GRADUATE CERTIFICATE
EBA 5004
PRACTICAL LANGUAGE PROCESSING (PLP)
PROJECT MODULE

PROJECT REPORT

GROUP 1
SEMANTIC FORCE

Yatharth Mahesh Sant (A0286001R)
Kristofer Roos (A0285949A)
Goh Min Hua (A0285810A)
Tan Li Ming (A0027883W)
Chua Kian Yong Kenny (A0056377W)

Table of Contents

1. EXECUTIVE SUMMARY

The SemanticForce project developed a system that can automatically process annual reports, 10-K filings, and relevant news articles for a specified set of companies. The system will then generate concise summaries, extracting key financial information, strategic outlooks, and media sentiment surrounding these companies.

This NLP-powered solution offers numerous benefits. It streamlines the information gathering process, saving significant time and resources compared to manual analysis. Additionally, by processing large volumes of text data, the system can identify trends and insights that might be missed by traditional methods. The generated summaries will provide a comprehensive overview of each company's financial health, strategic direction, and public perception, empowering consultants to process and interpret large datasets in an efficient manner and improve the quality of insights provided to clients.

2. BUSINESS CASE AND MARKET RESEARCH

2.1 Project Sponsor and Purpose

The SemanticForce project was developed in a collaborative effort with a leading strategic consulting firm, which has requested to remain anonymous for the publication of this project. This partnership was not merely supportive but integrative, with the consulting firm advising on the design and business needs of the project. This deep involvement ensured that the tool was not only theoretically sound but also practical and tailored to meet the specific needs encountered in strategic consulting. The firm's commitment to the project underscores its potential and the trust placed in its capabilities to revolutionize data processing and analysis in consulting practices.

The consulting firm's primary interest in SemanticForce stems from its potential to efficiently manage, analyze, and summarize large volumes of unstructured textual data—a frequent and critical challenge in strategic consulting. The tool's ability to autonomously extract key business metrics from diverse documents, such as annual reports and market analyses, positions it as a pivotal asset in the consulting toolkit. By adopting SemanticForce, the firm aims to significantly enhance the efficiency and accuracy of its consulting services, enabling consultants to derive actionable insights more swiftly and support clients in making well-informed, data-driven decisions.

The consulting firm’s teams played a crucial role in shaping the design of SemanticForce, providing extensive input to ensure the tool meets the high demands of strategic consulting. Their interests were instrumental in refining features such as sentiment analysis and Q&A interactions, making these elements highly relevant to current consulting practices. The feedback from the firm helped tailor SemanticForce to handle the nuanced language and complex constructs typical of strategic reports and financial documents. This collaboration was key in aligning the tool’s capabilities with the real-world needs of consultants, ensuring it is both effective and user-friendly.

Following the development phase, SemanticForce is planned for immediate deployment within the consulting firm to validate its practical utility in live consulting scenarios. This phase of the project is critical, as it will demonstrate the tool's ability to enhance the firm's analytical processes and decision-making prowess. By improving how consultants process and interpret large datasets, SemanticForce is expected to significantly boost operational efficiency and the quality of insights provided to clients. The firm anticipates that this strategic advantage will translate into deeper, more informed analyses and better outcomes for their clients, reinforcing the tool's value proposition in competitive business environments.

After the ISS exams, a strategic meeting is planned with the consulting firm to facilitate the deployment of SemanticForce and integrate it seamlessly into their business operations. This session will focus on customizing the tool to fit perfectly within the existing technological and operational frameworks of the firm, ensuring that it complements and enhances current methodologies. The hands-on involvement in setting up and optimizing the tool will be crucial for smoothing any technical and user experience edges, making SemanticForce a core component of the firm's consulting infrastructure. This deliberate and careful integration process is designed to guarantee that the tool not only functions efficiently but also becomes an indispensable part of the consulting workflows, delivering on its promise to streamline data-driven strategic decision-making.

2.2 General Market Needs and Opportunities

Businesses today face the challenge of processing large volumes of unstructured textual data efficiently. With the increasing importance of data-driven decision-making, there is a growing demand for tools that can transform unstructured textual data into actionable business insights.

However, amidst the time constraints and the overwhelming information available from multiple data sources, businesses struggle with managing diverse data sources, analyzing large volumes of unstructured data, and extracting actionable insights in a timely manner. Therefore, there is an urgent need for accurate and reliable text analysis tools that can address these challenges and enable businesses to derive meaningful insights from textual data within tight time constraints.

Thereby, our proposed solution, SemanticForce, is developed to meet this growing demand by leveraging on text analytics and machine learning methodologies. SemanticForce is designed to autonomously extract key business metrics from the diverse user-submitted reports. Furthermore, it offers sentiment analysis capabilities to extract insights from latest news pertaining to the company of interest. Additionally, SemanticForce will enable text summarization and facilitate Q&A interactions on submitted reports.

The market potential for SemanticForce is significant, with a primary focus on meeting the needs of strategic consulting firms. SemanticForce aims to empower consultants to extract key business insights efficiently, thereby facilitating an informed and data-driven decision-making process.

3. SYSTEM DESIGN AND MODEL

3.1 Architecture Overview

The system provides a user-friendly interface to streamline the process of gathering current and comprehensive information on the company of interest.

Streamlit Chatbot Interface Description

Our Streamlit-based chatbot integrates the capabilities of OpenAI's Llama2 model to provide a responsive and adaptive user experience for processing financial documents and answering inquiries. Designed with functionality in mind, the interface accepts user inputs such as uploaded annual and 10K reports, and requests for company-specific news or information extraction. The system is structured to seamlessly handle various types of queries:

1. Document Summarization: The chatbot employs specialized summarization models (T5 and BART) to generate concise summaries of extensive financial documents, including 10K reports and annual reports.
2. News Retrieval: Upon request, the chatbot fetches and summarizes the latest news pertaining to specified companies using an integrated information extraction model.
3. Information Extraction via Q&A: Leveraging a sophisticated question-answering model, the chatbot interprets and responds to user queries by extracting relevant information from the provided documents or accumulated data.
4. Contextual Q&A: A Retrieval Augmentation Generation (RAG) framework is being built upon with wikipedia pages as its document database to handle any questions that may be relevant to extract other insights outside of financial scope.
5. Interactive Dialogue: The Llama2 model also supports dynamic interaction, allowing the chatbot to handle follow-up questions and engage in a conversational manner based on the model’s judgments and learned knowledge.

This interface is designed to enhance decision-making and provide swift access to synthesized information, streamlining the analysis of complex financial data through an intuitive chat-based platform.

