University College London

MSc Business Analytics Dissertation 2023-24

Automating Marketing Insights: Evaluating Prompt Engineering Techniques and Large Language Model Performance

Word Count: 11,827

Date: 02/08/2024

Candidate Number: FYWW6

Disclaimer:

I hereby declare that this dissertation is my individual work and to the best of my knowledge and confidence, it has not already been accepted in substance for the award of any other degree and is not concurrently submitted in candidature for any degree. It is the end product of my own independent study except where other acknowledgement has been stated in the text.

Marking Sheet – MSc Business Analytics Consultancy Project/Dissertation 2023-24

Criteria/Weight	Supervisor's comments
Topic, theoretical framework, literature, and methodology (35%): Topic is clearly identified and boundaries are asserted. Knowledge of relevant theories and their limitations. Current and relevant literature coming from reliable sources. Appropriate and adequate methodology for topics. Detailed methodology facilitating replication of project and reproducibility of results.	
Analysis and conclusions /recommendations (35%): Use of primary and/or secondary data. Rigorous analysis and interpretations. Alternative interpretations/arguments are considered. Limitations are identified and justified by reasonable arguments. Conclusions/recommendations are fully consistent with evidence presented.	
Structure, originality and presentation (10%): Provides a concise summary. Demonstrates an understanding of business context. Coherent and appropriate structure. Adequate presentation, language, style, graphs, tables, and referencing. Appropriate use of visualisation. Presents business recommendations.	
Complexity of project scope and progress made towards business goals (10%): Progress made towards overcoming technical and operational challenges encountered during the project. Progress made in overcoming problem framing and theoretical and data related problems encountered during the project.	
Project Management (10%): Good use of project management and communication tools. Use of Kanban board for structuring project work. Evidence of objectives being broken down in appropriate tasks and timely engagement with primary supervisor.	

General marking guidelines

85+ Outstanding work of publishable standard.
70-84 Excellent work showing mastery of the subject matter and excellent analytical skills.
60-69 Very good work. Interesting analysis with original insights. Some minor errors.
50-59 Good work which only covers a basic analysis. Some problems but no major omissions.
40-49 Inadequate work. Not sufficiently analytical. Some major omissions.
39- Work seriously flawed. Lack of clarity and argumentation. Too descriptive.

Mark: _____

Abstract

In this work, we develop a data-to-text pipeline that transforms a client's marketing data into actionable insights on two different timescales. First, a simple weekly snapshot provides the client with an understanding of their marketing metrics for that week. Second, a monthly review not only offers an understanding of their metrics but also includes a comparison to their competitors. Through a systematic review of four different prompting techniques, we identify an optimal prompting framework for generating both types of insights. Additionally, we create an evaluation framework to monitor the quality of the outputs, reducing the need for manual intervention. This evaluation framework is validated against human judgments to assess its effectiveness. Finally, we conduct a cost-benefit analysis comparing state-of-the-art LLMs (Advanced models) to their standard counterparts (Base models) and provide recommendations for future use.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Business Problem

The vast availability of data over the past decade has transformed the business landscape. Companies across various industries are increasingly leveraging Big Data Analytics (BDA) to gain insights, optimise operations and create competitive advantages. The shift towards making data driven decisions represents a significant change from the traditional methods that relied heavily on intuition and experience. BDA provides organisations with the ability to derive actionable insights from extensive datasets which can lead to improved performance and innovation (Grover et al., 2018). By integrating BDA into business processes, firms can identify previously undetectable patterns and trends, enabling more informed and timely decision-making. This capability is becoming a crucial factor in driving innovation and performance, giving companies a deeper understanding of their operations (Arias-Pérez et al., 2022; Chen & Liang, 2023). Consequently, adopting BDA is essential for firms seeking to enhance both internal and external business efficiencies through actionable knowledge (Jiwat & Zhang, 2022).

