,,

Universiteit
Leiden
The Netherlands

Computer Science & Economics

Using LLMs for data-driven business decisions

Ange Mutijima

Supervisors:
Zhaochun Ren & Yumeng Wang

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

16/8/2024

Abstract

This thesis explores how OpenAI's technologies can enhance data-driven decision-making in the face of increasing data volumes. It evaluates the effectiveness of OpenAI's large language models (LLMs) compared to traditional methods in information retrieval and answer generation. The findings suggest that while OpenAI's embeddings do not surpass traditional retrieval techniques in document identification, GPT-3.5 Turbo excels in generating clear and comprehensive answers. However, its reliability as a sole decision-making tool is limited and requires careful validation. Future research should focus on integrating OpenAI's embedding models with other retrieval methods to boost effectiveness and explore their potential for improving information retrieval and summarization processes. Such studies could uncover more efficient ways to leverage these models, potentially revolutionizing data analysis and decision-making strategies.

Contents

1 Introduction

The amount of data organizations generate has grown exponentially. Big Data refers to extremely large and complex datasets characterized by the "Three Vs": Volume, the vast amount of data generated; Velocity, the speed at which this data is created and processed; and Variety, the different types and sources of data (1). Understanding Big Data is essential for organizations to maintain their competitive advantage, create business strategies (2), and make decisions effectively (3). By leveraging insights from these vast and varied datasets, organizations can drive innovation, optimize operations, and better meet customer needs.
For instance, data in customer relationship management (CRM) systems can be analyzed to identify sales trends, predict customer behavior, and tailor marketing efforts. Other data, such as social media posts, can provide insights into customer sentiment and brand perception, helping companies to adjust their public relations strategies.
Research shows that Big Data has disrupted the decision-making at the board-level management of organizations (4). However, the same research also suggests board members and directors do not possess the capabilities to deal with Big Data which can negatively impact the decisions made.
Powerful technologies are required to make the Big Data interpretable. Recent developments have introduced new possibilities for data analysis. Generative Artificial Intelligence (GAI), such as Large Language Models (LLM), has grown exponentially in recent years. The disruptive development is rapidly changing various industries, including business. The rise of GAI has sparked the need for research on its possibilities and effects on business processes. One comprehensive research (5) has made an overview of how LLM are currently employed in marketing, customer service, finance, and many more business processes. The majority of the tasks relate to automation, personalization, and assistance.
Currently, data scientists are responsible for transforming Big Data into valuable insights. This task requires a multidisciplinary skill set, including expertise in Python and SQL (6). The work of decision makers is therefore at a standstill until the data report is provided which might cost valuable time and money. But what if decision-makers could extract valuable insight from the data themselves without learning statistics or programming skills?
Organizations accumulate a large volume of financial reports each year, all containing valuable information. Navigating this vast landscape to find the right data source can be challenging. Furthermore, analyzing these diverse data sources is often time-consuming and may require specialized expertise. This research aims to use LLMs, specifically OpenAI's technologies, to simplify business analysis so that decision-makers can easily access key insights from data. Examples of questions that this study will focus on include "what is the net change in net revenue during 2015?" or "what was the percentage change in rental expense for operating leases from 2015 to 2016?". This way, decision makers will have direct access to the information they need which could save money and time. In addition, the feasibility of eliminating data scientists from data-driven decision-making will be explored. The saved resources during decision-making could be redistributed to data-driven research to provide decision-makers with innovative and futuristic insights.
The specific research questions are the following:

1. How can LLMs be used to simplify the statistical analysis needed for data-driven

business decisions?

2. To what extent can LLMs make statistical analysis accessible in business to those with limited knowledge of statistics?

3. How can the performance of an LLM be examined in a Question Answering (QA) use case?

Firstly, previous research will be discussed in the Related Works section (Section 2). This will include an overview of the evolution leading up to the current state of Large Language Models (LLMs), providing insights into the developments that preceded this technology. The focus will then shift to the reasoning capabilities of LLMs, exploring their proficiency in logic and how to fully exploit these abilities to enhance performance. Additionally, we will examine the application of LLMs in finance and their capability to perform mathematical operations. Following the discussion of related works, the experimental setup will be explained in the Experiment section (Section 3). To research how LLMs can simplify business analysis, traditional information retrieval and text generation methods will be compared to OpenAI's equivalents. A novel additional step during the information retrieval will be introduced which aims to increase accuracy. The proposed information retrieval method will be tested and compared to sparse retrieval techniques. To what extent LLMs can make statistical analysis accessible will be assessed on their ability to select the most relevant documents and answer financial questions. The research will also assess the performance of the language models using standard text generation metrics alongside OpenAI's GPT-3.5 Turbo, to determine the quality of the answers and to see which approach provides a more accurate picture of their effectiveness.
The results of the experiment will be discussed in the Results section (Section 5). While OpenAI's embeddings and information retrieval architecture seem to perform worse than traditional methods, their question-answering abilities demonstrate substantial improvements compared to the established T5 language model by Google AI.
To conclude the research, future research directions will be proposed to further explore the potential of the methods proposed in this study.

2 Related Work

2.1 The Evolution of Large Language Models

Throughout the years, humans have gone to great lengths to speak the language of computers. By creating and mastering programming languages, the power of technology can be effectively leveraged. However, AI has seen rapid developments in the last few years that have caused a switch; computers are starting to speak our language. Large Language Models (LLMs) have seen tremendous growth and can speak and understand our natural language (7). Leveraging the power of technology no longer requires programming knowledge with the rise of LLMs. While it might seem like an overnight development, the history of current LLMs is long and rich.

For any models to understand a language, they must be able to create a Language Model (8). Language Modeling (LM) is crucial for Natural Language Processing (NLP) tasks, which are tasks that converse written and spoken natural languages into structured data (9). With the help of a large text corpus, models learn the patterns and structure of natural languages. The process of Language Modeling enables the models to understand and generate text. Since the 1980s there have been 4 major developments in Language Modeling.

2.1.1 Statistical Language Models

The first major development was the creation of Statistical Language Models (SLMs) in the 1980s. Before the 1980s, Language Models were rule-based and relied on their creators to provide them with a set of well-defined rules to understand and generate language (10). SLMs completely changed that and were the first language models that learned from a text corpus. From this large text corpus, SLMs learn probability distributions for the next word based on the previous words using statistical estimations (11).
SLMs were originally intended for speech recognition; spoken words could be recognized more accurately when considering the previous words. However, due to its flexible nature, it was applied to various other NLP tasks such as machine translation and information retrieval.
The performance of SLMs greatly depended on the amount of available training data. The rise of the internet happened alongside the rise of SLMs. This meant that there was plenty of textual data available to train the models with. This resulted in significant improvements in their performance. However, after 20 years it seemed like SLMs had reached their full potential. No amount of extra data could improve their performance.
SLMs simply learn the order in which words occur to give a probability distribution. A limitation of this type of modeling is that it does not take language into account. The probability distribution can be made without actual understanding of the human language. The same model would work on an arbitrary language with arbitrary symbols (12). The need to develop Language Models that understood human language grew.
Since SLMs use the preceding words in a sequence as parameters to determine the probability distribution of the next word, the number of parameters can grow significantly. This issue is known as the curse of dimensionality and especially becomes an issue when generating large pieces of text. N-gram models became the new state-of-the-art model. While traditional SLMs used all preceding words in a sequence, N-grams only used the preceding N words to make a probability distribution. The N-grams served as approximation methods for SLMs and were therefore widely used (13).

2.1.2 Neural Network Language Models

Neural Network Language Models (NLMs) were developed to address the curse of dimensionality. NLMs use neural networks to learn distributed representations of words (14). The distributed representations, also known as embeddings, represent words as vectors. These embeddings can capture the meaning and semantic relationship between words. The embeddings are continuous vectors that are updated during training to reflect their meaning and usage in the different contexts the model encounters. It aims to capture words in a way where a

closer distance in the vector space indicates similarity between the words. This enables more understanding and linguistic knowledge.
Unlike SLMs, NLMs are able to predict words in sequences even if those sequences were not seen during training. NLMs can use similar word contexts it has encountered instead. Because of their effective distributed representations, NLMs use fewer parameters than SLMs and generalize well. This makes them appropriate for larger datasets and more complex tasks as well.
To predict the probability distribution of the next word, NLMs transform the preceding words into a sequence of feature vectors. These feature vectors represent the semantic meaning of the words in the sequence. Essentially, the model predicts what word is likely to follow based on the context provided by the sequence.

2.1.3 Pre-trained Language Models

The introduction of Pre-trained Language Models (PLMs) marked a significant advancement in NLP tasks. PMLs are context-aware; they capture the meaning of individual words as well as the context in which those words appear. PMLs are trained on vast amounts of text data, enabling them to understand and predict language more accurately (15).
PMLs ability to capture nuanced language features has made them highly effective in understanding the broader context and subtleties of natural language. They demonstrated superior performance in NLP tasks such as sentiment analysis and text classification (16).
Despite their impressive linguistic understanding, PLMs had a few limitations. They possessed limited knowledge which made them unable to perform various NLP tasks. The models must be fine-tuned before they can be used for various tasks (17). The need for further fine-tuning meant that, while powerful, PLMs were not immediately applicable for widespread use across diverse tasks.

