO Tax BASE architecture mapped to the provided code aligns with the system's operational flow, as follows:

2.2.1 Part I: Data Collection

- **bdo_XX_scrape.py:** This script automates the collection of tax-related articles from BDO UK's website. It uses Selenium with Chrome WebDriver to simulate user interaction, navigating the site, triggering dynamic content loading, and clicking 'Show More' to access

and scrape additional articles. BeautifulSoup parses the HTML content to extract article details, which are then prepared for export.
- **merge_scraped_files.py:** This utility script consolidates scraped data from multiple BDO regional sources. It ensures that all text is encoded uniformly and assembles the individual scraped files into a single comprehensive dataset. Each entry is re-indexed to create a globally unique identifier that amalgamates information from all the different BDO regional insights.

Together, these scripts form the data collection mechanism of Tax BASE, providing the raw input for subsequent processing and analysis.

2.2.2 Part II & III: Data Transformation and AI Interaction Detail

- **Data Transformation (`VectorDatabaseManager`):** This class in `**TAX_BASE.py**` manages the lifecycle of the vector database. It imports documents, supports text and PDF formats, splits them for processing, and creates a vector database using embeddings. These embeddings are generated via OpenAI's `**text-embedding-ada-002**` model, ensuring data is in an AI-friendly format for querying.
- **AI Interaction (`LLMRunner`):** This class abstracts complexities of the ChatGPT API, allowing developers to concentrate on prompt engineering. It retrieves documents from the vector database using similarity search, crafts prompts to assess relevance, and captures the essence of queries for the AI to process. The LLM, specifically `**gpt-3.5-turbo**`, is utilized to generate responses that are contextually relevant to the user's query.

These components transform raw data into an optimized format for retrieval and utilize advanced AI models to interpret user queries and provide relevant responses.

2.2.3 Part IV: Front-End Integration Detail

The `**app.py**` file establishes the front-end web application of Tax BASE, using Flask to serve a responsive user interface. It integrates with the back-end through API endpoints to manage user interactions:

- **Flask Setup:** Flask is configured to handle web requests and serve the main page using `render_template`.
- **Environment Variables:** Securely stored Azure API keys are loaded to interact with the ChatGPT API for backend processing.
- **User Query Handling:** The `/get_response` endpoint processes user queries, leveraging `LLMRunner` to access the vector database and generate responses.
- **Front-End Design:** Accompanying **HTML** and **CSS** files create a user-friendly interface, enabling easy query submission and presenting AI-generated responses in a conversational format.

This module provides the crucial link between Tax BASE's robust AI backend and the end-user, enabling seamless access to the system's capabilities through a web interface.

3 BACK-END LOGIC

3.1 Data sources and management:

3.1.1 Data Sources

Tax BASE leverages documents from currently 4 BDO's regional websites (US, UK, Netherlands, Australia), amounting to approximately 1,500 documents (example: Tax | Insights - BDO). These

documents consist of tax insights, updates, and analyses that are pivotal for delivering tax-related advisory services.

3.1.2 Data Scraping and Collection:

The system uses an automated data scraping process to harvest documents from the aforementioned BDO websites, implemented via the following technologies and methods:

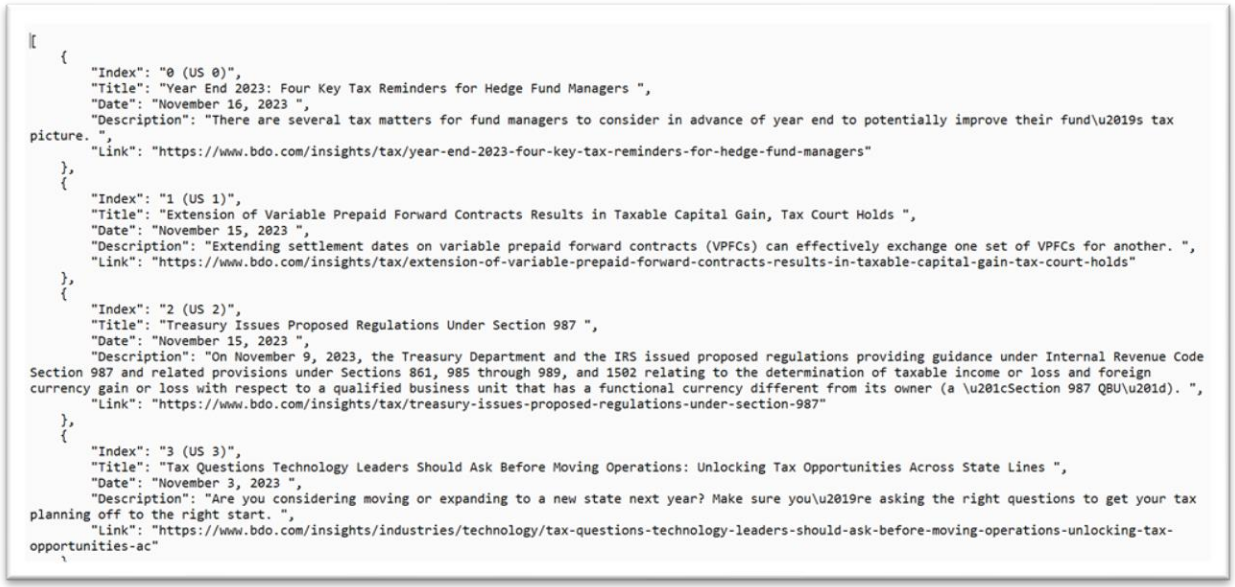
- **Selenium WebDriver**: A browser automation framework used to simulate user interactions with web pages.
    - **ActionChains:** Part of Selenium, used to perform complex mouse actions like hovering, which can trigger dynamic content loading. This is necessary as some BDO insight pages trigger pop-ups that prevent successful scraping otherwise.
- **WebDriverWait and Expected Conditions:** These tools from Selenium help in synchronizing the scraping process to wait for certain conditions, ensuring that dynamically loaded elements are present before proceeding. This is necessary due to the built-in animations in the BDO insights pages.
- **BeautifulSoup**: A Python library for parsing HTML documents, extracting relevant information from the web page's DOM structure. Though the websites look similar, the HTML structure are different and need manual implementation with each additional website.
- **Data Encoding and Storage:** Different character encodings, such as UTF-8 and ISO-8859-1, are accounted for when reading scraped content to ensure textual data is accurately captured and preserved.

3.1.3 Data Storage and Management

For storage and management of the processed data, Tax BASE utilizes:

- **ChromaDB (Vector Database)**: An AI-native open-source embedding database optimized for storing and querying vectorized representations of text documents, facilitating semantic searches.
    - https://python.langchain.com/docs/integrations/vectorstores/chroma
    - https://api.python.langchain.com/en/latest/vectorstores/langchain.vectorstores.chroma.Chroma.html
- **LangChain:** An open-source library that interfaces with Chroma to streamline operations, making the use of the vector database more accessible and efficient.
- **Export Functions:** Custom Python functions "*str_to_txt_file*" and "*read_file_with_fallback*" are implemented to write the scraped content to text files, with error handling for different encoding scenarios.

Visually, we can imagine the data looking as follows:




```
[
    {
        "Index": "0 (US 0)",
        "Title": "Year End 2023: Four Key Tax Reminders for Hedge Fund Managers ",
        "Date": "November 16, 2023 ",
        "Description": "There are several tax matters for fund managers to consider in advance of year end to potentially improve their fund\u2019s tax
picture. ",
        "Link": "https://www.bdo.com/insights/tax/year-end-2023-four-key-tax-reminders-for-hedge-fund-managers"
    },
    {
        "Index": "1 (US 1)",
        "Title": "Extension of Variable Prepaid Forward Contracts Results in Taxable Capital Gain, Tax Court Holds ",
        "Date": "November 15, 2023 ",
        "Description": "Extending settlement dates on variable prepaid forward contracts (VPFCs) can effectively exchange one set of VPFCs for another. ",
        "Link": "https://www.bdo.com/insights/tax/extension-of-variable-prepaid-forward-contracts-results-in-taxable-capital-gain-tax-court-holds"
    },
    {
        "Index": "2 (US 2)",
        "Title": "Treasury Issues Proposed Regulations Under Section 987 ",
        "Date": "November 15, 2023 ",
        "Description": "On November 9, 2023, the Treasury Department and the IRS issued proposed regulations providing guidance under Internal Revenue Code
Section 987 and related provisions under Sections 861, 985 through 989, and 1502 relating to the determination of taxable income or loss and foreign
currency gain or loss with respect to a qualified business unit that has a functional currency different from its owner (a \u201cSection 987 QBU\u201d). ",
        "Link": "https://www.bdo.com/insights/tax/treasury-issues-proposed-regulations-under-section-987"
    },
    {
        "Index": "3 (US 3)",
        "Title": "Tax Questions Technology Leaders Should Ask Before Moving Operations: Unlocking Tax Opportunities Across State Lines ",
        "Date": "November 3, 2023 ",
        "Description": "Are you considering moving or expanding to a new state next year? Make sure you\u2019re asking the right questions to get your tax
planning off to the right start. ",
        "Link": "https://www.bdo.com/insights/industries/technology/tax-questions-technology-leaders-should-ask-before-moving-operations-unlocking-tax-
opportunities-ac"
    \
```