While the advantages of BDA for businesses are widely acknowledged, resistance to change remains a common challenge, especially when it involves adopting new methods for long-established processes (Scholkmann, 2021). Research indicates that BDA can have a disruptive effect on board-level decision-making (Merendino et al., 2018). This challenge often arises from the need for individuals lacking the skills to effectively interpret and utilise big data. Additionally, cognitive biases can lead individuals to rely heavily on past

experiences, which may skew their interpretation of new data, causing them to overlook insights that contradict their preconceived notions. Being presented with vast amounts of data may cause cognitive overload, making it difficult to discern what is important. These factors collectively impede the full realisation of big data's benefits.

The challenges identified in the Merendino et al. (2018) highlight the need for professionals who understand data to bridge the gap between complex data insights and decision-makers who may lack technical training. Data experts can play a pivotal role in making data more accessible and interpretable, ensuring that employees across the business can leverage big data effectively. By translating intricate data into clear, actionable insights, these professionals can facilitate more informed and strategic decision-making processes at the highest levels of an organisation.

1.2 Collaborating Company

This project was completed in collaboration with a start-up brand value agency that provides marketing and PR consulting to a wide range of brands. Alongside the consultancy services, the company operates a platform that allows clients to track all their brand's communications across paid, owned, earned, and social media, relative to their competitors. The platform is organised into four dashboards, each displaying various metrics. The four components of this platform include:

Table 1. Description of the four key dashboards on the marketing analytics platform

Dashboard	*Description*
Market Intelligence	Offers tailored competitor intelligence, tracking key metrics like share of voice, traffic, search, and excess share of search. It provides insights into the company's market position through search ranking analysis and keyword opportunities, revealing competitors' budget allocations for a comprehensive competitive landscape view.
Brand Exposure	Tracks global media monitoring, social listening, and paid campaign performance, integrating bespoke commercial metrics and ROI. It simplifies complex data, helping companies identify effective and ineffective strategies.
Audience & Traffic	Provides a comprehensive understanding of audience engagement across devices, locations, media, and creative content, linking behaviour to commercial impact. It monitors inbound traffic segmented by paid, owned, earned, and social media, and offers insights into the value of each link by tracking the consumer journey through to revenue generation.
Sales & Product	Tracks sales performance at both macro and micro levels, including orders, revenue, average order value, and product breakdowns. It offers a real-time overview of inventory, SKUs, and product categories, detailed by region and sub-brand, providing an up-to-date picture of the company's product landscape.

On the platform, users can select the date range for the data they want to view, and the relevant interactive graphs will be generated. Users can hover over specific points in time to view detailed data. While these features are useful for those familiar with the metrics, the platform offers over 30 different metrics, making it time-consuming to track them all. Marketing and communications professionals spend a significant amount of time reporting on performance to justify ROI, often doing this with written insights. Writing these insights is often a repetitive and arduous task, focusing on summarising historical data to describe past performance. This process can be optimised through a data-to-text pipeline that automates the generation of insights, significantly reducing the time and effort required.

The main goals of this project are to: i) Establish an effective framework through prompt engineering to create a data-to-text pipeline for automated insight generation, ii) Develop an evaluation framework for a large language model to ensure the quality of output, and iii) Conduct a cost-benefit analysis to determine the most suitable LLM to use.

2 Literature Review

2.1 What is Natural Language Generation?

Traditional Natural Language Generation (NLG) involved creating computer systems that generate text from non-linguistic representations of information (Reiter, 1996). The generated text can vary from single phrases to full-page explanations (Dong et al., 2022). Early NLG systems were used for writing business letters (Springer et al., 1991), automating weather reports (Goldberg & Driedger, 1994), generating football match reports (Theune et al., 2001). As well as a tool that could break down technical information to non-technical users (Rambow & Korelsky, 1992).

The early NLG systems were developed before the advent of large language models, limiting the quality and scope of the generated text compared to modern capabilities. Nonetheless, these pioneering efforts laid the groundwork for future advancements in the field. For instance, they enabled the transformation of neonatal care data into text tailored for different audiences whether for doctors, nurses, or parents (Mahamood & Reiter, 2011) and generated sports reports from different team perspectives (Van Der Lee et al., 2017). To appreciate the progress and current state of the field, we will briefly examine the construction of these initial systems.

