Generative AI Data Ingestion System - Analytical Data Systems

In this project, you'll have the opportunity to work with advanced language models like ChatGPT and Mistral 7B and explore retrieval-augmented generation (RAG) systems. RAG is a powerful approach that enhances the capabilities of pre-trained language models by integrating external knowledge retrieval. This means you can build systems that generate more accurate and coherent responses by leveraging relevant information from a knowledge base.

Throughout the project, you'll gain hands-on experience with popular technologies such as Python, Google Colab, LlamalIndex, and ChatGPT. You'll learn how to integrate with the Google Docs API to retrieve documents, convert them to a unified markdown format, and efficiently chunk the content for processing. You'll also dive into the world of embedding generation, experimenting with various techniques like word embeddings, sentence embeddings, and document embeddings to create dense vector representations of text data.

One of the key aspects of this project is evaluating the retrieval quality of the generated embeddings. You'll design and implement comprehensive testing frameworks to assess the accuracy and relevance of the retrieved documents based on different queries. This will involve developing evaluation metrics, conducting experiments, and analyzing the results to identify the most effective embedding techniques and hyperparameter configurations.

By the end of this 5-week project, you'll have developed a robust LLM Embedding Ingestion Engine that showcases your skills in document processing, natural language processing, embedding generation, and retrieval quality evaluation. This project will not only enhance your technical abilities but also provide you with valuable experience in designing and conducting experiments, analyzing results, and optimizing performance.

If you're passionate about AI and want to work on a project that combines cutting-edge techniques with real-world applications, this Generative AI Data Ingestion System is the perfect opportunity for you. Get ready to dive into the exciting world of generative AI and make a meaningful contribution to the field. Let's get started!

**Technologies:** Python, Google Colab, LlamalIndex, ChatGPT, Git

**Overall Project Goal:** To build a python library that a data ingestion engine from a notebook that can process and quantitatively test and evaluate various chunking, embedding and retrieval techniques with llamaindex for use in AI pipelines.

**IP Rights:** Analytical Data Systems will own all rights to deliverables, including code, data, prompts, and documentation. Feel free to use the know-how to advance your career.

**Project Scope:**

**Week 1:**
Set up the Google Colab environment and install the necessary dependencies for the project, including libraries for Google Docs API integration, document conversion, and natural language processing.

Obtain access to a Google Docs folder containing a diverse set of documents (e.g., text documents, PDFs, presentations) that will be used as the input data for the ingestion engine.

Develop a Python script that utilizes the Google Docs API to authenticate and retrieve the list of documents from the specified folder.

Implement a document conversion module that handles different file formats (e.g., .docx, .pdf, .txt) and converts them to a unified markdown format.

Explore and evaluate various document conversion libraries to ensure accurate and reliable conversion of the documents to markdown.

Perform thorough testing of the document conversion module to handle edge cases, formatting variations, and potential errors gracefully.

Week 2:
- Design and implement a chunking strategy that efficiently splits the converted markdown documents into meaningful and manageable chunks.
- Experiment with different chunking approaches, such as fixed-size overlapping windows, sentence-based chunking, or semantic chunking based on topic coherence.
- Embed and store embeddings in ChromaDB.
- Develop a query pipeline with llmaindex with retrievers to query the embedded data.

Week 3:
- Develop evaluation metrics and conduct experiments to assess the effectiveness of the chunking strategy in preserving the context and coherence of the original documents.
- Run and store experiments with quantifiable testing output to compare the performance of the different strategies based on the questions being asked.

Week 4:
- Design and implement a comprehensive testing framework to evaluate the retrieval quality of the generated embeddings with llmaindex.
- Develop a set of relevant queries and ground truth data to assess the accuracy and relevance of the retrieved documents based on the embeddings.
- Conduct repeated tests using different embedding techniques and hyperparameter configurations to identify the most effective approach for the given dataset.
- Analyze the retrieval quality metrics to quantitatively evaluate the performance of the embedding ingestion engine.

Week 5:
- Prepare a detailed report summarizing the findings, including the comparison of different embedding techniques, chunking strategies, and their impact on retrieval quality based on the type of document and type of question.
Throughout the project, you will gain expertise in working with document conversion, chunking strategies, embedding generation, and retrieval quality evaluation. You will also develop skills in designing and conducting experiments, analyzing results, and optimizing the performance of the embedding ingestion engine.

The project will span 5 weeks, with each week dedicated to specific tasks and milestones. Regular check-ins and progress updates will be conducted to ensure smooth progress and address any challenges encountered along the way.

By the end of this project, you will have developed a robust and efficient LLM Embedding Ingestion Engine that can process documents from a Google Docs folder, convert them to markdown, chunk the content, generate embeddings, and evaluate retrieval quality. This experience will equip you with valuable skills in document processing, natural language processing, embedding techniques, and evaluation methodologies.