2.1.4 Large Language Models

This brings us to the current landscape of Language Models. Large Language Models (LLMs) were developed to achieve exceptional performance across a wide range of NLP tasks (12). LLMs are trained on a massive amount of data. It is due to this extensive training that LLMs possess so such versatile capabilities and knowledge.
LLMs are also highly adaptable. They can be fine-tuned to excel in particular applications (12). Its adaptability offers great opportunities for highly customized utilization, allowing organizations and researchers to tailor these models to meet specific needs and objectives.
The evolution of language models continues to drive innovation in NLP, promising even more sophisticated and context-aware AI systems. These advancements are further bridging the gap between human and computer communication.

2.2 Exploring the Reasoning Capabilities of LLMs

Fundamentally, LLMs generate text by predicting the next word in a sentence. According to the Cambridge Dictionary, intelligence is "the ability to learn, understand and make judgments or have opinions that are based on reason" (18). While LLMs learn from a vast amount of

data and understand the relationships between words, the question remains whether they simply replicate patterns found in their training data or if they engage in logical reasoning. This distinction is crucial in evaluating the true capabilities and limitations of LLMs as it delves into the deeper issue of whether these models demonstrate genuine understanding and cognitive processes similar to human intelligence.

2.2.1 The Fundamental Logic of LLMs

Research has explored whether the reasoning and judgments of LLMs stem from genuine intelligence or merely from the numerical patterns of words and their relationship (19). Although LLMs perform well on complex reasoning tasks, they can easily be misled and often struggle to stand their ground when challenged with flawed or nonsensical arguments. This suggests that LLMs still lack a deep understanding of logic and language. Their responses seem more reflective of statistical correlations learned from the data rather than true cognitive reasoning, highlighting the gap between human-like intelligence and AI capabilities.

2.2.2 Techniques to Optimize Reasoning

While it is still unclear whether LLMs possess genuine intelligence, they perform quite well on complex reasoning tasks [?]. Drawing inspiration from human practices where knowledge is effectively gathered through debates, researchers have explored the concept of multi-agent debates among LLMs (20). This approach allows LLMs to learn from each other, improving their ability to handle tasks and capitalize on each other's strengths. By engaging in debates, LLMs can enhance their performance across various tasks. However, research indicates that comparable results can be achieved using a single LLM with a well-designed prompt which is known as prompt engineering

Prompt engineering is an effective approach to enhance the performance of LLMs (21). It involves carefully crafting the input prompts to provide clear context and guidance, helping the LLM better understand the task. Prompt engineering can significantly improve the model's ability to process complex information and produce more accurate responses by supplying the necessary context and information for better reasoning.

Few-shot learning is a prompt engineering method where a model is provided with a small number of labeled examples that serve as instances or demonstrations of the task at hand (22). These examples guide the model to improve its performance on similar tasks. Few-shot learning aims to provide task-specific customization similar to fine-tuning but with reduced computational costs and minimal data requirements (23). This approach makes models tailored for specific tasks more accessible compared to traditional fine-tuning methods. Studies have demonstrated that few-shot models can achieve performance comparable to fine-tuned models, even without altering the model architecture itself (24). However, the effectiveness of few-shot learning depends on the similarity between the provided examples and the target task (25). If the new task significantly differs from the examples used during few-shot learning, the model's performance may worsen due to its limited adaptability and flexibility to handle slightly different tasks. While effective in specific scenarios, few-shot learning may struggle to generalize to tasks that significantly deviate from its training examples.

Conversely, zero-shot learning, another prompt engineering method, operates without relying on labeled examples. Instead, it utilizes the foundational knowledge ingrained in the model during its training phase. Zero-shot learning tasks the model with applying its learned understanding to novel tasks it has not been explicitly trained on, deriving responses based on its existing knowledge and contextual comprehension (26). Therefore, while few-shot learning refines the model with specific examples, zero-shot learning relies on the model's ability to generalize and utilize its acquired knowledge autonomously.
Due to its adaptable nature, zero-shot learning is typically employed for tasks that span a wide range or evolve (27). However, zero-shot learning faces challenges when confronted with specific or nuanced tasks (28), which can result in sub-optimal and inconsistent performance for such scenarios.

2.2.3 Advanced Complex Reasoning and Multi-Hop Abilities

Data-driven decision-making often involves gathering insights from multiple sequential reasoning steps. LLMs do not possess the inherent multi-hope reasoning abilities and require specialized training to do so (29). There is growing research on enhancing their performance in addressing queries that involve multi-step reasoning. This includes developing methods that enable LLMs to analyze complex patterns, interpret relationships between those patterns, and generate informed insights. These enhanced capabilities would allow LLMs to be used for more sophisticated and data-informed business strategies.
A survey on multi-hop question answering (MHQA) defines a multi-step reasoning agent as one that derives one or more intermediate conclusions necessary to reach the final answer (30). To assist LLMs in performing complex reasoning tasks, a straightforward yet effective technique is the Chain-of-Thought (CoT) approach. Implementing CoT can be as simple as incorporating prompts like 'Let's think step by step' into the model's input. Despite its simplicity, CoT has been shown to significantly enhance the ability of LLMs to engage in complex reasoning (31).
CoT can also be integrated with few-shot learning by providing the model with examples of CoT reasoning patterns. Research indicates that CoT prompts can achieve high accuracy in solving linguistic math problems, often outperforming models that have been fine-tuned for specific tasks (31). However, its efficacy varies with model size. LLMs with 100 billion or more parameters benefit the most from CoT prompting, demonstrating improved performance and logical coherence in their reasoning processes. Smaller LLMs may struggle with CoT prompting, producing illogical intermediate steps that reduce overall accuracy compared to standard prompting methods. Improper intermediate steps in reasoning could lead to incorrect responses known as hallucinations.
Hallucinations refer to any output of an LLM that deviates from the user's intended query, lacks consistency, or contains inaccuracies (32). They pose a significant challenge for LLMs. To address hallucinations within the context of the CoT process, researchers have introduced the Chain of Question (CoQ) framework (33). This approach breaks down complex questions into multiple sub-questions, each contributing to the overall answer. Instead of generating potentially inaccurate intermediate statements, CoQ focuses on straightforward sub-questions that the model can answer with higher confidence. This approach effectively reduces the occurrence of hallucinations compared to traditional CoT methods.

Furthermore, the CoQ approach aligns with the Chain-of-Verification (CoVe) strategy, known for its effectiveness in reducing hallucinations (34). CoVe involves generating an initial response and posing verification questions to fact-check the response. These questions are answered independently to mitigate bias, ensuring a verified response with greater accuracy. This method highlights the importance of breaking down complex queries into smaller, verifiable sub-questions to lessen hallucinations effectively.
Tree-of-Thought (ToT), building upon the principles of Chain-of-Thought, introduces a strategy aimed at enhancing extensive problem-solving capabilities (35). Inspired by human cognitive processes, ToT initiates a multi-round conversation with the LLM to construct a solution space. The interaction begins with a user prompt, prompting the LLM to generate an initial intermediate step. This step undergoes evaluation for validity and quality. If deemed sufficient, the model proceeds along that path, continuing to generate subsequent intermediate steps similarly. When an intermediate step fails the validity test, the model can backtrack to the last confirmed step and explore alternative paths. This iterative method allows LLMs to explore various ideas while preventing errors from spreading throughout the entire thought process, thereby improving the robustness of the model.
However, the world of LLMs continues to undergo significant developments. Artificial General Intelligence (AGI) refers to artificial intelligence that can perform any intellectual task that a human can do, with the same level of understanding and skill without needing specialized prompting (36). The GPT-4 model from OpenAI has demonstrated exceptional performance across diverse tasks and domains. Early research even suggests that its capabilities are approaching human-level performance and it may even exhibit traits of AGI (37).

2.3 Enriching data with tailored data/tailoring LLMS to specific use cases

LLMs are trained on vast amounts of data, enabling them to acquire understanding and knowledge across various domains and be widely applicable. However, their performance on specific tasks is contingent on the presence of relevant data during training. When an LLM lacks specific knowledge, it cannot perform those tasks adequately. This limitation is particularly problematic when deploying LLMs in unfamiliar environments they were not initially trained for. To address this issue, research has explored numerous methods for integrating custom data into LLMs, thereby enhancing their applicability in new and specialized environments.