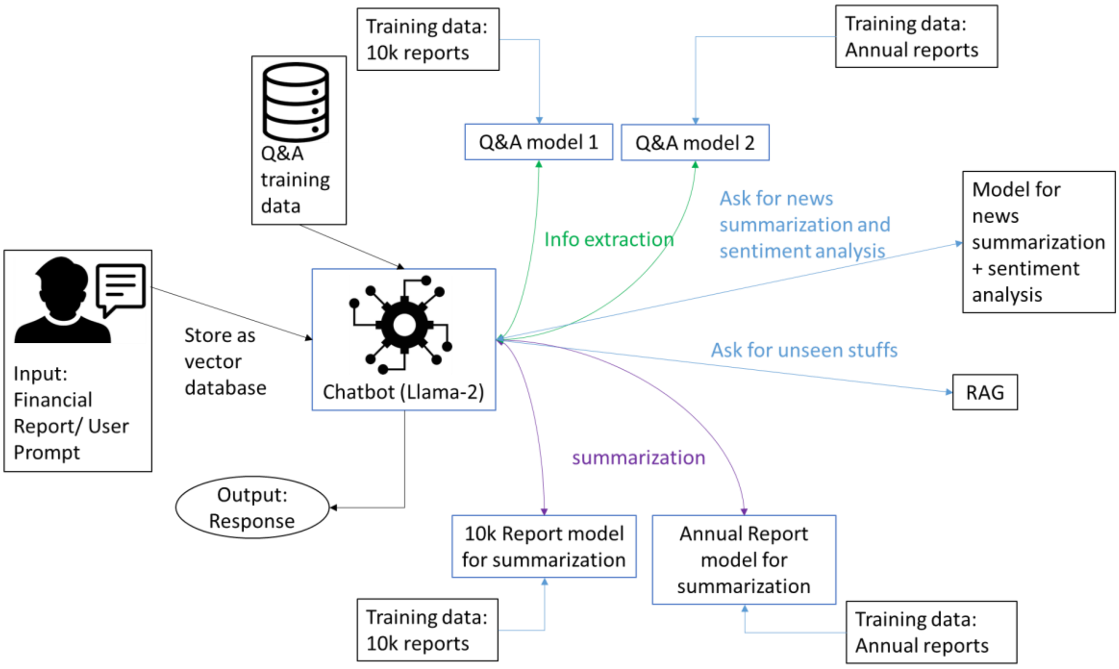

Training data: 10k reports
Training data: Annual reports
Q&A training data
Q&A model 1
Q&A model 2
Info extraction
Ask for news summarization and sentiment analysis
Model for news summarization + sentiment analysis
Input: Financial Report/ User Prompt
Store as vector database
Chatbot (Llama-2)
Ask for unseen stuffs
RAG
summarization
Output: Response
10k Report model for summarization
Annual Report model for summarization
Training data: 10k reports
Training data: Annual reports

Figure 1: System Architecture

3.2 Detailed Workflow

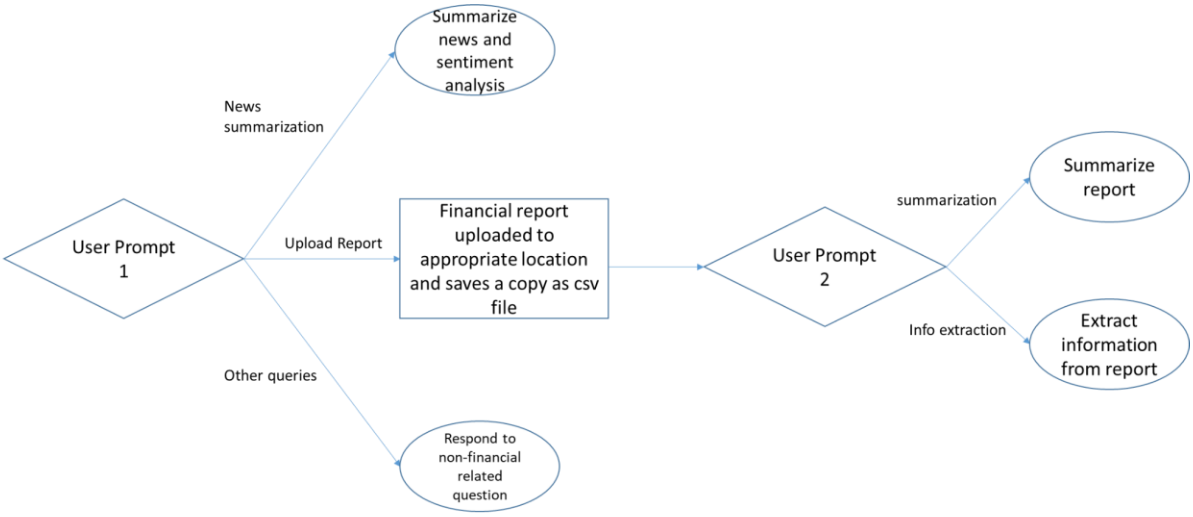

Summarize news and sentiment analysis
News summarization
User Prompt 1
Upload Report
Financial report uploaded to appropriate location and saves a copy as csv file
Other queries
Respond to non-financial related question
User Prompt 2
summarization
Summarize report
Info extraction
Extract information from report

Figure 2: Workflow

1. **User Prompt 1:**

The user starts off by either uploading a financial report then tell the chatbot, or can straight up ask the chatbot for news summarization with sentiment analysis, or a question outside of the financial report scope.

2. **User Prompt 2:**

The continuation of uploading the financial report allows user to ask for a summarization of the financial report or extract specific information available from the report.

3.3 Technical Stack and Architecture

3.3.1 Frontend:

- **Technologies:** Streamlit
- **Interface:** User interfaces are presented as web pages utilizing Streamlit, an open-source Python framework that is suitable for machine learning or data science practitioners.
- **Interaction:** The frontend communicates with the backend via functions imported from other Python files.

3.3.2 Backend:

- **Technologies:** Python, Huggingface, Langchain
- **NLP Engine:** Processes user inputs and manages chatbot interactions.
- **Retrieval Augmented Generation (RAG):** LLM is Gemini with Wikipedia pages as documentation

3.3.3 Database:

- **PDF File:** Store financial reports in a compiled folder
- **CSV File:** Convert financial reports to csv file to be stored in designated folders. Also store chat history and news as csv files.
- **Pickle File:** Convert financial report to vector embeddings as pickle file

○ **Binary File:** Store Llama2 model's weights as binary file

3.3.4 Security:

○ **API Security:** Utilizes token from Gemini to secure API calls.

3.4 Data Management

3.4.1 Data Sources

10K Reports:

A dataset comprising 132 numbers of 10K reports (obtained from around 44 companies with an average of 3 reports between 2021 and 2023) served as the foundational training data for our summarization and information extraction task on these reports. This diverse dataset provided a comprehensive representation of corporate financial disclosures and operational insights, facilitating robust model training and enhancing the accuracy and effectiveness of our summarization algorithms.

News:

News articles related to 47 companies were extracted over four time periods. GoogleNews library was utilized to search for relevant articles and the newspaper3k library to extract the text content. The extracted news articles form the training data for subsequent model training for text summarization.