3.1.4 Data Integration and Indexing:

The final stage of data preparation involves:

- **Merging Data:** Scraped content from each BDO region is combined into a single text file, ensuring a consolidated database for Tax BASE.
- **Global and Local Indexing:** A unique indexing system is applied where each document is assigned a global index and a local index corresponding to its source (e.g., BDO US, UK, NL, AU), facilitating document tracking and retrieval.

3.2 Embedding and Machine Learning

3.2.1 Embedding Process

The embedding process in Tax BASE is a critical step that transforms textual information into a numerical format that can be interpreted by machine learning algorithms. This process involves:

- **Text Vectorization:** Textual data from documents is converted into vectors using algorithms that capture the semantic meaning of the words. This is done by analysing the context in which each word appears within the document and mapping it to a high-dimensional space.
- **Embedding Models:** Tax BASE utilizes pre-trained models provided by OpenAI, which are designed to generate embeddings. These models have been trained on a wide corpus of text, enabling them to understand and encode a rich array of semantic nuances.
- **Dimensionality:** The resultant embeddings are high-dimensional vectors that effectively represent the semantic signature of each document or chunk of text, which allows for nuanced comparison and retrieval based on content similarity.

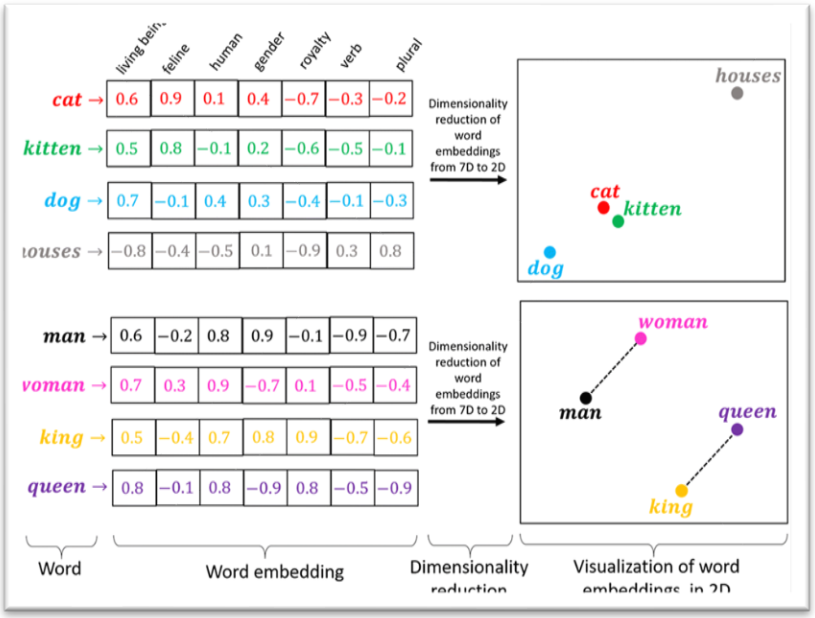


Figure 2 Words numerically represented as vectors through Embeddings

3.2.2 Machine Learning Models

The machine learning models in Tax BASE are responsible for understanding user queries and matching them with the most relevant document embeddings. These models are characterized by:

- **Model Selection:** Tax BASE employs models that are suitable for natural language processing (NLP) tasks, such as those based on the Transformer architecture, known for their effectiveness in understanding context and generating human-like text.
    - Key literature for in depth understanding (advanced): https://arxiv.org/abs/1706.03762
- **Development and Training:** While the core models are pre-trained by OpenAI, Tax BASE may fine-tune these models using specific tax-related documents to enhance their performance on domain-specific queries.
- **Updating Process:** The models are periodically updated to incorporate new data and user feedback, ensuring that the system remains current with the latest tax information and advisory practices.
- **Feedback Loop:** User interactions and query results are analysed to continuously improve the model's accuracy and efficiency, creating a dynamic system that evolves with its user base and data environment.

By employing these embedding and machine learning processes, Tax BASE ensures that tax professionals receive precise, contextually relevant information in response to their queries, leveraging the latest advancements in AI and natural language understanding.

3.2.3 Integration with Azure services

Tax BASE utilizes two key Azure resources for its core functionalities:

1. **Embedding Model Connection:** Utilizes Azure's `text-embedding-ada-002` model to convert documents into embeddings.

2. **Retrieval/LLM Functionality:** Employs the `gpt-3.5-turbo` model for query processing and document retrieval.

To set up these resources in Azure, follow this path: *Azure OpenAI resource → Model deployments → Manage deployments → Create new deployments*.

These models, selected for their efficiency and accuracy, are central to Tax BASE's operation. While not yet tested with alternative models, the system is designed to be easily adaptable to different Azure AI models if necessary. This flexibility ensures Tax BASE can evolve with technological advancements and shifting user needs.

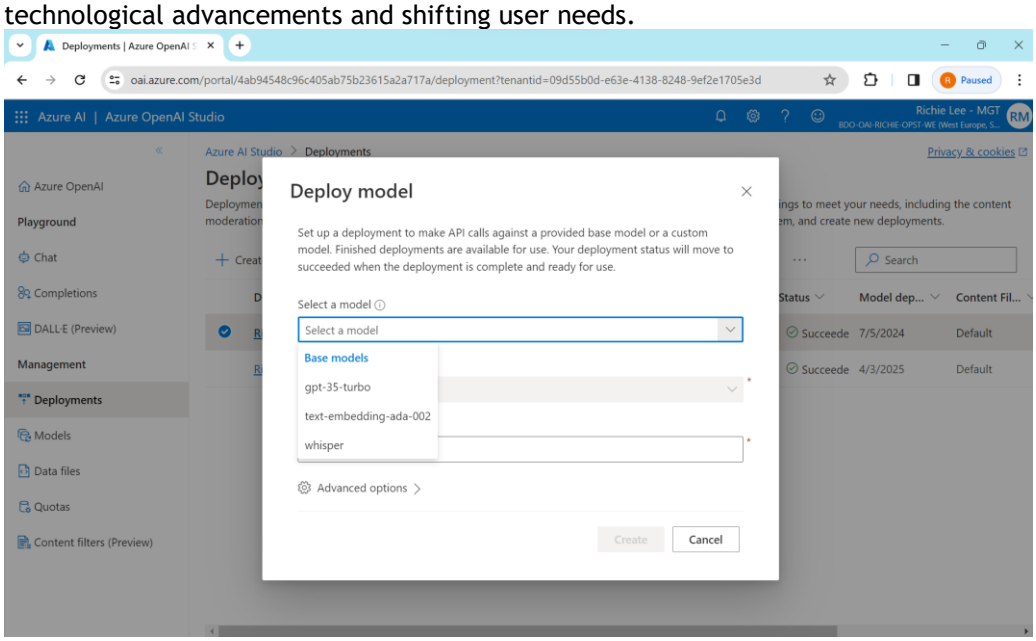


Figure 3 Azure OpenAI studio - specifying embedding and retrieval model

3.3 Query Processing and Retrieval

3.3.1 Retrieval Algorithms:

Tax BASE employs a sophisticated retrieval algorithm to efficiently narrow down the most relevant document embeddings from its extensive database, based on user queries. This process involves:

- **Query Interpretation**: Utilizing the `gpt-3.5-turbo` model from Azure, the system first interprets the user's query to understand the context and specific information needs.
- **Semantic Search**: The query is then processed through a semantic search algorithm, which compares the query's embedding (generated by the `text-embedding-ada-002` model) with the embeddings of documents stored in the vector database.
- **Scalable Retrieval**: The retrieval algorithm is designed for scalability, allowing it to handle an increasing volume of documents and queries without significant loss in performance. This is achieved by optimizing the search within the high-dimensional space of embeddings, ensuring rapid and accurate results even as the database grows.

3.3.2 Relevance Scoring:

Relevance scoring is a critical component of Tax BASE's retrieval process, determining how closely each document matches the user's query. This process includes:

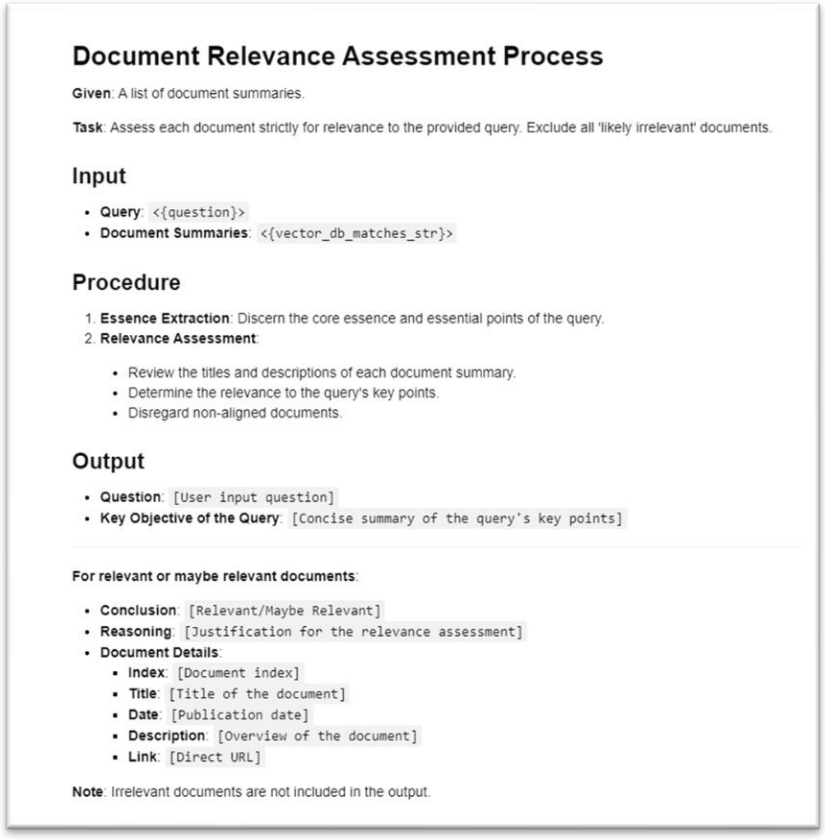
- **Cosine Similarity:** The system uses cosine similarity to score the relevance of documents. First, we obtain the embedded version of the questions, as well as all the embedded documents from the vector database. Then, Cosine similarity measures the cosine of the angle between the query vector and document vectors, with a higher cosine value indicating greater relevance.
- **Ranking and Shortlisting:** Documents are ranked based on their relevance scores. The system then shortlists a subset of the highest-scoring documents for presentation to the user.
- **Thresholding:** To further enhance efficiency, a threshold can be set to filter out documents with relevance scores below a certain level, ensuring only the most pertinent documents are considered. This step is executed through a normal LLM prompt.

This combination of advanced retrieval algorithms and relevance scoring underpins Tax BASE's ability to deliver targeted, relevant tax information efficiently. The system's design facilitates scalability, handling growing data and user requests while maintaining its effectiveness and speed in information retrieval.

3.3.3 Prompt engineering

Once a shortlist of potentially relevant documents is created (considerably reducing the scalability limitation), we can now apply LLM prompts. The prompting follows concepts from https://learn.deeplearning.ai/chatgpt-prompt-eng/lesson/8/chatbot.