2.3.1 Fine-tuning

The first method is fine-tuning an LLM. This process involves taking a pre-trained LLM and providing it with additional data specific to a particular task or domain. By further training the model on this new data, its parameters and weights are adjusted to optimize performance for the given use case (38). Fine-tuned models excel at providing clear and precise responses for specialized tasks and can manage large datasets during the fine-tuning process (39). However, fine-tuned models, like pre-trained ones, are static. To update them with new knowledge, they must undergo retraining. One significant drawback of fine-tuning is its computational expense, making it a costly process. As a result, fine-tuned models can

quickly become outdated in fast-changing environments and may be inaccessible to various organizations and teams due to the high costs involved.

2.3.2 Retrieval Augmented Generation

Another method to provide an LLM with more knowledge is Retrieval Augmented Generation (RAG). RAG enhances the LLM's performance by providing additional context and knowledge. It works by retrieving relevant data sources based on the user input. The retrieved data, combined with the original user query, is fed into the LLM, enabling it to respond with the newly acquired knowledge. Unlike fine-tuning, which alters the LLM model, RAG enriches the user prompt with more context. The data sources are pulled from a dynamic knowledge base that can be updated regularly, unlike the static knowledge base of an LLM. Implementing RAG allows LLMs to possess specific and current knowledge without the expensive computational costs of fine-tuning a model. Research also shows RAG can consistently outperform unsupervised fine-tuned models (40).

Despite the flexibility and cost-efficiency of RAG, the process does have limitations (38). RAG can struggle with semantic search, as its sensitivity to language nuances can negatively impact performance. Additionally, chunking, the process of breaking up the knowledge base into manageable segments, can lead to information loss if not designed optimally (41). These factors can detract from the overall effectiveness of the model.

To effectively manage the library of data sources, it is crucial to store them in an organized manner that optimizes the RAG pipeline. Typically, RAG pipelines use fixed chunk sizes, but this approach may not always be ideal given the varying sizes and nature of data sources. For instance, in financial reports, element-based chunking has been shown to yield better retrieval scores than static chunking strategies (42). In element-based chunking, the creation of new chunks is triggered by new titles and tables, aligning the chunks more closely with the document's structure. This method not only improves retrieval accuracy but also requires fewer chunks, making it a more efficient and optimal technique. Additionally, adapting chunking strategies to the specific nature of the data can enhance the overall performance of the RAG pipeline, ensuring more relevant and accurate responses.

In the context of multi-hop question answering (QnA), research indicates that traditional RAG systems are often inadequate (43). As a result, adapting RAG systems to better support multi-hop QnA has become a significant area of interest. One approach is multi-hop dense retrieval, which iteratively encodes the user query and previously retrieved documents in a vector space to find the next relevant document (44). When introduced, the multi-hop dense retrieval method matched the performance of the best existing methods while being ten times faster, making it a highly efficient alternative. This method involves a continuous process where each step builds upon the previous one, ensuring that the system can handle complex, multi-step queries more effectively. The efficiency and effectiveness of this approach make it a promising solution for enhancing the capabilities of RAG systems in multi-hop QnA scenarios.

There are several RAG strategies available, typically categorized into sparse and dense retrieval methods. Sparse retrieval methods use Bag-of-Words (BOW) vectors to represent text for natural language processing (NLP). In this representation, text is considered an unordered collection of words, and a BOW vector is a sparse vector that records the word

count for every word in a corpus for each text element (45). While this model is simple and can be effective for certain tasks, it has notable downsides, including high dimensionality, extreme sparsity, and an inability to capture the actual meaning of textual data.
To address these limitations, researchers have explored ways to enhance the BOW model by integrating it with newer technologies that can better capture semantic meaning.
This hybrid approach aims to retain the simplicity and computational efficiency of BOW while mitigating its drawbacks by leveraging the strengths of more advanced language models.
Dense retrieval methods utilize word embeddings to enhance information retrieval. By capturing language through dense vector representations, dense retrieval (DR) models can more effectively understand the meaning and semantics of text in fewer dimensions. Word embeddings translate words into continuous vector spaces where semantically similar words are located closer together, thereby improving the model's ability to comprehend and retrieve relevant information. In this approach, both the knowledge base and the query are embedded into a vector space. The system then retrieves the closest knowledge vectors to the query, which are used to generate the answer (46).
The use of dense retrieval methods offers significant advantages over traditional sparse methods. For example, dense vectors can encapsulate nuanced meanings and relationships between words that sparse methods like Bag-of-Words cannot. This results in more accurate and contextually relevant information retrieval, making dense retrieval methods particularly valuable for complex queries requiring a deep understanding of language.
RAG strategies are often evaluated based on the top k passages retrieved, as LLMs can usually only consider a limited number of passages when generating responses. Therefore, ensuring that these top few passages are precise is crucial for the effectiveness of RAG methods (47). A common practice to enhance the quality of these top passages is re-ranking them, which improves the overall results of RAG strategies (48). The retrieval process then becomes a two-step procedure: initial retrieval and subsequent re-ranking. Various approaches to re-ranking exist. A common method involves initially retrieving a large set of documents using simpler, sparse models. These documents are then re-ranked using dense retrieval methods that leverage neural models, which are more adept at capturing the semantic meaning of the text (49). Such hybrid approaches effectively combine the strengths of both sparse and dense retrieval methods; by first casting a wide net with sparse retrieval and then honing in on the most relevant documents with dense re-ranking, these strategies can significantly improve the performance of LLMs in RAG frameworks, making them more capable of handling complex and nuanced queries.

2.4 LLMs in Finance

Large Language Models (LLMs) are revolutionizing finance by leveraging advanced natural language processing capabilities to address complex challenges in financial analysis and decision-making. These models excel in analyzing diverse data sources and performing multi-hop numerical reasoning, essential for tasks such as financial report analysis.
LLMs play a crucial role in enhancing decision-making processes in finance. They provide accurate insights through tasks like financial sentiment analysis, where they achieve high accuracy by uncovering nuanced sentiments in reports and financial statements (50). This

ability significantly reduces the time and effort traditionally required for analyzing financial reports across various firms (51).
Moreover, LLMs contribute to portfolio optimization by providing investors with nuanced risk and reward analyses based on textual data (52). They also aid in market forecasting, complementing quantitative methods with qualitative insights to predict market trends and stock prices (53).
In addition to their prowess in textual data analysis, LLMs leverage their computational power to analyze real-time market data and customer preferences. This capability has led to the development of Robo-advisors, which offer personalized financial advice tailored to individual preferences, including strategies for risk management (54).
One of the key strengths of LLMs is their adaptability to different market environments and client needs. They mitigate human bias in decision-making processes by providing insights based on comprehensive data analysis (55). Fine-tuned LLMs excel in responding to user queries, simplifying complex financial terms, and improving overall question-answering capabilities (56).
In conclusion, LLMs are transforming finance by enhancing analytical capabilities, improving decision-making processes, and providing innovative solutions that adapt to the dynamic nature of financial markets and individual client needs. Their integration marks a significant advancement in leveraging artificial intelligence for strategic and informed financial decision-making.

2.4.1 Arithmic capabilities of LLMs

Large Language Models (LLMs) are designed primarily for natural language understanding and generation rather than mathematical computations, which are crucial in finance. Tasks such as solving math word problems and answering financial questions often involve complex mathematical operations beyond basic arithmetic like addition, subtraction, and simple multiplication. LLMs typically struggle with these tasks, particularly when confronted with large numbers or lengthy mathematical queries (57). Their limitations in handling intricate mathematical computations pose challenges in applications requiring precise numerical analysis and computation-heavy tasks within the finance domain.
To address this limitation, researchers have explored various techniques to enhance the accuracy of LLMs in numerical reasoning tasks. One effective approach involves replacing numerical symbols with their corresponding English expressions. Studies have demonstrated that this method can significantly improve the model's performance when processing numerical information (58). This approach aids LLMs in comprehending and manipulating numerical data within the framework of natural language, aligning it with the data format that the LLM is intended for.
Using zero-shot Chain-of-Thought (CoT) has been demonstrated to enhance results in mathematical reasoning tasks (59). To further refine the accuracy of LLMs in numerical reasoning, Program-of-Thoughts (PoT) complements CoT by employing language models to generate a program encapsulating the necessary reasoning and calculations, which are then executed to derive the answer (60). This approach directs LLMs to produce a program that encodes the required logic, outsourcing the computations to more suitable tools like Python, thereby ensuring precise arithmetic calculations (61). Self-verification of responses

and outcomes also increases mathematical performance (62).
In addition to these methods designed for pre-trained LLMs, fine-tuning can further improve LLMs' mathematical proficiency by training models to generate intermediate steps or refining them with specific mathematical datasets (63).