Companies:	47 (refer to Appendix A)
Time periods:	January 5-10, 2024
	February 15-20, 2024
	March 10-15, 2024
	April 1-5, 2024
Articles per company per time period	30-40

The number of successfully extracted articles per company/period varies depending on news volume. The text extraction process might fail for some URLs due to access issues or variations in website structure.

3.4.2 Data Preparation

10K reports:

In the initial phase of data preparation, the conversion of PDF documents to text format was pivotal. After thorough exploration of various Python packages, pdfplumber emerged as the optimal solution for this task, providing the necessary compatibility and accuracy crucial for subsequent processing in the Language Model (LLM).

Given the multipart structure of 10K reports, a strategy was devised to segment them into their distinct items. This approach aimed to manage text length effectively and facilitate the generation of meaningful summaries tailored to each section rather than cutting across sections. By systematically analyzing patterns observed across all reports, a pattern search mechanism was established. This enabled consistent identification, splitting, and storage of the respective sections into JSON files for future processing. However, there were some reports which deviated from the observed patterns and had to be excluded from the training dataset. These exceptions typically included reports containing images or employing different labelling conventions compared to the majority of other companies' 10K reports.

Anticipating the summarization task, the establishment of a gold standard summary became imperative. While evaluating various options, including GPT4 and Gemini, the decision was made to leverage Gemini for generating the target summaries. The generated summary then served as the benchmark for training datasets for subsequent summarization tasks.

News:

Similar news (near-duplicates) is an issue when extracting news due to fast-paced news propagation. When a story breaks, it is replicated almost identically to a number of other content providers. A rough estimate indicates that 30–40% of news stories are reprints. Removing these near duplicates is important to provide users with a more concise and diverse collection of news articles. It is also essential to remove the duplicated content to increase the quality of the training data for subsequent processing for model training.

The script finds similar news articles within a dataset based on their semantic meaning and helps to identify new or significantly different news articles from a stream of news data. The get_vector(sentence) function first defines functions to get sentence embeddings using a pre-trained sentence transformer model ("all-mpnet-base-v2") and saves the data into a data frame. The mean_pooling function converts a sequence of token embeddings into a single sentence embedding that captures the overall meaning of the news article. This sentence embedding is then used for similarity comparisons with other news articles to identify redundant content.

The get_similarity function takes a dataframe of news articles and calculates their embedding vectors. It iterates through the articles, comparing each news article's embedding vector with previous articles within 14 days using cosine similarity. If an article has low similarity (cosine similarity less than a threshold) to previous recent articles, it's considered unique and kept in the resulting data frame.

For the information extraction task, to optimize the processing of these articles and enhance the effectiveness of downstream analysis, an important step involved segmenting the articles into more manageable chunk sizes. This segmentation enhances computational efficiency by focusing on specific sections within articles, thus streamlining the extraction process.

Annual Report:

The preparation process for the annual report summarization model involved the collection and preparation of a substantial dataset from over 100 annual reports, resulting in a comprehensive corpus exceeding 160,000 words. This dataset was pivotal for training our summarization models. To build a robust training set, we manually extracted more than 1000 article-observation pairs from these reports. This process was essential to capture the diversity and complexity of information typically found in annual reports, which include varied financial statements and strategic disclosures crucial for informed business analysis and decision-making.

Initially, the data extraction process was performed manually. This method, while thorough, proved to be exceedingly time-consuming and labor-intensive, particularly given the voluminous nature of the data involved. Manual extraction, though meticulous, highlighted the need for a more streamlined approach to handle large volumes of text without compromising on the quality of the extracted data.

To address the inefficiencies of the manual extraction process, we implemented Robotic Process Automation (RPA). This technology facilitated the automation of repetitive tasks involved in data extraction, significantly enhancing our operational efficiency. Further streamlining was achieved through the development of a custom macro, utilizing tools akin to those available on G-Hub and similar functionalities as seen in AutoHotKey automation techniques from our coursework. The macro was designed to automate several steps: copying text from PDFs, pasting this text into Excel in the next available cell, and then returning to the PDF to continue the process. This automation markedly improved the speed and consistency of extracting over 1000 entries from the annual reports, allowing for quicker compilation of the dataset with reduced human error.

To establish a gold standard for our summarization task, we developed a Gemini summarization notebook. This tool was engineered to process each entry from the expansive dataset, extracting snippets from the annual reports and automatically generating concise summaries. These summaries were then placed alongside the original excerpts in the parallel column of an Excel spreadsheet. This procedure not only streamlined the summarization process but also emphasized the extraction of key business metrics relevant to strategic consulting. The resulting pairs of text and summary prepared the groundwork for subsequent model training, ensuring the model was fine-tuned to meet the specific analytical needs of business consultants.

3.4.3 Dataset Generation for LLAMA2 Model Training

The dataset for training llama2 models was curated using a novel approach leveraging the Gemini API to generate paraphrased questions from a set of seed questions. This method ensured a robust dataset that mimics a wide range of natural language variations encountered in real-world queries. Each seed question was input into the API, which then produced multiple paraphrased versions, enriching the dataset with diverse phrasing and lexical variety. This enhanced dataset contributes significantly to the training process, enabling our models to better understand and respond to nuanced user queries with greater accuracy and reliability.

4. SYSTEM DEVELOPMENT AND IMPLEMENTATION

4.1 Tools / Techniques

4.1.1 10K Reports

Overview:

T5 Model Fine-Tuning for Text Summarization

Introduction

The Text-to-Text Transfer Transformer (T5) model is a flexible framework designed to handle a variety of text-based tasks using a unified approach. Developed by Google, T5 converts all text-based tasks into a text-to-text format, allowing it to perform tasks such as translation, summarization, and question answering using the same model architecture. This report delves into the specifics of using T5 for fine-tuning on a summarization task involving 10,000 text-summary pairs.

Experimentation with Model Variants and Hyperparameters

In our experiments, we focused on two main variants of the T5 model: T5-Base and T5-Large. These models differ primarily in their size, capacity, and expected performance on complex tasks, with T5-Large being significantly larger and more computationally demanding.

Model Configurations:

- T5-Base: Configured with a moderate number of parameters, making it suitable for tasks requiring less computational power.
- T5-Large: Offers more capacity and is better suited for more complex summarization tasks, at the cost of increased computational requirements.

Hyperparameters:

- Batch Size: Varied to understand its impact on model performance and training efficiency.
- Max Length of Input: Adjusted to cater to average length of documents in the dataset.
- Summary Length: Set according to the desired length of output summaries to maintain a balance between detail and brevity.

Fine-Tuning Process

The fine-tuning process involved adjusting the model to specifically cater to the summarization task. This was done by prefixing input data with "summarize:" to explicitly prompt the model to generate summaries. The models were trained using different combinations of hyperparameters to find the optimal settings for the best summarization output.