2.1.1 Overview of core NLG Tasks

Early NLG systems were rule-based, using handcrafted rules and modular designs. These systems decomposed the generation process into specific tasks, each handled by a module. Reiter & Dale (2000) provided a six-step framework for these tasks:

Table 2. Six-step framework for NLG Reiter and Dale (2000)

Task	***Description***
Content selection	Determining which data to include or exclude in the generated text based on the target audience.
Content ordering	Arranging selected information in a logical sequence.
Aggregation	Combining multiple messages into single sentences to enhance readability.
Lexicalisation	Selecting specific words or phrases to express the message.
Referring expression generation	Characterised by Reiter & Dale (2000) as 'the task of selecting words or phrases to identify domain entities. Ensuring that the generated text provides enough information to distinguish one entity from others, deciding whether to use pronouns, proper names, or descriptive phrases.
Linguistic realisation	Forming well-structured sentences by combining outputs of previous steps.

2.1.2 Traditional Approaches to NLG

Traditional NLG systems were based on these six steps and employed the three main approaches to perform the core tasks of text generation:

i) Modular architectures

The original proposal for NLG architectures by Reiter (1994), suggested a clear division of sub-tasks: text-planning (deciding what to say), sentence-planning (deciding how to say it), and linguistic realisation (ensuring grammatical and logical coherence). This architecture creates distinct stages, with each task being fully completed before moving on to the next see Figure. 1. It has been used in rule-based systems, providing creators with high control over the output, but requiring significant manual effort to maintain. For example, this architecture was used by Goldberg & Driedger (1994) to generate weather reports, where the constraints of the task are clear.

Figure 1. Classical three-stage NLG architecture Reiter and Dale (2000). Darker segments illustrate the three main modules, lighter segments show the outputs.

This approach received some push back, particularly the "generation gap" (Meteer, 1995), which occurs when early decisions in the process have unintended consequences later. This can be problematic when generating larger sequences of text, leading to the exploration of more flexible architectures where the boundaries between various stages are blurred.

ii) Template-Based Approaches

Templates were a straightforward method used in early NLG systems, especially when the domain was narrow and the required variability in output was low. Templates contained placeholders for variables, which were filled with specific data points to generate text. While templates ensured grammatical correctness and simplicity, they lacked flexibility and scalability for more complex applications.

Example:

```
plaintextCopy The temperature in $location reached $temperature degrees at
$time.
```

This template could generate sentences like:

```
plaintextCopy The temperature in codeLondon reached code25 degrees at
code3pm.
```

While templates ensured simplicity and correctness, they could not adapt easily to new domains or generate more varied and nuanced text. This rigidity led to the exploration of more sophisticated methods, including statistical and machine learning approaches, which offered greater flexibility and scalability by learning from large datasets and incorporating context-aware decision-making processes.

iii) Planning based approaches

Unlike modular systems, which separate tasks into distinct stages, planning-based methods integrate strategic and tactical elements, viewing NLG as a cohesive process. Each action in this approach changes the context, including the discourse history, physical setting, and the user's beliefs and actions. An example of planning-based NLG is the KAMP system, which generates referring expressions by considering what the participants know and believe about the situation (Appelt, 1985). This system uses actions with preconditions and effects, ensuring that each generated phrase accurately updates the listener's understanding. For instance, to refer to an entity like "the tall oak tree," KAMP would first ensure that the listener can distinguish this tree from others by incrementally adding distinguishing properties (Gatt & Krahmer, 2018).

Planning-based approaches are particularly effective in dynamic environments, such as for dialogue systems. For example, in generating restaurant recommendations, a system might use planning-based NLG to adapt its suggestions based on user preferences and previous interactions, optimizing for clarity and relevance through reinforcement learning. This method allows for more adaptive and contextually appropriate text generation, managing complex communicative tasks more fluidly than traditional modular systems.

2.1.3 NLG in 2010-20

NLG gained mainstream media attention after an article was published in *Wired* magazine on 24 April 2012 with the title 'Can an Algorithm Write a Better News Story Than a Human Reporter?'. Around this time companies like Automated Insights and Narrative Science emerged, developing bespoke applications or frameworks for customers. These tools

integrated with business intelligence platforms like Tableau or Power BI, facilitating narrative generation alongside visual data. Although companies were able to provide NLG solutions, it involved highly skilled technical work to develop them (Dale, 2020).