3 Methodology

The aim of this research is to determine whether large language models (LLMs) can serve as a comprehensive solution for assisting business stakeholders in making data-driven decisions. Specifically, this study will compare traditional data analysis methods with the advanced capabilities offered by OpenAI's embeddings models and the GPT-3.5 Turbo LLM. The effectiveness of these technologies will be evaluated in terms of their capacity and additional value they may provide.
The research process is divided into two main components: information retrieval and response generation. Each of these processes will be tested and analyzed independently to assess their performance and applicability in business contexts.
The methodology is designed to be as flexible and straightforward as possible, accommodating the dynamic environments in which businesses operate. The chosen approaches aim to closely mirror real-life applications, ensuring practical relevance. Moreover, this study emphasizes democratizing data analysis by employing simple and accessible techniques, recognizing that not all stakeholders have access to extensive resources.
OpenAI's technologies, specifically the GPT-3.5 Turbo and their embeddings models, will be utilized in this research. OpenAI has been at the forefront of making AI accessible and practical for a broad audience, making its tools highly appealing for business integration. Given OpenAI's prominence and widespread adoption, its models will serve as the benchmark for new technology in this study.
Fine-tuning models is computationally intensive and lacks the agility needed for frequent updates, which is contrary to the study's objectives. Few-shot learning, while useful, requires regular updates and careful design to cover a wide range of tasks, leading to significant maintenance overhead. Therefore, zero-shot learning is considered the most suitable approach for this experiment, as it relies on robust prompting and can yield satisfactory results without the need for extensive customization.

3.1 Information retrieval

The RAG (Retrieval-Augmented Generation) process will be explored through three distinct methods, each compared against the others. The baseline traditional methods include BM25 and Dense Passage Retrieval (DPR).
BM25 is a well-established ranking function used for information retrieval. It assesses document relevance based on term frequency and inverse document frequency, where the latter measures how common or rare a term is across all documents. By ranking documents according to these criteria, BM25 retrieves the top k most relevant documents for a given search query.
Dense Passage Retrieval (DPR) represents a breakthrough in information retrieval, surpassing the effectiveness of BM25 in certain contexts (64). DPR employs a passage encoder to convert

text passages into dense, real-valued vectors. Simultaneously, it uses another encoder to embed the user's query into the same vector space. The similarity between the query and passages is then computed using the dot product of their vectors, enabling DPR to retrieve the top k passages that are most similar to the query. Leveraging the FAISS library for efficient indexing, DPR can efficiently handle vast amounts of data, facilitating scalable clustering and similarity searches.

In contrast to DPR's focus on passage retrieval, OpenAI's text embeddings specialize in measuring the relatedness between textual elements. These embeddings are versatile, supporting tasks beyond document retrieval, such as classification and recommendation systems. OpenAI employs a unified model for embedding both passages and queries. Utilizing cosine similarity, which calculates the angle between vectors, OpenAI retrieves the passages that are most closely related to the query based on semantic similarity.

OpenAI models are versatile language models capable of handling broad variety of tasks such as text generation, translation, and summarization. In contrast, DPR (Dense Passage Retrieval), developed by Facebook AI, is tailored for retrieval tasks.

It is standard practice to retrieve a set of top k passages during the RAG process. However, when generating the final response, typically only one passage can be included in the prompt. Ensuring the selected passage is accurate is essential for the LLM to answer the query correctly, especially considering the answers to the questions are not covered by publicly available data.

To optimize this process, an intermediate step will be introduced between retrieval and generation. Once the IDs of the top 5 most relevant passages are collected, summaries generated by the models themselves will be retrieved. Alongside the original user query, these pre-generated summaries will be presented to the LLM. Using this brief summary and the context provided by the query, the LLM will then select the most suitable passage from the identified top 5 options. This method aims to aid the language model in accurately selecting the relevant passage for generating the final response. Without the correct context, the LLM cannot accurately answer the query as. With the top 5 passages, there is ample diversity in potential results, yet the number remains conducive for effective decision-making by the LLM.

The following images further illustrate and clarify the novel approach.

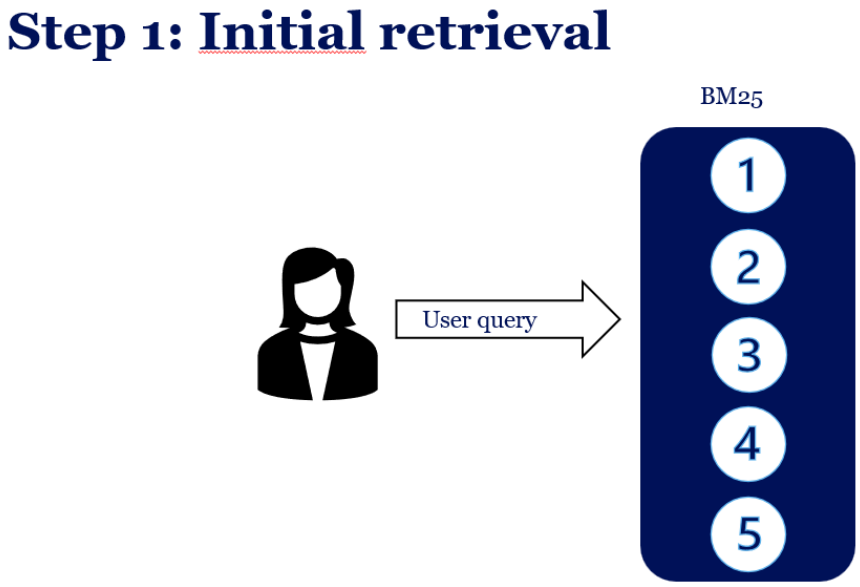

Figure 1: The top 5 most relevant documents to the user query are retrieved using BM25.

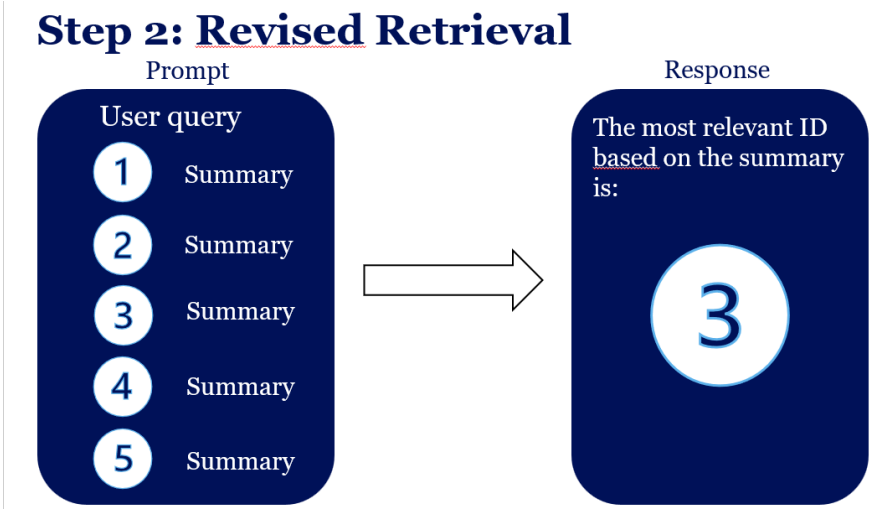

Figure 2: Based on the summaries of the top 5 retrieved documents, the language models pick the most relevant summary to the user query.

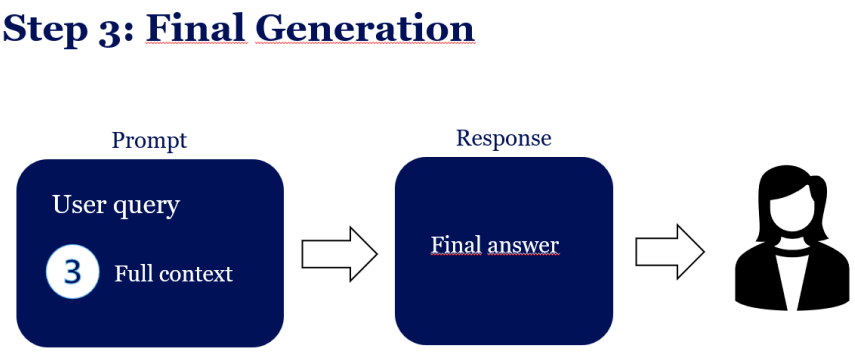

Figure 3: The language models generate a response to the user query with the full context of the chosen document.

This methodological step is designed to refine the retrieval and generation process, leveraging summary information to guide the LLM toward producing more accurate and contextually appropriate responses.
To assess the effectiveness of the RAG strategies, we will examine whether the top 5 retrieved passages include the intended passage. This evaluation involves two main metrics: Accuracy and Mean Reciprocal Rank at 5 (MRR@5). Accuracy is particularly crucial here as it directly measures the accuracy of including the correct passage essential for answering the query.
Following the initial retrieval phase, the analysis will focus on whether the precision of the RAG process improves when the LLM selects the most relevant passage. While MRR@5 provides a broader understanding of how often the intended passage ranks highly, precision focuses specifically on the accuracy of selection, aligning closely with practical applicability in real-world scenarios.
This dual evaluation approach ensures a comprehensive assessment of RAG strategies, emphasizing both the ability to retrieve relevant content and the accuracy in choosing the correct passage for generating responses.