Model Name	Epochs	Learning Rate	Max Length	Summary Length	Bleu	Rouge 1	Rouge 2	Rouge L	Rouge L Sum
T5-base	5	1.00E-04	512	150	0.2843	0.5167	0.3223	0.4174	0.4178
T5-base	10	1.00E-04	512	150	0.3018	0.5703	0.3744	0.4657	0.4655
T5-base	5	1.00E-04	256	256	0.2312	0.5413	0.3503	0.4392	0.4398
T5-base	10	1.00E-04	256	256	0.2382	0.5925	0.3956	0.4888	0.4898
T5-large	4	1.00E-04	512	256	0.2843	0.5167	0.3223	0.4174	0.4178
T5-large	10	1.00E-04	512	256	0.2616	0.6209	0.4179	0.52	0.5197

Other Models Explored for Summarization Task

Besides T5 model, GPT2-Summarizer, fine-tuning of GPT2-small, GPT2-medium and BART was also explored. However, the performance of these models did not match that of the T5 model, as indicated by the Rouge scores and qualitative assessment of the generated summaries. Specifically, while BART showed promising results, the outputs from the fine-tuned GPT2 models exhibited a lack of coherence, particularly towards the latter part of the generated text.

For the fine-tuned GPT2 models, GPT2Tokenizer was used to tokenize the text and create input sequences for the model training. AdamW optimizer and CrossEntropyLoss loss function were utilized for the training process. Both small and medium-sized GPT2 pretrained models were explored with varying numbers of epochs (1, 4, and 10), with the higher epoch GPT2-medium model showing better performance but still which still lacked coherence in general.

To explore how the GPT2 base models (small and medium) compares to the GPT2-Summarizer model, the GPT2-summarizer model was ran without training. The results were much better providing understandable sentence structure but may not have captured the essence of the original text. The better performance is expected as the GPT2-summarizer model had undergone specific training for summarization.

To compare the performance of the GPT2 base models (small and medium) with that of the GPT2-Summarizer model, the GPT2-Summarizer model was ran without any fine-tuning. The results showed a marked improvement in terms of providing understandable sentence structures. However, it was noted that these summaries might not have fully captured the essence of the original text. This enhanced performance can be attributed to the fact that the GPT2-Summarizer model has undergone specific training tailored for the summarization tasks.

Lastly, a BART-based pretrained model was explored using the BART tokenizer. The generated summaries showed significant improvement compared to those from the GPT2 models, further highlighting the efficacy of BART for text summarization tasks.

Model Name	Epochs	Learning Rate	Rouge 1	Rouge 2	Rouge L	Rouge L Sum
GPT2_Small	1	1.00E-04	0.06496	0.004792	0.05722	0.06613
GPT2_Small	4	1.00E-04	0.09658	0.009822	0.08103	0.09385
GPT2_Small	10	1.00E-04	0.1071	0.01403	0.0921	0.5042
GPT2_Medium	4	1.00E-04	0.05618	0.003363	0.05012	0.4102
GPT2_Medium	10	1.00E-04	0.1812	0.03258	0.1397	0.5284
GPT2_Summarizer	0	NA	0.3157	0.1315	0.2062	0.2306
BART	10	1.00E-04	0.2852	0.2195	0.2641	0.2667

4.1.2 News

Overview:

Named Entity Recognition (NER)

A pre-trained model, dslim/bert-base-NER, is used for NER. This model identifies and classifies named entities like organizations within the text. A custom function, get_org(doc), utilizes this model to extract organizations from a provided document (article text).

Sentiment analysis

The vaderSentiment library is used for sentiment analysis. A function, add_sentiment_analysis(df), takes a dataframe containing news articles and performs sentiment analysis on the article text. Sentiment scores are categorized into labels like "Positive", "Neutral", or "Negative" based on a predefined threshold.

This demonstrates potential applications of NLP techniques for news extraction and analysis such as gathering information about specific entities (organizations) and gauging the overall sentiment surrounding them within the news landscape.

Another potential use case of sentiment analysis with regards to organizations is to identify the opinion of a particular organization, for example, the organization's opinion about energy sustainability. This is particularly relevant to news articles where organizations have explicitly expressed opinions about specific topics. However, such use case requires further research considering such opinion search results needs to reflect the natural distribution of positive and negative opinions as quantitative information are critical pieces of information for sentiment analysis.

T5 model fine-tuning

This section explores the effectiveness of creating concise and informative summaries of news articles using different models. A generic model t5-base is selected and trained on a dataset of news articles paired with the corresponding summaries.

Model Name	Epochs	Learning Rate	Max Length	Summary Length	Bleu	Rouge 1	Rouge 2	Rouge L	Rouge L Sum
T5-base	8	1.00E-04	512	150	0.1204	0.4213	0.1686	0.2625	0.2620
T5-base	5	1.00E-04	512	150	0.1179	0.4237	0.1664	0.2597	0.2591
T5-base	2	1.00E-04	512	150	0.1119	0.4173	0.1659	0.2640	0.2639
BART	10	1.00E-04	512	256	0.0900	0.4098	0.1659	0.2532	0.2525
BART	5	1.00E-04	512	256	0.13	0.4261	0.1627	0.2585	0.2579

Information Extraction:
The development of the information extraction model for news reports involved a combination of tools, including Langchain libraries, Hugging Face embeddings, FAISS vector stores, and CTransformers LLMs.

Hugging Face's embeddings, specifically the all-MiniLM-L6-v2 model, is employed to generate contextual embeddings for the text chunks. The generated embeddings were then used to construct a vector store using the FAISS library. FAISS enables efficient similarity search and retrieval, facilitating quick access to relevant text chunks during information extraction.

The Langchain library facilitated the integration of language models (LLMs) for natural language processing tasks. In this case, the CTransformers library was used to load a pre-trained LLM, specifically the "llama-2-7b-chat.ggmlv3.q2_K" model.

To enable context-based question answering, a RetrievalQA chain was constructed using the Langchain library. Upon receiving a query, the model processed the question and retrieved relevant information from the vector store. The effectiveness of the model was demonstrated through testing with a series of queries, where the model consistently generated correct answers.

4.1.3 Annual Reports

Modelling Process

Within the annual report summarization section of the SemanticForce product, the Text-to-Text Transfer Transformer (T5) was selected as the primary modeling tool due to its exceptional capabilities in handling diverse text-based tasks. The decision to utilize T5 was influenced by its high flexibility in adapting to various text lengths and complexities, making it particularly effective for the extensive and varied content of annual reports. Furthermore, T5's pre-training on a vast corpus has equipped it with a superior understanding of context, a critical feature for interpreting the nuanced language of financial narratives commonly found in these reports.

To tailor the T5 model to our specific needs, an extensive parameter tuning process was undertaken. The optimal performance was achieved using the T5-base model configured with the following parameters: a learning rate of 1e-4 to ensure gradual and stable learning, max length and summary length both set at 256 to balance detail capture with computational efficiency, and a training duration set to 10 epochs to allow thorough learning without overfitting. Consistency across training sessions was maintained by fixing the seed, ensuring comparability of results.