Traditional data-to-text generation systems relied on rules or human-crafted templates, filling placeholders with dialogue inputs during execution (Reiter, 1996). In this process, natural language expressions were transformed into templates by replacing data-specific words with placeholders, following rules based on text and data patterns. During the data-to-template generation phase, these templates were used to create complete sentences, resulting in coherent text outputs (Osuji et al., 2024). Despite these advancements, challenges such as scalability and maintaining quality and consistency across different domains persisted. Integrating sophisticated linguistic knowledge into NLG systems to enhance performance in complex scenarios became a significant focus of ongoing research (Dale, 2020; Gatt & Krahmer, 2018) Future developments would go on to focus on advanced neural network architectures led to the support of multimodal content generation, combining text with visual and auditory data to create richer and more contextually relevant outputs. This focus resulted in the development of large language models, which transformed the landscape entirely, addressing these challenges with their unprecedented capabilities.

2.2 Large Language Models (LLMs)

Large Language Models (LLMs) represented a significant advancement in the field of Natural Language Processing (NLP), revolutionising the way machines understand and generate human-like text. These models, built on the foundation of deep learning, specifically neural network architectures, and have demonstrated high level capabilities in language comprehension, generation, and even reasoning tasks.

The emergence of LLMs began with the introduction of the Transformer model by Vaswani et al. (2017), which proposed a novel network architecture based on attention mechanisms. The Transformer model's architecture is based on self-attention mechanism, which allowed it to capture long-range dependencies in text and parallel processing. This made it possible to train models on large text corpora. The ability to maintain contextual coherence over extended periods which has led to significant gains in accuracy in downstream tasks such as language understanding (Liu et al., 2019), translation (Lample & Conneau, 2019), summarisation (Lewis et al., 2019) and others. The Transformer's architecture typically employs an encoder-decoder (Brown et al., 2020) structure. The encoder processes the input text to produce a set of hidden states, which are subsequently used by the decoder to generate the output text. This architecture can be extended and scaled using approaches such as the Mixture of Experts model (Fedus et al., 2022), which significantly enhances performance by increasing the number of parameters or experts involved.

2.2.1 Model training Methods

Training LLMs involves a two-stage process: pre-training and fine-tuning. During the pre-training phase, the model is exposed to extensive textual data from a variety of sources, learning to understand and generate human language by predicting the next word in a sentence. This stage involves the adjustment of billions of parameters to optimize the model's performance (Brown et al., 2020).

Following pre-training, fine-tuning can occur, where a pre-trained model undergoes additional training on a more specific dataset related to a specific target task. This enhances the model's performance in specific applications such as question answering, text

classification, or fraud detection. In the fine-tuning phase, instruction-tuning can be employed, where the model receives explicit instructions and examples to better align its responses with user-specific requirements. This approach, combined with techniques like reinforcement learning from human feedback (RLHF), refines the model's outputs based on human evaluations, ensuring more accurate and contextually appropriate responses (Gunel et al., 2020). Research has shown that with the correct training, models with fewer parameters have the capability to outperform larger models (Schick & Schütze, 2020)

2.2.2 Emerging Capabilities Through Scalability

Despite the ability to fine-tune large language models, research has shown that by just increasing the model size, it can also lead to significant improvements in performance. Early work on LLMs such as the GPT-2 model demonstrated that language models trained on large-scale datasets could perform various NLP tasks with minimal to no task-specific training data, displaying the model's few-shot learning capabilities (Radford et al., 2018). GPT-3 further expanded this potential by scaling up the model parameters, leading to even more remarkable performance in generating coherent, contextually appropriate, and human-like text (Brown et al., 2020).

Research by Kaplan et al. (2020) showed the potential of scaling LLMs, their research showed that there were consistent log-linear trends in performance across multiple tasks as model size increased. Confirming that by making a model larger, researchers can improve performance across various tasks. Encouraged by these results, researchers continued to add parameters to their models to capitalise on the benefits of larger models performing better. GPT-4 introduced significant advancements by incorporating a multimodal approach, accepting both text and image inputs, and significantly improving performance on a wide range of NLP benchmarks through enhanced model architecture and training techniques

(OpenAI, 2023). This extensive training allows LLMs to generate text that is contextually relevant, coherent, and often indistinguishable from human-written content.