3.2 Generating final response

A critical aspect of this study focuses on the final generation of answers. It aims to determine whether language models can accurately produce correct answers given the appropriate context. To achieve this, all data entries will be tested under the assumption that the retriever has successfully identified the relevant contextual information. This comprehensive analysis will thoroughly assess the capabilities of LLMs in practical applications.
For generating the final answers, two models will be compared. ChatGPT, widely recognized for its text generation capabilities, will serve as the benchmark against which OpenAI's technologies are evaluated.
T5, developed by Google AI, will be used as the base model (65). T5 approaches various NLP tasks as text-to-text problems, enabling its application across a diverse range of tasks including question answering. Trained on extensive datasets, T5 has consistently achieved state-of-the-art performance in multiple benchmarks.
Given the requirement for logical reasoning, the models will be encouraged to adopt a step-by-step reasoning approach aligned with the CoT methodology to enhance their mathematical

proficiency and overall accuracy.
The output of both models will be assessed using various BLUE and ROUGE scores; two widely recognized metrics for evaluating NLP tasks. These metrics gauge the quality of text generated by the models compared to the expected answers.
ROUGE-L measures the longest common subsequence between the generated text and the reference text. It emphasizes the order of words, crucial for preserving the intended meaning of the text.
BLEU-4 is a precision-based metric that evaluates up to 4-grams (sequences of 4 words) in the generated text against the reference text. It calculates how many of these n-grams from the reference appear in the generated text, penalizing shorter outputs to ensure fluency and accuracy.
To ensure the accuracy and reliability of these metrics, evaluations will involve both automated assessments using GPT and a partial manual review to validate the results. This dual approach aims to provide robust validation of the models' performance in generating accurate and contextually appropriate responses.

3.3 Dataset

The proposed methodology will undergo testing using the FinQA dataset (66), renowned for its extensive question-answering tasks designed by financial experts. This dataset is particularly suitable for this study due to its emphasis on multi-step numerical reasoning within the context of financial reports. It comprises 6,251 data entries extracted from 2,700 financial reports.
A data entry consists of the following elements:

pre_text: The texts before the table; #important shows what is in table/acts as header.

post_text: The text after the table; #gives clarification + more info.

table: The table.

id: Unique example ID. Composed by the original report name plus example index for this report. #unique ID I need.

qa:
question: The question.
program: The reasoning program.
gold_inds: The gold supporting facts.
exe_ans: The gold execution result.
program_re: The reasoning program in nested format.

To prepare the data entries from the FinQA dataset (66) for embedding, an element-based chunking approach will be employed. The dataset is already pre-divided into different elements for every data entry. These chunks will then be individually processed for embedding purposes. This approach ensures that all relevant information within each data entry is effectively captured and encoded, ready for further analysis.

4 Experiment

All experiments will be conducted in Python with various libraries to ensure both efficiency and accessibility. Python is chosen due to its widespread use and accessibility, making it an ideal language for conducting experiments in data science and machine learning.

4.1 The Retrieval Process

First, the data must be prepared for analysis. The data entries, already separated into key elements such as pre-text, post-text, and tables, are further divided into chunks of 500 tokens each. These chunks are then processed based on the chosen embedding method.
For OpenAI embeddings, the OpenAIEmbeddings function from the LangChain library is used. LangChain simplifies working with large language models (LLMs) by providing easy access to various LLM functionalities. With a valid API key, this function generates embeddings with 1536 dimensions. Since OpenAI is a closed-source platform, the exact details of how these embeddings are generated are not publicly available. The embeddings are then stored in a Chroma database, which efficiently manages high-dimensional vectors. The same embedding process is applied to user queries, and Chroma's similarity search function is used to retrieve the top 5 most relevant documents.
In this research, the DPR (Dense Passage Retrieval) model from Hugging Face's Transformers library is employed to generate embeddings. The process starts by tokenizing the text into manageable units, which are then processed by the model's encoder to produce 768-dimensional vectors. Questions undergo the same encoding process. These embeddings are indexed using Facebook AI Similarity Search (FAISS), which facilitates quick retrieval of the most relevant passages through a nearest-neighbor search. The top 5 most similar passages to each query are then extracted.
The BM25Retriever from the LangChain library simplifies the implementation of the BM25 retrieval model. Data entries are retrieved based on their segmented elements, and the IDs of the top 5 relevant items are identified.
The top 5 retrieved document IDs are stored in Pandas dataframes, and these results are then compared to the true IDs of the questions. By performing straightforward column matching and comparisons, the accuracy of the results can be evaluated.

4.2 The Revised Method Process

To implement the new revised method, summaries will first be generated for the data entries. The language models will then be provided with the full context and will produce responses based on these summaries. The following is the prompt used for generating the summaries:

```
Summarize the following text in 1-2 sentences,
mentioning the year if included and detailing any information
   ↪ found in tables:
{text chunk}

Summary:
```

The summaries are stored in DataFrames for easy access during the revised prompt phase. Using the top 5 retrieved IDs, the corresponding summaries are fetched. The language models then use these summaries to identify and return the most relevant ID.

```
        Given the question {user query}, and the following
           ↪ retrieved IDs and their summaries:

        1. ID: {1st most relevant ID}, Summary: {summary of 1st
           ↪  most document}
        2. ID: {2nd most relevant ID}, Summary: {summary of 2nd
           ↪  most document}
        3. ID: {3rd most relevant ID}, Summary: {summary of 3rd
           ↪  most document}
        4. ID: {4th most relevant ID}, Summary: {summary of 4th
           ↪  most document}
        5. ID: {5th most relevant ID}, Summary: {summary of 5th
           ↪  most document}

        Pick the most relevant ID strictly based on the
           ↪ information provided above and return it exactly
           ↪ as it is shown.

        Relevant ID:
```

4.3 Final Response Generation Process

The comparison will involve Google AI's T5 language model and OpenAI's GPT-3.5 Turbo. The base T5 model will generate responses with no maximum length constraint. All responses will be recorded in a Pandas DataFrame for further analysis and evaluation. For OpenAI, the GPT-3.5 Turbo version will be utilized. The following prompt will be employed to generate the final responses:

```
        Given the following text:

        {full passage}

        Answer the following question:
```

```
Question: {user query}

Answer:
```

4.4 Evaluation Process

To calculate BLEU and ROUGE scores, two Python libraries can be used. The NLTK library provides the sentence-bleu function to compute BLEU scores from the DataFrames, while the rouge-scorer library facilitates the calculation of various ROUGE scores.
For automatic evaluation, the GPT language model is given the true answers along with those generated by T5 and GPT. It is then tasked with comparing and identifying the correct answers based on the true responses. The following prompt is used during this automatic evaluation:

```
Compare the following answers to determine if they are
   ↪ the same as the model answer.
Answer "1" if they are the same and "0" if they are not
   ↪ .
Return the result as a list in the format: [id,
   ↪ ans1_binary, ans2_binary]

ID: {ID}

Model Answer: "{true answer}"

Generated Answer 1: "{GPT generated answer}"
Is this essentially the same as the model answer? (1
   ↪ for Yes, 0 for No)

Generated Answer 2: "{T5 generated answer}"
Is this essentially the same as the model answer? (1
   ↪ for Yes, 0 for No)
```

The responses are then processed and converted into DataFrames for analysis and to obtain the final results.

5 Experimental results

5.1 Retrieval Results

After preparing the text chunks in their respective formats, the retrievers were assigned the task of retrieving the 5 most relevant documents for each question from the dataset. The retrieved documents were then ranked from most to least relevant. Each question was

associated with a single target document that it should ideally retrieve. By comparing the true ID of the required document with the retrieved IDs, we can calculate metrics such as accuracy and MRR@5 (Mean Reciprocal Rank at 5). The table below illustrates the matches between the retrieved document IDs and the true IDs.

	BM25	DPR	OpenAI Embeddings
1st retrieved document match	564	621	392
2nd retrieved document match	536	576	346
3rd retrieved document match	425	463	288
4th retrieved document match	339	388	271
5th retrieved document match	239	209	186
Total	**2103**	**2257**	**1483**

Table 1: Comparison of retrieval performance across BM25, DPR, and OpenAI embeddings. The table displays the number of correctly retrieved documents for each rank (1st through 5th) and the total number of matches for each method. Traditional methods BM25 and DPR consistently show higher accuracy in retrieving relevant documents than OpenAI embeddings.

	BM25	DPR	OpenAI Embeddings
Accuracy	9,0%	9,9%	6,3%
MRR@5	33,6%	36,1%	23,7%

Table 2: Comparison of retrieval performance metrics between BM25, DPR and OpenAI embeddings. The table presents two key metrics: Accuracy and Mean Reciprocal Rank at 5 (MRR@5). BM25 and DPR perform better with higher values for Accuracy and MRR@5 compared to OpenAI embeddings.