Performance evaluation relied on BLEU and ROUGE scores, standard metrics for assessing text summarization quality. Our findings indicated that variations in learning rate and max length did not significantly impact the BLEU score, suggesting robustness in the model's performance across these parameters. Notably, the T5-base model excelled in the ROUGE-1 and ROUGE-2 scores, demonstrating its effectiveness in capturing keywords and phrases. This capability was complemented by acceptable accuracy in the comprehensive ROUGE-L and ROUGE-L Sum scores, confirming the model's proficiency in producing coherent and contextually accurate summaries.

Product Development and Deployment

Following the successful modeling phase, the focus shifted towards the deployment and integration of the summarization tool to ensure it was accessible and functional for end-users in strategic consulting environments. The model, optimized for accuracy and efficiency, was deployed on Hugging Face, a platform that facilitates easy sharing and integration of machine learning models. This cloud-based deployment allows the model to be readily available for use across various devices and platforms, significantly enhancing its accessibility and practical utility.

Concurrent with the deployment, an inference script was developed to serve as the interface between the model and its users. This script was meticulously integrated with a custom PDF parser, designed specifically for this project. The parser's role is to preprocess the input PDFs, converting them into a format suitable for summarization by the T5 model. By handling complex document structures effectively, the parser ensures that the textual content is optimally prepared for accurate summarization.

To enable larger annual reports and more robust functionality of the tool, a PDF segmenter was created. This segmenter divides large PDF files into smaller, manageable segments, facilitating a more focused and effective summarization process. Each segment is processed individually by the model, allowing for detailed attention to each part of the document. This segmented approach not only improves the quality of the summaries but also ensures that no critical information is overlooked in large documents.

The summarization process itself is meticulously designed to maintain the coherence and continuity of the final output. After the individual sections of the PDF are summarized, they are reassembled into a single document. This reassembly process is critical as it combines the segmented summaries into a cohesive final report, preserving the flow and narrative structure of the original document.

This comprehensive approach to product development and deployment ensures that the summarization tool is not only technically robust but also user-friendly and highly applicable in real-world consulting scenarios. The integration of advanced NLP tools with practical deployment strategies exemplifies a model of innovation in technology application, aimed at enhancing the efficiency and effectiveness of strategic decision-making in business contexts.

4.1.4 PEFT on LLMs

Overview:

Parameter-efficient fine-tuning techniques such as QLORA (Quantized Low-Rank Adaptation) are crucial for adapting large pre-trained language models like LLaMA-2 to specific tasks efficiently. QLORA combines the benefits of Low-Rank Adaptation (LORA) with quantization, reducing computational overhead and enhancing model adaptability without extensive retraining.

Quantized Low Rank Adaptation (Q-LORA):

QLORA advances LORA by integrating quantization into the fine-tuning process. By focusing changes on low-rank modifications of weight matrices, LORA minimizes trainable parameters, thus mitigating overfitting and computational costs. QLORA extends this by quantizing these matrices, leading to further reductions in memory usage and computational demands, ideal for deployment in limited-resource settings.

Detailed explanation of QLORA:

1. **Decomposition:** The model's weight matrices, especially in the transformer layers' attention and feed-forward networks, are decomposed into low-rank matrices.
2. **Quantization:** These matrices are then quantized to fewer discrete levels, significantly lowering the bit representation.
3. **Training:** Fine-tuning is confined to these quantized matrices, keeping other weights static, which slashes training resources and time.
4. **Integration:** Post-training, these matrices are integrated back without substantially impacting performance.

4.1.5 Chatbot model (LLama2):

Overview:

LLaMA-2 is optimized for natural language processing tasks like text generation, featuring an efficient architecture that balances performance and resource usage.

Architecture:

- **Embedding Layer**: Maps 32,000 vocabulary tokens to 4096-dimensional vectors, with specific settings to exclude padding tokens from gradient updates.
- **Decoder Layers:** Comprise 32 layers with specialized 4-bit quantized attention mechanisms and Low-Rank Adaptations (LoRA) for effective pre-trained model adaptation.
- **Rotary Positional Embeddings and Feed-Forward Network:** Enhance word position comprehension and complex feature extraction.
- **Normalization Layers:** Stabilize deep network training.
- **Output Projection:** Converts decoder outputs to vocabulary predictions.

4.1.6 RAG

Overview

Retrieval Augmented Generation (RAG) lets large language models (LLMs) access and use information from external sources, expanding their knowledge for better responses. When a user asks a question, RAG first finds relevant information and feeds it to the LLM along with the original question. This allows the LLM to give a more informed and accurate answer.

RAG is useful because it enhances LLMs by filling in their knowledge gaps, preventing incorrect answers, keeping them up-to-date with the latest information, and helping them better understand user questions.

Pipeline

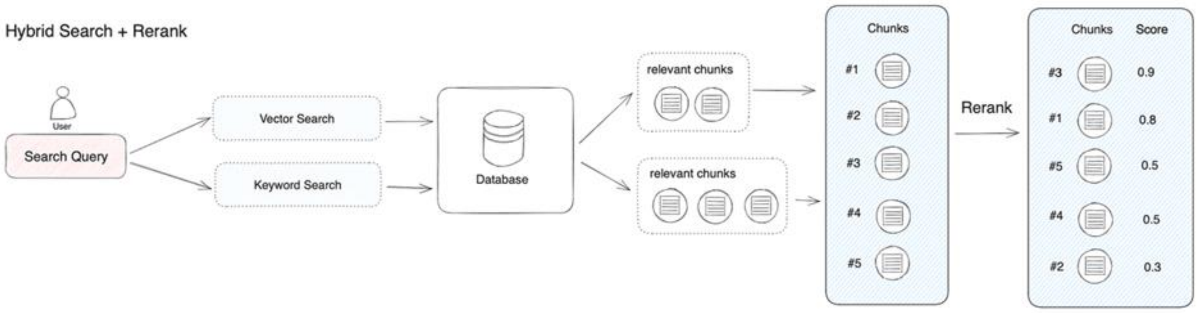


Figure 3: Diagram of Designed RAG Pipeline

Figure 3 above shows the entire pipeline of the designed RAG in this project, whereby it consists of a hybrid search engine and a reranking of chunks of document.

1. **Hybrid Search:** When a user asks a question, the system searches for relevant information in two ways:
   - **Keyword Search (BM25):** Finds text chunks that directly match words from the question.
   - **Semantic Search (Vector Embedding):** The document database is being converted to a vector embedding using Google's embedding model for semantic search. It looks for chunks that are conceptually similar to the question, even if they don't share exact words.
   - **Combined Results:** The system takes the top 5 results from each method, creating a pool of potentially useful information.
2. **Reranking:** Since some retrieved chunks might be more relevant than others, the system uses a cross-encoder to analyze and rank them:
   - **Cross-Encoder:** Understand the relationship between the query and candidate documents by comparing the question and each text chunk, assigning a score based on how well they relate to each other.