The current prompt that is used looks as follows:



Document Relevance Assessment Process

Given: A list of document summaries.

Task: Assess each document strictly for relevance to the provided query. Exclude all 'likely irrelevant' documents.

Input

- **Query**: `<{question}>`
- **Document Summaries**: `<{vector_db_matches_str}>`

Procedure

1. **Essence Extraction**: Discern the core essence and essential points of the query.
2. **Relevance Assessment**:
   - Review the titles and descriptions of each document summary.
   - Determine the relevance to the query's key points.
   - Disregard non-aligned documents.

Output

- **Question**: `[User input question]`
- **Key Objective of the Query**: `[Concise summary of the query's key points]`

For relevant or maybe relevant documents:

- **Conclusion**: `[Relevant/Maybe Relevant]`
- **Reasoning**: `[Justification for the relevance assessment]`
- **Document Details**:
  - **Index**: `[Document index]`
  - **Title**: `[Title of the document]`
  - **Date**: `[Publication date]`
  - **Description**: `[Overview of the document]`
  - **Link**: `[Direct URL]`

Note: Irrelevant documents are not included in the output.

Figure 4 LLM prompt that assesses document relevance from (retrieved) shortlist.

3.4 User interface and interaction

The user interface and interaction design of Tax BASE focuses on simplicity and efficiency, ensuring that users can easily access the powerful capabilities of the backend without needing to navigate a complex interface. The Flask framework facilitates a seamless connection between the frontend and the sophisticated query processing and retrieval logic of the backend.

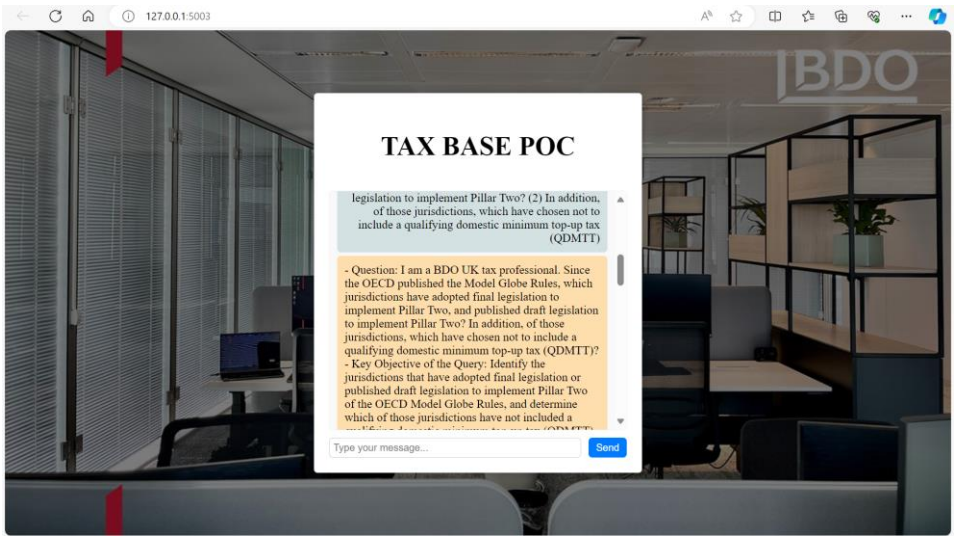


Figure 5 Tax BASE POC user interface

3.4.1 Interface Description

The user interface of Tax BASE is designed to be intuitive and user-friendly, enabling easy interaction for tax professionals. The system uses a web application framework, specifically Flask, to create a responsive and accessible interface. Key aspects include:

- **Web Application Framework:** Flask, a lightweight and versatile web framework in Python, is used to build the interface.
- **Main Page Rendering:** The homepage, rendered through `render_template("index.html")`, presents users with a clean and straightforward layout for inputting queries.
- **Responsive Design:** The interface is designed to be responsive, ensuring usability across various devices and screen sizes.