With a dataset comprising 6251 entries, there is still considerable room for improvement in terms of retrieval accuracy and overall performance. The BM25 method demonstrated more effectiveness by correctly retrieving 2103 relevant IDs from the top 5 documents, whereas OpenAI's method retrieved only 1484 relevant elements. DPR was the most effective by retrieving 2257 documents when considering the top 5. Despite OpenAI's advanced dense embeddings, which are designed to capture intricate textual nuances, it has not surpassed the performance of established technologies. In fact, OpenAI's information retrieval method is 30% less accurate than BM25 and 36,4% accurate than DPR. Traditional methods, such as BM25, continue to prove more effective and efficient for information retrieval tasks. This highlights the relevance of classic retrieval techniques, even in the face of OpenAI's newer technologies.

Beyond the final retrieval results, the underlying processes for each method also differ significantly. OpenAI's information retrieval involves substantial computational costs due to its complex embedding procedures. This process entails generating embeddings for each text segment and mapping them into a high-dimensional vector space, which is computationally intensive. These tasks are performed in batches, making it particularly taxing when dealing with large volumes of text. Despite this, once the embeddings are created, they only need

to be computed once for a persistent database, and subsequent updates or additions can be embedded in smaller batches.
DPR is even more computationally demanding than both OpenAI embeddings. It requires substantial processing power and a large amount of RAM for embedding and retrieving vectors. This high demand for computational resources may not be feasible for many organizations, limiting accessibility. Additionally, the time required to retrieve the most similar embeddings is longer, further reducing its practicality in real-world applications.
In contrast, BM25 is much less demanding on computational resources. Its efficiency stems from its reliance on traditional sparse retrieval techniques, which do not require the extensive processing associated with dense embeddings. As a result, BM25 is more streamlined and less resource-intensive, making it well-suited for environments where computational efficiency is crucial. This efficiency in BM25 highlights a practical advantage over more computationally heavy methods like those used by OpenAI, especially when handling large datasets.
The storage requirements for different retrieval methods are also a crucial factor. Dense retrieval techniques necessitate significant storage capacity due to the need to manage large volumes of multi-dimensional vectors. These vectors represent complex embeddings for each text chunk, which can demand substantial data storage. Although efficient indexing can alleviate some of these concerns, the sheer volume of data involved remains a critical consideration, particularly when dealing with extensive datasets.
On the other hand, sparse retrieval methods offer a more storage-efficient alternative. These methods rely on simpler representations, such as term frequency-inverse document frequency (TF-IDF) scores or other sparse vector forms, which require less storage space. Even as the dataset scales up, the storage needs for sparse retrieval methods increase at a slower rate compared to dense methods. This inherent efficiency makes sparse retrieval techniques advantageous in scenarios where storage resources are limited or when managing very large volumes of data.
The retrieval process also varies significantly between methods. Sparse retrieval techniques, such as BM25, are markedly faster than dense retrieval methods. This speed advantage becomes particularly evident when handling large-scale retrieval tasks, such as processing thousands of documents in batches. In real-world scenarios, where individual retrieval requests are typically made one at a time, the difference in speed may seem less critical. However, as the size of the knowledge base grows, the efficiency of the retrieval process can become increasingly important.
Despite the advanced nature of OpenAI's embedding techniques, they do not yet surpass older technologies like BM25 and DPR in terms of retrieval performance. Dense retrieval methods, while sophisticated, incur high computational costs due to the complex embedding and retrieval processes. In contrast, BM25 remains a straightforward and effective approach with significantly lower computational requirements. This efficiency highlights BM25's practical advantages, particularly in scenarios where computational resources are constrained or where rapid, scalable retrieval is essential.

5.2 Revised result

The quality of summaries generated by OpenAI's GPT and Google AI's T5 varied. GPT provided more detailed and comprehensive summaries, adhering closely to the prompt's

instructions and including more nuanced information. In contrast, T5 produced more concise and straightforward summaries, though it often omitted relevant details. As a result, GPT's summaries were more aligned with the prompt's requirements, but some information was still lost in the process.
The following are the summaries generated by GPT and T5 about the same document:

ChatGPT Summary

The table shows annual aircraft fuel consumption and costs for mainline and regional operations in 2018, 2017, and 2016. The company does not currently have any fuel hedging contracts and is susceptible to fluctuations in fuel prices. Various factors such as natural disasters, political disruptions, and changes in fuel-related governmental policy could impact fuel supply and prices in the future, affecting the company's operating results and liquidity.

T5 Summary

A significant portion of our business is dependent on the price and availability of aircraft fuel.

After the initial retrieval of 5 documents, the language models were asked to pick the most relevant ID based on self-generated summaries. For this, the BM25 retrieval method was used as it's the most accurate. Out of 6251 data entries, 2103 true IDs were included in the top 5 most relevant. So if the LLM retrieved the perfect summary every time, the accuracy of the revised retrieval would be 2103.
When presented with the ID and summaries of the 5 retrieved documents, OpenAI's large language model was able to pick the right ID of 569 instances. The accuracy of BM25 without any additional steps is 564. The increase in precision is negligible considering the additional steps required to implement the revision. The T5 language model performed worse and was only able to correctly pick 76 true ids. After reviewing the output generated by T5, the output of the model seemed to not follow the stick instruction given to the output. So while 76 true ids were retrieved clearly, the amount of id's retrieved while correcting for small mistakes from the language model is unclear.

5.3 Answer generation result

For the final answer generation, all 6251 questions, along with their relevant contexts, were provided to the language models.
The quality of responses from the two models varies notably. T5 tends to deliver more straightforward answers with minimal elaboration, while GPT provides more comprehensive responses that often include detailed reasoning. This additional context and explanation can enhance the user's understanding of the answers and reveal potential errors, thereby improving the transparency and reliability of the information.

The answers produced were then assessed using various BLUE and ROUGE scores. The average scores for these metrics are summarized in the table below.

Model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**ROUGE-L**	**ROUGE-1**	**ROUGE-2**
OpenAI	0.0077	0.0029	0.0020	0.0476	0.0482	0.0116
T5	0.0047	0.0017	0.0012	0.0224	0.0226	0.0017
Significant?	Yes	Yes	Yes	Yes	Yes	Yes

Table 3: This table presents the average performance scores of two models, OpenAI and T5, evaluated using BLEU-1, BLEU-2, BLEU-3, ROUGE-L, ROUGE-1, and ROUGE-2 metrics. The scores indicate how well each model performs in generating text, with higher values representing better performance. OpenAI outperforms T5 across all metrics, with notably higher BLEU and ROUGE scores. The "Significant?" row confirms that these differences are statistically significant across all evaluated metrics, highlighting a meaningful performance gap between the two models.

The scores of the standard metrics are low, with many values being zero for both models. The table below shows the counts of zero values. Despite the low metric scores, OpenAI's model appears to outperform the T5 model in generating answers with 4617%.

	BLEU-1	**BLEU-2**	**BLEU-3**	**ROUGE-L**	**ROUGE-1**	**ROUGE-2**
OpenAI	5615	5615	5615	2775	2775	5384
T5	6183	6183	6183	5926	5926	6234

Table 4: This table displays the count of zero scores for BLEU-1, BLEU-2, BLEU-3, ROUGE-L, ROUGE-1, and ROUGE-2 metrics across two models, OpenAI and T5.

However, assessing numerical results posed significant challenges for automatic evaluations. GPT's language model tends to present numerical answers in a more natural language format. For instance, while a specific table might represent $3.8 million as 3800, GPT often responds with "3.8 million" rather than adhering strictly to the table's representation. This discrepancy complicates the process of automatically and uniformly integrating and correcting numerical values across different tables.
OpenAI's language model GPT3.5 Turbo was presented with the true answer, GPT3.5 Turbo's answer, and T5's answer. The LLM was asked whether or not the answers generated by the language models corresponded with the true answer. Out of the 6251 responses, 5089 successfully adhered to the task. The results of the automatic evaluation can be found in the following table:

Model	Identified as Correct	Percentage
GPT-3.5 Turbo	1438	28.3%
T5 Model	30	0.6%

Table 5: This table compares the performance of two models, GPT-3.5 Turbo and T5, in terms of the number and percentage of responses identified as correct by ChatGPT. GPT-3.5 Turbo outperforms the T5 model, with 1,438 correct identifications, representing 28.3% of the total, while the T5 model has only 30 correct identifications, accounting for just 0.6%. This contrast suggests the superior accuracy and reliability of GPT-3.5 Turbo compared to the T5 model in generating correct responses.

A manual review was conducted on 100 random responses of those flagged by GPT3.5 Turbo as incorrect for both models. The review revealed that 54 of these GPT responses were correct, despite being marked as incorrect during the automated evaluation. Conversely, 4 responses from the T5 model were identified as false negatives. This highlights the need for clear and consistent evaluation guidelines. The discrepancies often arose from inconsistencies in numeral formatting or the number of decimal places used.
Regarding computational resource requirements, there are notable differences between the two models. T5, being a locally run model, can place a substantial strain on computational resources, especially with large datasets. In contrast, OpenAI's model operates via server calls, thereby minimizing the demand on local resources. When running on a server, this distinction becomes less significant, as both models exhibit negligible differences in response times.