    - **Ranked List:** The chunks are then ranked by score, ensuring the most relevant ones rise to the top.
3. **Feeding the LLM:** Only the highest-ranked chunk is passed on to a large language model (in this case, it is Gemini).
    - **Augmentation:** The LLM processes the relevant information and the original question, crafting a final answer that is grounded in facts and sounds natural.

In a nutshell, this pipeline ensures that the language model is only given the most relevant information, increasing the accuracy and quality of the final answer.

As for the documentation, it is made out of Wikipedia pages of the list of companies available from the dataset of both 10K and Annual reports since it is believed that the Wikipedia pages consist of a wide and generic information database of the companies, hence it is believed the documentation has the sufficient data to answer any questions that stray far from financial related information.

Rerank

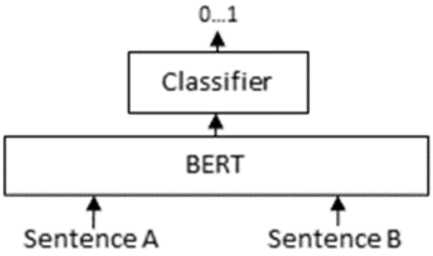


Figure 4: Diagram of Cross-Encoder Reranker

For this project, MS Marco cross-encoder is used as the reranker. Using a BERT architecture, it performs reranking fast, takes up little memory space of around 18MB, yet can attain a high accuracy, hence it has a good balance between accuracy and speed.

MS Marco is a large scale information retrieval corpus that was created based on real user search queries using Bing search engine. The provided models can be used for semantic search, i.e., given keywords / a search phrase / a question, the model will find passages that are relevant for the search query. The training data consists of over 500k examples, while the complete corpus consists of over 8.8 million passages.

RAG Evaluation

RAG Assessment (RAGAs) is a tool designed to analyze the performance of Retrieval Augmented Generation (RAG) pipelines. It provides a set of metrics to assess various aspects of a RAG system's output, considering the process from information retrieval to the accuracy of the generated answer. RAGAs assigns scores between 0 and 1 to provide a comprehensive view of a RAG system's performance. This helps developers understand how well the system finds information, uses it, and generates the final response, ultimately facilitating the refinement of RAG systems for better performance.

The result for using RAGAs to evaluate on the designed RAG can be found in Appendix B. The document used is still the document database created for this project, and Gemini is used to generate the gold standard ground truth. 50 questions were tried on the designed RAG model and only 3 were chosen randomly to show the result in Figure within the Appendix. On average the answer relevancy is 0.89.

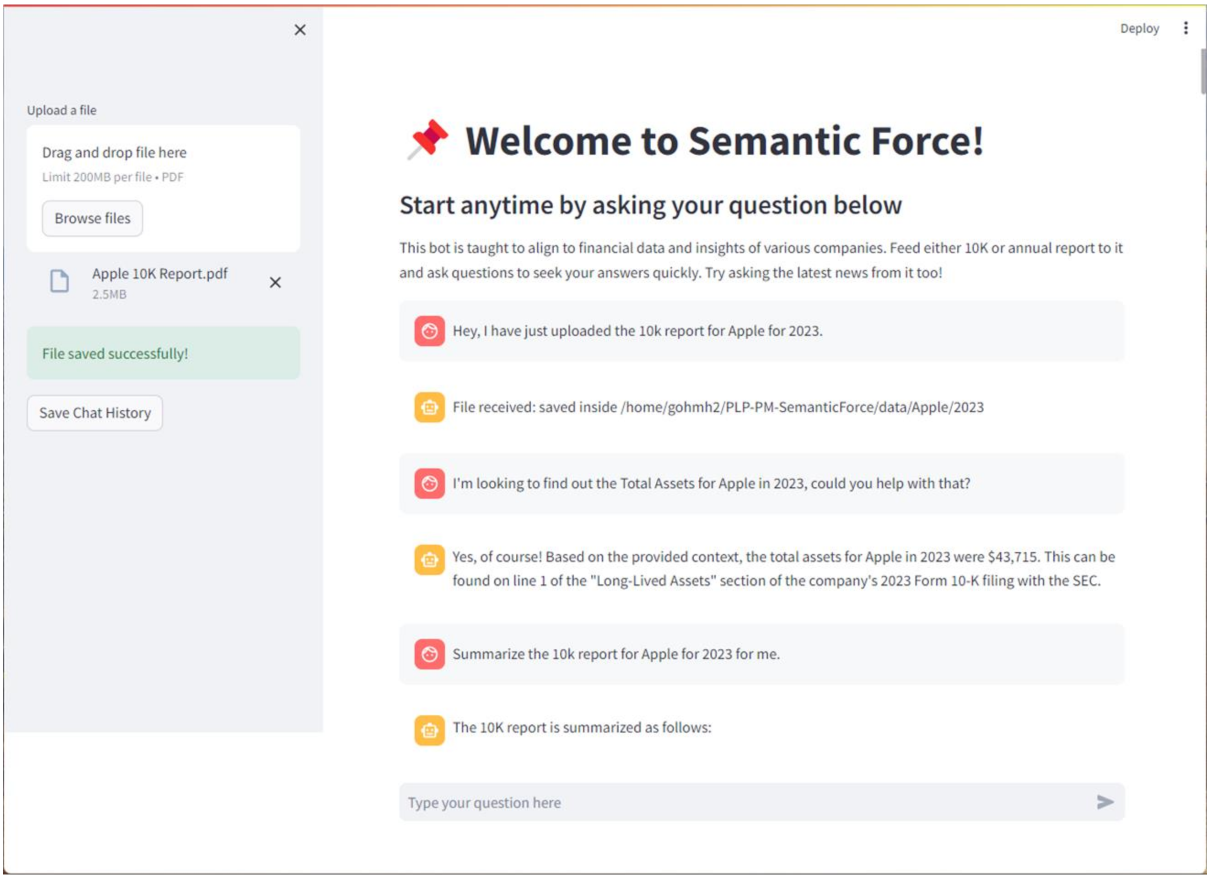
4.1.6 Backend

Python is the language used to code the scripts for preprocessing, training of models, scrapping of news and generating the overall sentiment analysis of user's interested company. Huggingface is utilized to download transformer models and LLMs like Llama2 and T5 and finetune on them. Langchain is another framework utilized for building RAG architecture to support any various queries that the user might have on interested companies.

4.1.7 Frontend and Others

About frontend frameworks

Streamlit is a framework that has been utilized to design the frontend of the project. Streamlit is being chosen because of its simplicity, which means it is fairly easy to deploy.



📌 Welcome to Semantic Force!

Start anytime by asking your question below

This bot is taught to align to financial data and insights of various companies. Feed either 10K or annual report to it and ask questions to seek your answers quickly. Try asking the latest news from it too!

Hey, I have just uploaded the 10k report for Apple for 2023.