3.4.2 Query Handling and Presentation:

The process of handling user queries and presenting results is streamlined and efficient:

- **Query Submission:** Users submit their queries via an input form on the web application. This is managed by the `/get_response` route, which handles POST requests.
- **Query Processing:** Upon receiving a query, the Flask application communicates with the backend, where the `LLMRunner` and `VectorDatabaseManager` classes process the query. This involves:
    - Selecting a subset of documents from the vector database using the query.
    - Running the large language model to generate a response based on the query and selected documents.
- **Error Handling:** The system includes error handling to provide feedback to the user in case of any issues during query processing.
- **Result Presentation:** The response from the large language model, which includes the most relevant information extracted from the vector database, is then sent back to the front end and displayed to the user in a clear and concise format.

3.4.3 Configuration and Security:

Set up the Azure API key and base/endpoints, follow this path: *Azure OpenAI resource → Keys and Endpoints (Resource management)*.

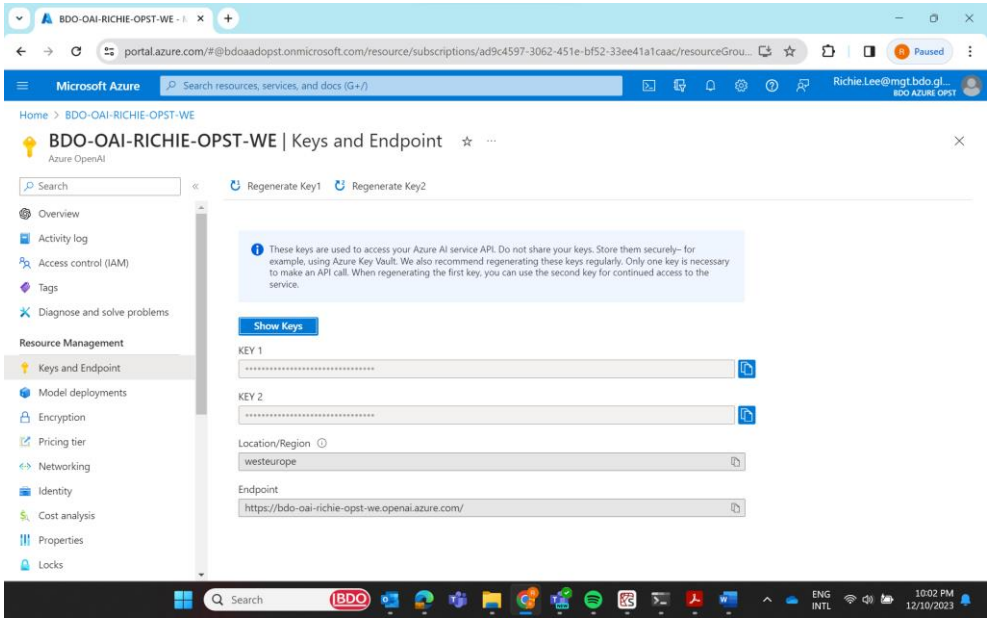


Figure 6 Azure API key and endpoint/base location

As for security and deployment, we can consider the following:

- **Environment Variables**: Sensitive information such as Azure API keys is securely stored and accessed using environment variables, ensuring the security of the system.
    - **Note:** for seamless configuration the environment variables should be named "*AZURE_OPENAI_API_KEY*" and "*AZURE_OPENAI_API_BASE*" for the API key and endpoint respectively.
- **Deployment**: The Flask application can be run locally for development and testing, as shown in the code, and can also be deployed on a server for broader accessibility.

4 REQUIREMENTS

4.1 Core concepts (for developers)

In order to fully understand the back-end, developers approaching the Tax BASE project should have a good understanding of the following key areas:

- **NLP & LLMs:** Understand Natural Language Processing and Large Language Models like GPT-3 for language interpretation and response generation.
- **Semantic Embeddings:** Knowledge in converting text to numerical vectors for semantic analysis in AI models.
- **Vector Database Management:** Skills in managing (language model) databases like ChromaDB for storing and querying vector data.
- **Retrieval Algorithms:** Familiarity with search algorithms for efficiently ranking data relevance.
- **Prompt Engineering:** Ability to craft prompts that effectively communicate with AI models to extract desired outcomes.

A comprehensive overview of all relevant terminology can be found in the Terminology section.