6 Discussion

The retrieval results from the experiment demonstrate that BM25 surpasses OpenAI Embeddings in both accuracy and computational efficiency for retrieving the top 5 relevant documents from a dataset of 6,251 entries. Although dense embeddings are theoretically beneficial for capturing semantic similarities, the current implementation of OpenAI's method does not outperform the established BM25 technology.
To enhance document selection, summaries were generated using OpenAI's GPT and Google AI's T5 models. These summaries were then used to determine the most relevant document among five options. The improvement in accuracy was minimal, indicating that while language models excel at understanding and generating text, they may still struggle with comprehending nuanced relevance without the support of dense vectors. This suggests that their ability to accurately identify relevant information remains limited when not guided by numerical data representations.
For the final answer generation, despite both models scoring relatively low on standard metrics, OpenAI's language model outperforms the T5 model. OpenAI's model not only demonstrates greater accuracy but also delivers more comprehensive and interpretable responses. Its ability to provide detailed explanations and clearer answers makes it a valuable tool for users, particularly in breaking down complex tasks and aiding in the democratization of data analytics. This enhanced capability positions OpenAI's model as a superior choice for tackling intricate queries and improving overall understanding.

While research shows remarkable results with OpenAI's GPT4 LLM when it comes to financial statement analysis (67), this study has shown the opposite. This could be due to the nature of the task being different in both studies. The tasks in the mentioned study were mainly focused on analyzing textual data. The tasks in this study mainly focused on successfully interpreting tables and numbers to answer questions. This could explain the difference in the results.
To address the first research question:

How can large language models (LLMs) be used to simplify the statistical analysis needed for data-driven business decisions?

LLMs can streamline the process of analyzing textual data by presenting it in a more accessible format. They provide enhanced interaction with the data, which can be particularly useful for users who may not fully understand complex concepts. LLMs can facilitate statistical analysis and potentially save time during in-depth evaluations.
OpenAI's language model outperformed the T5 model by an impressive 4616%, making it a powerful tool for simplifying data-driven business decisions. This drastic increase in performance allows for more accurate and efficient decision-making compared to older technologies. However, this research also highlights that while LLMs like GPT can assist in contextualizing information, they are not yet fully reliable for determining the most relevant context on their own. This limitation underscores the need for integrating LLMs with other analytical methods to ensure comprehensive and accurate data analysis.
To address the second research question:

To what extent can large language models (LLMs) make statistical analysis accessible in business for those with limited knowledge of statistics?

LLMs have the potential to simplify text analysis and make interactions with complex data more user-friendly. They can facilitate understanding by breaking down information and providing explanations that can aid users with limited statistical knowledge. While LLMs can enhance accessibility and speed up the analytical process, they cannot be relied upon entirely; users must still have some foundational knowledge to verify and interpret results accurately. GPT's accuracy of 28.3% indicates that there is substantial room for improvement before it can be considered a reliable tool for data analysis.
Despite their advancements, OpenAI's embedding models, in particular, seem to not currently match the performance of traditional information retrieval methods. They perform roughly 30% worse than traditional methods like BM25 and DPR.Their primary strength remains in text generation rather than in retrieval efficiency.
To address the third research question:

How can the performance of a large language model (LLM) be assessed in a Question Answering (QA) use case?

Evaluating the performance of language models in QA scenarios has been proven to be challenging, particularly due to discrepancies between numerals in tables, model answers, and generated responses. These inconsistencies complicate the assessment of model accuracy

and relevance. To improve evaluation, it may be beneficial to preprocess and standardize the dataset, tables, and answers to ensure alignment. This preprocessing would help in making the performance assessment clearer and more straightforward. Additionally, leveraging GPT to review and validate answers, provided numerical representations are straightened out, could further enhance the accuracy of performance evaluations.

Although OpenAI's embedding methods have not shown superiority over existing techniques in isolation, they could potentially offer advantages when combined with sparser retrieval approaches. Integrating OpenAI's embeddings with a sparse retrieval method might enhance their effectiveness compared to traditional methods. This combined approach could be particularly advantageous given that embeddings are less computationally intensive than Dense Passage Retrieval (DPR) methods. By focusing on a reduced set of passages to embed and compare, OpenAI's embeddings may yield better performance and efficiency, especially in scenarios where computational resources are limited. Further investigation into this combined approach could reveal its full potential and practical benefits.

Since the revised information retrieval relied on summaries, the quality and content of these summaries could have impacted the results. The effectiveness of self-generated summaries and their ability to highlight key information likely influenced the accuracy of the revised document retrieval. Additionally, presenting the language models with five summaries might have been too much, potentially affecting their ability to discern relevance accurately. Testing with fewer summaries could help the models better differentiate between options and improve accuracy. Further research is needed to explore this approach and to determine whether language models can effectively identify the most relevant passages without relying on numerical vectors.

One limitation of this study is the dataset used. Although the method was tailored specifically for the FinQA dataset, which minimized the need for extensive preprocessing, the dataset's chunked structure—where elements are already separated—might not represent real-world scenarios accurately. In practical applications, more nuanced chunking approaches would likely be necessary, potentially affecting both the information retrieval process and the accuracy of the final results. Additionally, while the data entries were sized appropriately to fit within the prompt constraints, it may not always be the case with broader datasets which could introduce new complexities to the implementation of the approach.

Another limitation of the dataset pertains to the nature of the questions. Many questions are broad in scope, and the dataset includes over 2,700 reports, making it challenging to pinpoint the specific report relevant to each question. The accuracy of information retrieval often relies on specific details, such as specific years or companies, which are crucial for identifying the correct report. To address this, requiring users to provide additional specifications could help narrow the search space and improve retrieval accuracy. Further research should focus on developing strategies to refine the search space for financial reports by leveraging key distinguishing features to enhance precision in information retrieval.

Implementing and utilizing OpenAI's language models in contexts where they interact with internal documents requires trust. The integration of these models involves granting them access to potentially sensitive and proprietary information. This access raises significant concerns, as the models process large volumes of data without a clear understanding of how this data is managed, stored, or protected. Consequently, organizations may hesitate to adopt and deploy large language models due to the potential risks associated with data security and privacy breaches.

7 Conclusions and Further Research

The volume of data generated by companies has increased significantly. Yet, top management and decision-makers often struggle to make sense of this vast amount of information. Large language models (LLMs) present a promising solution for enhancing data accessibility and comprehension. This research aimed to evaluate whether OpenAI's technologies could serve as a comprehensive solution for data-driven business decisions, effectively bridging the gap between complex data and actionable insights.

This study compared traditional methods to OpenAI's technologies to research if LLMs have additional value during data-driven decision-making. OpenAI's technology was applied in two distinct processes: information retrieval and answer generation. The dense vector embeddings generated by OpenAI have not proven to surpass traditional retrieval methods in effectiveness. Even when given the task of selecting the correct ID from the top 5 most relevant documents, OpenAI's system struggles to consistently identify the correct one.

Final answer generation is an area where OpenAI's LLMs excels. It provides interpretable and comprehensive answers that can significantly aid decision-makers by supporting their data analysis efforts. However, while GPT-3.5 Turbo offers valuable insights, it is not yet reliable enough to serve as a sole decision-making tool. It requires careful verification and should be used with caution to ensure accuracy.

Further research should focus on integrating OpenAI's embedding models with other methods to fully explore their potential. By combining these embeddings with complementary retrieval techniques, it may be possible to enhance their effectiveness and determine whether they can outperform traditional methods. Additionally, given their potential for lower computational costs compared to conventional dense retrieval approaches, such research could uncover ways to leverage these models more efficiently and economically.

The proposed method of using language models to select summaries warrants additional research. Future research could explore whether this approach holds promise for simplifying information retrieval. If language models can effectively identify and prioritize key passages from summaries, it could significantly streamline and enhance the retrieval process. Additionally, strategically narrowing the search space using distinctive textual elements, rather than relying solely on numerical embeddings, could further improve the efficiency and accuracy of information retrieval.

References

[1] E. Curry, "The big data value chain: definitions, concepts, and theoretical approaches," *New horizons for a data-driven economy: A roadmap for usage and exploitation of big data in Europe*, pp. 29–37, 2016.

[2] I. A. Ajah and H. F. Nweke, "Big data and business analytics: Trends, platforms, success factors and applications," *Big Data And Cognitive Computing*, vol. 3, no. 2, p. 32, 2019.

[3] J. S. Guerrero-Prado, W. Alfonso-Morales, E. F. Caicedo-Bravo, B. Zayas-Pérez, and A. Espinosa-Reza, "The power of big data and data analytics for ami data: A case study," *Sensors*, vol. 20, no. 11, p. 3289, 2020.

[4] A. Merendino, S. Dibb, M. Meadows, L. Quinn, D. C. Wilson, L. Simkin, and A. I. Canhoto, "Big data, big decisions: The impact of big data on board level decision-making," *Journal of Business Research*, vol. 93, pp. 67–78, 2018.