File received: saved inside /home/gohmh2/PLP-PM-SemanticForce/data/Apple/2023

I'm looking to find out the Total Assets for Apple in 2023, could you help with that?

Yes, of course! Based on the provided context, the total assets for Apple in 2023 were $43,715. This can be found on line 1 of the "Long-Lived Assets" section of the company's 2023 Form 10-K filing with the SEC.

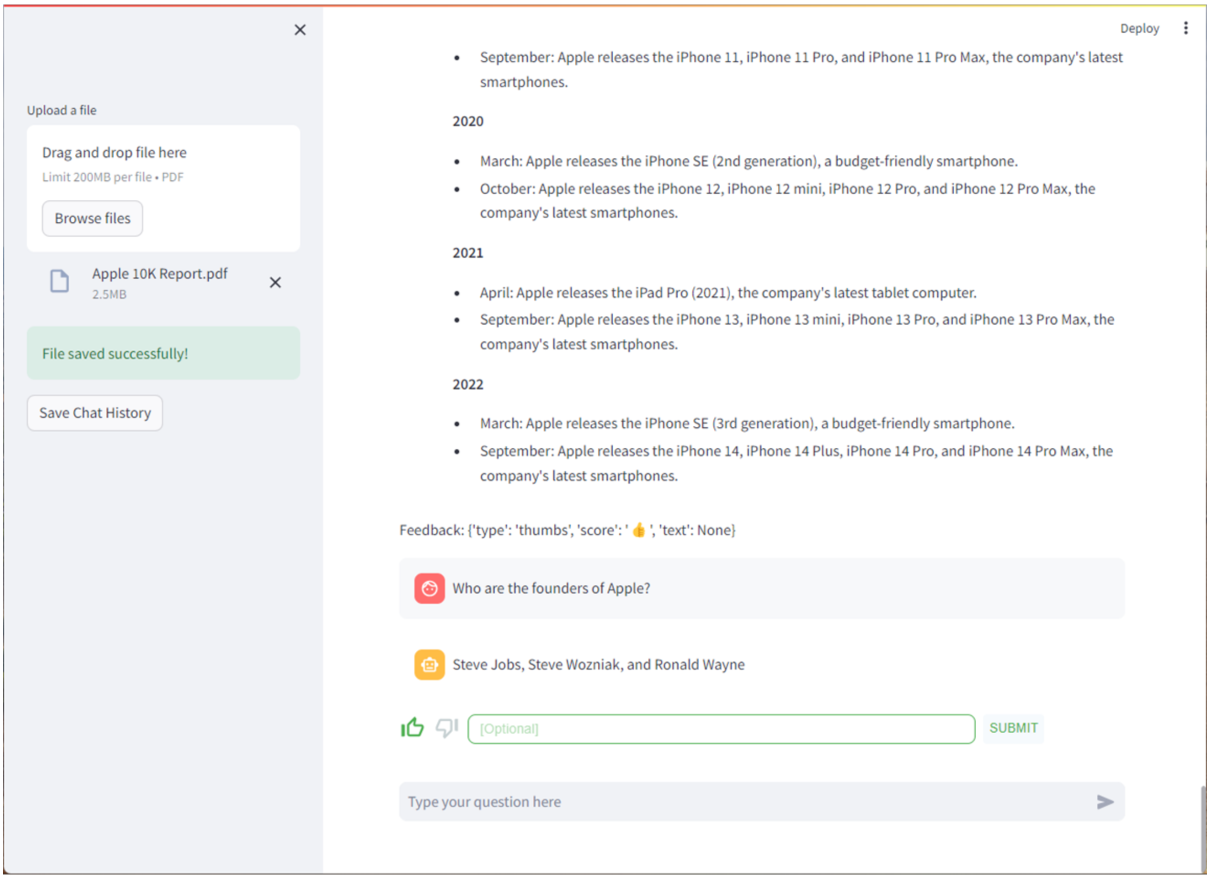
Summarize the 10k report for Apple for 2023 for me.

The 10K report is summarized as follows:

Figure 5: Screenshot of Semantic Force's Webpage

As shown in Figure 5 above, the webpage is similar to that of ChatGPT in which the chatbot interface takes up the entire webpage screen and other functions like uploading pdf files and saving the chat history are included. If the button to save chat history is being pressed without initiating a conversation first, it will prompt the user to first talk to the chatbot.

Upon arriving at the webpage, some instructions are given near the top of the page to guide the user on how to interact with the chatbot. Some actionable items the user can prompt the chatbot to do are uploading a pdf report (to ensure the file is being saved in an appropriate location and converted to csv file), extracting information from said report or summarizing the report. In addition, the user can independently ask questions that might not be financially-relevant to the interested company and ask for the latest news of said company too. For every time the chatbot returns a response to the user, a feedback in the form of ‘thumbs up’ and ‘thumbs down’ will prompt the user to click it, in which the feedback form with question and response pairs will be saved locally. Figure 6 below shows another screenshot of the webpage in which feedback by user is prompted.



- September: Apple releases the iPhone 11, iPhone 11 Pro, and iPhone 11 Pro Max, the company's latest smartphones.

2020

- March: Apple releases the iPhone SE (2nd generation), a budget-friendly smartphone.
- October: Apple releases the iPhone 12, iPhone 12 mini, iPhone 12 Pro, and iPhone 12 Pro Max, the company's latest smartphones.

2021

- April: Apple releases the iPad Pro (2021), the company's latest tablet computer.
- September: Apple releases the iPhone 13, iPhone 13 mini, iPhone 13 Pro, and iPhone 13 Pro Max, the company's latest smartphones.

2022

- March: Apple releases the iPhone SE (3rd generation), a budget-friendly smartphone.
- September: Apple releases the iPhone 14, iPhone 14 Plus, iPhone 14 Pro, and iPhone 14 Pro Max, the company's latest smartphones.

Feedback: {'type': 'thumbs', 'score': '👍 ', 'text': None}

Who are the founders of Apple?

Steve Jobs, Steve Wozniak, and Ronald Wayne

Figure 6: Screenshot of Semantic Force’s Webpage with Feedback Prompt

Overall, the simplicity of the web page allows the user to understand the instructions directly and makes it intuitive for the user to interact with.

About third-party authentication

API token from Gemini needs to be initiated on the user’s side in order for RAG, more specifically the loading of Gemini LLM, in order to fully work.

4.2 Challenges and Solutions

4.2.1 ChatGPT or other LLM usage during Project

Specifically on project outcomes, LLMs like T5 and BART were used for summarizing financial documents and news articles, utilizing their strengths in transforming complex text into concise summaries. The Llama2 model was specifically fine-tuned for enhancing user interactions within the chatbot by recognizing user intents accurately. The performances can be referenced from their respective technical sections above. Additionally, the Gemini model was used to create gold standard summaries, setting the benchmark for training other models.

These LLMs collectively streamline the project's capabilities in document summarization, sentiment analysis, and interactive question answering.