4.2 Installation guide

- **Dependencies:** Python 3.11.7, Langchain 0.0.337, pypdf 3.16.4, openai 0.28.1, chromadb 0.4.17, tiktoken 0.5.1
- **IDE**: Anaconda (Spyder)

5 FAQ

- **Q: How scalable is the system?**
  A: Tax BASE is highly scalable. By embedding data, we significantly reduce its size, ensuring that even a large volume of documents is manageable. Additionally, the LLM's shortlisting functionality means that response generation isn't limited by data size.

- **Q: Does the system experience hallucinations in responses?**
  A: To minimize the risk of hallucinations, we've implemented a strategy that matches queries strictly to relevant documents. This approach also maintains a human-in-the-loop, enhancing the reliability of the responses.

- **Q: What content is used for matching queries to documents?**
  A: We primarily use article titles, dates, descriptions, and links for matching. Full documents can also be included for more context, but this adds complexity to the backend. For the proof of concept, we've prioritized efficiency and timeline, hence the current approach. Only BDO documents are used as sources, with no external data.

- **Q: Is the system's response always correct?**
  A: While the AI is adept at inference, its accuracy is contingent on the available data. If the information needed to answer a query isn't in the database, the response may be limited. It's important to distinguish between data quality and the AI's functionality.

- **Q: Does Tax BASE use any proprietary data?**
  A: No, Tax BASE only utilizes publicly available information from BDO documents and does not rely on private data sources. However, users should be cautious about including private information in their queries.

- **Q: How can components effectively communicate with each other when deployed across multiple infrastructure resources and is TLS the preferred method for this?**
  A: Final decision TBD, but for now:
  "TLS can be used in this case. The Azure services for each identified component should be defined according to the project architecture. On a high level, these components will usually be communicating through APIs via HTTP. Services such as Azure App Services and Azure Storage services have dedicated panels allowing administrators to configure TLS settings in a straightforward way."

6 TERMINOLOGY

Term	Category	Description
Agents	AI & Machine Learning	Entities in AI that perform actions in an environment to achieve specific objectives.
Azure OpenAI API	Cloud Computing	Microsoft Azure's platform for accessing OpenAI's powerful AI models.
Chains	AI & Machine Learning	Sequences of actions or events in AI processes, often used in decision-making models.
Chroma	Data Storage	A vector database optimized for storing and querying high-dimensional data.
Context	AI & Machine Learning	Information that AI models use to understand and generate relevant responses.
Dimensionality	Data Management	The number of dimensions (attributes or features) that represent data in a model.
Embedding (Semantic)	Data Preprocessing	Converting text into numerical vectors to capture semantic meanings in AI models.
Environment Variables	Software Development	Variables in an operating system used to configure application behavior.
Flask	Web Development	A lightweight Python web framework for building web applications.
GPT-3.5 Turbo	AI Models	A version of OpenAI's generative pre-trained transformer model.
Hallucinating	AI & Machine Learning	When an AI model generates incorrect or nonsensical information.
LangChain	AI Integration	An open-source library facilitating efficient integration with language models.
Large Language Model (LLM)	AI & Machine Learning	Advanced AI models capable of understanding and generating human-like text.
Memory	AI & Machine Learning	The component of an AI that stores information from past interactions or data inputs.
Natural Language Processing (NLP)	AI & Machine Learning	The field of AI focused on the interaction between computers and human language.
Prompt Engineering	AI Interaction	Crafting effective prompts to elicit desired responses from AI models.
Python	Programming Language	A high-level programming language known for its readability and broad applicability in web development.
Retrieval Algorithms	Data Retrieval	Algorithms used to find and rank the relevance of data in response to queries.
Semantic Search	Data Retrieval	A search technique that understands the user's query contextually.
Temperature	AI & Machine Learning	A parameter in AI models that controls the randomness of the generated responses.
Text Embedding-ada-002	AI Models	An OpenAI model used for generating text embeddings.
Thresholding	Data Analysis	The process of setting a cut-off value to determine the significance level in data analysis.
Tokens	AI & Machine Learning	Basic units of data processed by language models in NLP.
Vector	Data Management	A mathematical representation of data in AI models, often in high-dimensional space.
Webscraping	Data Collection	Automated process of extracting data from websites.