[5] M. Armanr and U. R. Lamiyar, "Applied quantitative analysis (aqa)," *International Journal For Global Academic Scientific Research*, vol. 3, pp. 46–67, 2023.

[6] C. C. Da Silveira, C. B. Marcolin, M. Da Silva, and J. C. Domingos, "What is a data scientist? analysis of core soft and technical competencies in job postings," *Revista Inovação, Projetos e Tecnologias*, vol. 8, no. 1, pp. 25–39, 2020.

[7] M. Du, F. He, N. Zou, D. Tao, and X. Hu, "Shortcut learning of large language models in natural language understanding: A survey," 08 2022.

[8] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," 2016.

[9] S. C. Fanni, M. Febi, G. Aghakhanyan, and E. Neri, *Natural Language Processing*, pp. 87–99. Cham: Springer International Publishing, 2023.

[10] P. Johri, S. K. Khatri, A. T. Al-Taani, M. Sabharwal, S. Suvanov, and A. Kumar, "Natural language processing: History, evolution, application, and future work," in *Proceedings of 3rd International Conference on Computing Informatics and Networks* (A. Abraham, O. Castillo, and D. Virmani, eds.), (Singapore), pp. 365–375, Springer Singapore, 2021.

[11] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[12] M. U. Hadi, Q. A. Tashi, R. Qureshi, *et al.*, "A survey on large language models: Applications, challenges, limitations, and practical usage," *TechRxiv*, 2023.

[13] K. Jing and J. Xu, "A survey on neural network language models," 2019.

[14] Y. Bengio, "Neural net language models," *Scholarpedia*, vol. 3, no. 1, p. 3881, 2008. revision #140963.

[15] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[16] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, vol. 25, pp. 51–65, 2023.

[17] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 4, pp. 1413–1430, 2024.

[18] C. McIntosh, *Cambridge Advanced Learner's Dictionary*. Cambridge University Press, fourth ed., 2013.

[19] B. Wang, X. Yue, and H. Sun, "Can chatgpt defend its belief in truth? evaluating llm reasoning via debate," 2023.

[20] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu, "Chateval: Towards better llm-based evaluators through multi-agent debate," 2023.

[21] Y. Zhang, S. Mao, T. Ge, X. Wang, A. de Wynter, Y. Xia, W. Wu, T. Song, M. Lan, and F. Wei, "Llm as a mastermind: A survey of strategic reasoning with large language models," 2024.

[22] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[23] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," March 2022.

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020.

[25] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–40, 2023.

[26] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1–8, June 2015.

[27] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning," *ACM Transactions On Intelligent Systems And Technology*, vol. 10, no. 2, pp. 1–37, 2019.

[28] G. Bhatt, S. Chandhok, and V. N. Balasubramanian, "Learn from anywhere: Rethinking generalized zero-shot learning with limited supervision," July 2021.

[29] Z. Jiang, J. Araki, H. Ding, and G. Neubig, "Understanding and improving zero-shot multi-hop reasoning in generative question answering," 2022.

[30] V. Mavi, A. Jangra, and A. Jatowt, "A survey on multi-hop question answering and generation," *arXiv*, 2022. Cornell University.

[31] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and Google Research, Brain Team, "Chain-of-thought prompting elicits reasoning in large language models," in *36th Conference On Neural Information Processing Systems (NeurIPS 2022)*, 2022.

[32] F. Liu, Y. Liu, S. Lin, H. Huang, R. Wang, Z. Yang, and L. Zhang, "Exploring and evaluating hallucinations in llm-powered code generation," *arXiv*, 2024. Cornell University.

[33] Q. Huang, F. Huang, D. Tao, Y. Zhao, B. Wang, and Y. Huang, "Coq:an empirical framework for multi-hop question answering empowered by large language models," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11566–11570, 2024.

[34] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-verification reduces hallucination in large language models," 2023.

[35] J. Long, "Large language model guided tree-of-thought," 2023.

[36] B. Goertzel, "Artificial general intelligence: concept, state of the art, and future prospects," *Journal of Artificial General Intelligence*, vol. 5, no. 1, p. 1, 2014.

[37] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang, "Sparks of artificial general intelligence: Early experiments with gpt-4," 2023.

[38] L. Zhang, K. Jijo, S. Setty, E. Chung, F. Javid, N. Vidra, and T. Clifford, "Enhancing large language model performance to answer questions and extract information more accurately.," January 2024.

[39] A. Balaguer, V. Benara, R. L. De Freitas Cunha, R. De M Estevão Filho, T. Hendry, D. Holstein, J. Marsman, N. Mecklenburg, S. Malvar, L. O. Nunes, R. Padilha, M. Sharp, B. Silva, S. Sharma, V. Aski, and R. Chandra, "Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture," January 16 2024.

[40] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-tuning or retrieval? comparing knowledge injection in llms.," December 2023.

[41] S. Setty, K. Jijo, E. Chung, and N. Vidra, "Improving retrieval for rag based question answering models on financial documents," 2024.

[42] A. J. Yepes, Y. You, J. Milczek, S. Laverde, and R. Li, "Financial report chunking for effective retrieval augmented generation," 2024.

[43] Y. Tang and Y. Yang, "Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries," January 2024.

[44] W. Xiong, X. L. Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, Y. Mehdad, W. tau Yih, S. Riedel, D. Kiela, and B. Oğuz, "Answering complex open-domain questions with multi-hop dense retrieval," 2021.

[45] A. Sethy and B. Ramabhadran, "Bag-of-word normalized n-gram models.," in *INTERSPEECH*, pp. 1594–1597, 2008.

[46] A. Lommatzsch and J. Katins, "An information retrieval-based approach for building intuitive chatbots for large knowledge bases.," in *LWDA*, pp. 343–352, 2019.

[47] B. Reichman and L. Heck, "Retrieval-augmented generation: Is dense passage retrieval retrieving?," 2024.

[48] R. Ren, Y. Qu, J. Liu, W. X. Zhao, Q. She, H. Wu, H. Wang, and J.-R. Wen, "Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking," 2023.

[49] B. Mitra, N. Craswell, *et al.*, "An introduction to neural information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 13, no. 1, pp. 1–126, 2018.

[50] F. Xing, "Designing heterogeneous llm agents for financial sentiment analysis," 2024.

[51] U. Gupta, "Gpt-investar: Enhancing stock investment strategies through annual report analysis with large language models," 2023.

[52] C. Jeong, "Domain-specialized llm: Financial fine-tuning and utilization method using mistral 7b," *Journal of Intelligence and Information Systems*, vol. 30, p. 93–120, Mar. 2024.

[53] A. Lopez-Lira and Y. Tang, "Can chatgpt forecast stock price movements? return predictability and large language models," 2023.

[54] T. Baker and B. G. C. Dellaert, "Regulating robo advice across the financial services industry," *Social Science Research Network*, 2017.

[55] H. Zhao, Z. Liu, Z. Wu, Y. Li, T. Yang, P. Shu, S. Xu, H. Dai, L. Zhao, G. Mai, N. Liu, and T. Liu, "Revolutionizing finance with llms: An overview of applications and insights," 2024.

[56] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," 2023.

[57] Z. Yuan, H. Yuan, C. Tan, W. Wang, and S. Huang, "How well do large language models perform in arithmetic tasks?," 2023.

[58] J. An, J. Lee, and G. Gweon, "Does chatgpt comprehend the place value in numbers when solving math word problems?," in *Proceedings of the Workshop "Towards the Future of AI-augmented Human Tutoring in Math Learning" co-located with The 24th International Conference on Artificial Intelligence in Education (AIED 2023), Tokyo, Japan, July 3, 2023* (D. R. Thomas, J. Lin, and K. R. Koedinger, eds.), vol. 3491 of *CEUR Workshop Proceedings*, pp. 49–58, CEUR-WS.org, 2023.

[59] S. Imani, L. Du, and H. Shrivastava, "Mathprompter: Mathematical reasoning using large language models," 2023.

[60] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," 2023.

[61] K. S. Phogat, C. Harsha, S. Dasaratha, S. Ramakrishna, and S. A. Puranam, "Zero-shot question answering over financial documents using large language models," 2023.

[62] A. Zhou, K. Wang, Z. Lu, W. Shi, S. Luo, Z. Qin, S. Lu, A. Jia, L. Song, M. Zhan, and H. Li, "Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification," 2023.

[63] J. Ahn, R. Verma, R. Lou, D. Liu, R. Zhang, and W. Yin, "Large language models for mathematical reasoning: Progresses and challenges," 2024.

[64] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. tau Yih, "Dense passage retrieval for open-domain question answering," 2020.

[65] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023.

[66] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, and W. Wang, "Finqa: A dataset of numerical reasoning over financial data," pp. 3697–3711, 01 2021.

[67] A. G. Kim, M. Muhn, and V. V. Nikolaev, "Financial statement analysis with large language models," tech. rep., Chicago Booth Research Paper, 2024.