During the course of work, ChatGPT was helpful in troubleshooting code errors in general. However, its assistance was found to be limited when it came to resolving issues related to the deployment of recent models like Gemini. While it could efficiently address common errors when prompted appropriately, it struggled to provide effective solutions for newer and more complex issues that it had seemingly not encountered before. In such situations, turning to forums for guidance proved to be a more fruitful approach for resolution.

4.2.2 Nature of User Prompt

User prompts like telling the chatbot that a financial report is being uploaded can be reduced in order to make the user interface more intuitive. Since the user prompt of telling the chatbot that a financial report is being uploaded is to save the report into the appropriate location and conversion to csv file, these functions should be transferred to actionable items once the file is being uploaded already.

4.2.3 High Amount of Resource for System Requirement

In order to run the webpage, at least 16GB of RAM is required to run the models at the backend smoothly. Despite that, the models still take at least a minute before generating a response.

5. DISCUSSION (NEXT STEPS)

5.1 Roadmap for Future

5.1.1 Future Considerations

As the system is currently positioned as a Minimum Viable Product (MVP), it is essential to explore further enhancements to enhance its functionality and user experience. Providing customizable and personalized features within SemanticForce will cater to the diverse needs of users. This could include allowing users to tailor analysis pipelines, customize dashboards, or define specific criteria for data extraction and analysis. This will reduce the need for repeated prompts in the system making it a much more efficient experience for the user.

Additionally, an issue encountered during the project was with the extraction of information from images within the reports. In future once the Optical Character Recognition (OCR) technology matures, integrating SemanticForce with OCR capabilities to extract and analyze text from images would enable users to analyze textual content from a wider range of sources, including scanned documents.

Moreover, the feedback system in the webpage could be used to train a reward model via Reinforcement Learning with Human Feedback (RLHF) to enhance the chatbot’s response. While the chatbot will claim that it does not know the answer instead of hallucinating a wrong answer, meaning model hallucination is not an issue in this case, the reward model will teach the chatbot not to answer controversially such that it might produce its own opinions.

Lastly, besides 10K reports, news, annual reports, SemanticForce could be further augmented to accept other data sources such as market research reports and social media platforms. These would provide a more holistic source of information and offer a more comprehensive view leading to better insights for informed decision-making.

6. CONCLUSIONS

The developed minimal viable product (MVP), SemanticForce, has successfully demonstrated its ability to achieve our defined outcomes. It successfully extracts key business metrics from various user-submitted reports, including annual reports, 10-K filings, and news articles. SemanticForce offers text summarization capabilities, condensing complex reports into concise overviews. It also analyzes sentiment from relevant news, providing valuable insights into public perception. Additionally, its Q&A functionality allows users to delve deeper and gain specific insights from the processed data. Finally, the product validates its practical utility in live consulting scenarios through a conversational user interface.

Looking ahead, the platform can be integrated with existing business intelligence processes, creating a seamless workflow. Beyond its current capabilities, future development can introduce features such as industry-specific metrics extraction and enhanced visualization tools.

In essence, SemanticForce fulfills the initial goal of creating a customized and holistic solution for business report analysis. Its proven success and planned advancements position it as a powerful platform ripe for scalability and significant future growth.

APPENDIX A - COMPANIES

1. Amazon
2. Advanced Micro Devices
3. Apple
4. AT&T
5. AbbVie
6. 3M Company
7. Bank of America Corporation
8. Caterpillar Inc
9. Chevron Corporation
10. Comcast Corporate
11. ConocoPhillips
12. Delta Air Lines
13. eBay Inc
14. Exxon Mobil Corporation
15. Ford Motor Company
16. General Electric Company
17. General Motors Company
18. Halliburton Company
19. Intel Corporation
20. Johnson & Johnson
21. JPMorgan Chase & Co
22. Lockheed Martin Corporation
23. MacDonald’s Corporation
24. Merck & Co Inc
25. Microsoft Corporation
26. Morgan Stanley
27. Nike Inc
28. NVIDIA Corporation
29. Oracle Corporation
30. PepsiCo Inc
31. Pfizer Inc
32. Salesforce
33. Schlumberger Limited
34. Starbucks Corporation
35. T-Mobile US Inc
36. Tesla Inc
37. The Boeing Company
38. The Coca-Cola Company
39. The Goldman Sachs Group
40. The Kraft Heinz Company
41. The Procter & Gamble Company
42. The Walt Disney Company
43. United Airlines Holdings Inc
44. UnitedHealth Group Incorporated
45. Verizon Communications Inc
46. Walmart
47. Wells Fargo & Company

APPENDIX B - TABLE SHOWING RAGAS OUTPUT (A FEW EXAMPLES)

Question	Context	Answer	Ground Truth	Context Precision	Context Recall	Faithfulness	Answer Relevancy
Why has Apple been so successful in the tech industry?	"Apple's focus on design and user experience, its tightly integrated ecosystem of hardware and software, and its powerful brand image have all contributed to its success. The company has also made strategic acquisitions, such as Beats Electronics, that have expanded its offerings."	Apple's success stems from its innovative products, strong brand, strategic acquisitions, and emphasis on design and user experience.	Apple has been successful due to a combination of factors, including innovative products, strong marketing and brand recognition and strategic acquisitions and its focus on design and user experience.	0.87	1.00	0.78	0.93
How has Microsoft's leadership shaped its direction over time?	"Bill Gates' early focus on software development and operating systems established Microsoft as a dominant player. Satya Nadella's leadership has seen a shift towards cloud computing and subscription-based services, positioning the company for a new era in technology."	Key leaders like Bill Gates and Satya Nadella have shaped Microsoft's direction, navigating the company through technological changes and market shifts.	Microsoft's direction has been influenced by the visions and decisions of key leaders. These figures, such as Bill Gates and Satya Nadella, have steered the company through technological shifts, market changes, and internal challenges.	0.76	1.00	0.50	0.83
What were some major turning points in Apple's history, and how did they impact the company's trajectory?	"The return of Steve Jobs in 1997 marked a major turning point for Apple. Jobs refocused the company on innovative products like the iMac, iPod, and iPhone, revitalizing its brand and propelling it to new heights."	Apple's trajectory was altered by major turning points such as product launches, leadership changes, and market events. These led to shifts in strategy and new opportunities or challenges.	Apple faced pivotal moments that reshaped its path. These turning points could include the launch of a groundbreaking product, a change in leadership, a market crisis, or other significant events. These events led to strategic shifts, new opportunities, or significant challenges.	0.73	1.00	0.67	0.77

REFERENCES

LLMS used
Llama2-7b (Chatbot)
https://huggingface.co/meta-llama/Llama-2-7b

News Sentiment Analysis?
https://huggingface.co/sentence-transformers/all-mpnet-base-v2

T5 Base Model (For 10K report summarization, annual report summarization, news summarization)
https://huggingface.co/google-t5/t5-base

Gemini as LLM for RAG
https://gemini.google.com/