2.3 Introduction to Prompt Engineering

One technique used to obtain desired responses from LLMs is through prompt engineering, a prompt is the textual input usually a set of instructions or a question provided by users to guide output from a LLM (Amatriain, 2024). The prompt can influence the interactions and output generated from the LLM, it sets the boundaries for the conversation and guides the LLM on what the desired output should be (White et al., 2023). An analogy proposed is that we can think of an LLM as a (fuzzy) database and the prompt as a query helps explain why minor modifications in the query can result in significant variations in the output (Kaddour et al., 2023). As a result, both the phrasing and the sequence of examples provided in a prompt have been observed to impact the model's behaviour (Webson & Pavlick, 2021).

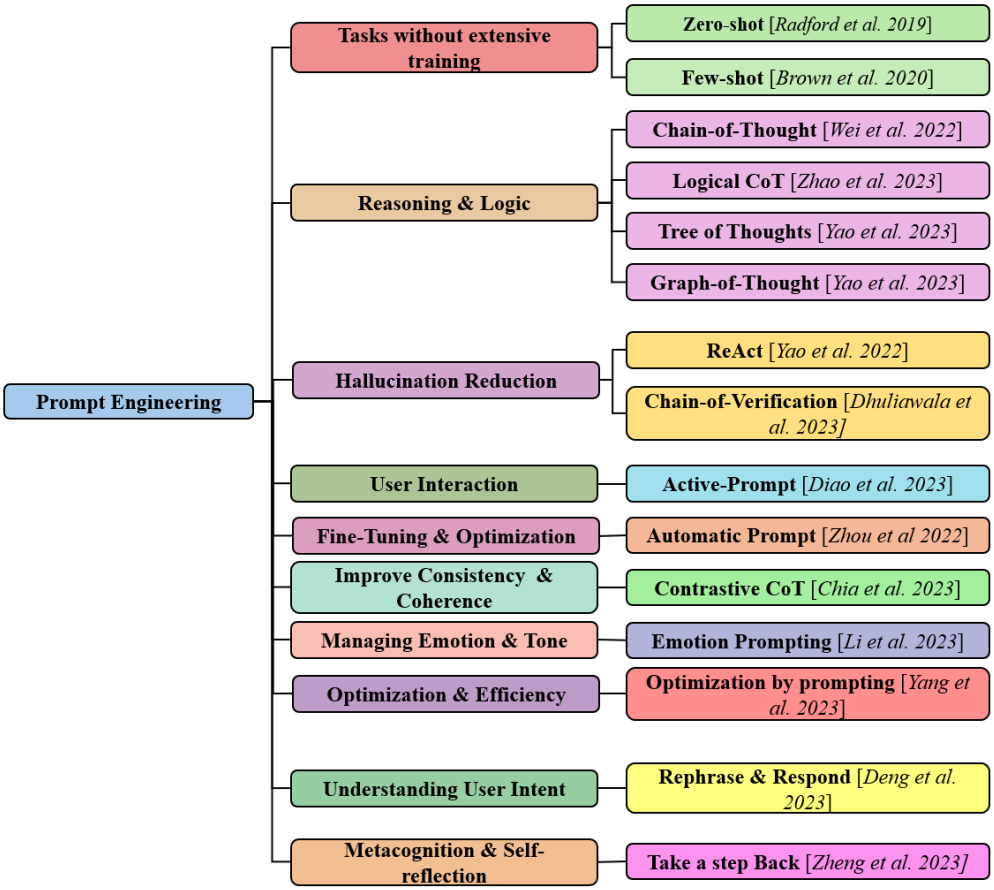


Figure 2. Comprehensive Classification of Prompt Engineering Strategies in LLMs, Categorized by Application Domains for Enhanced Customization and Adaptability (Adapted from Sahoo et al. 2023).

As demonstrated in Figure 2., there are numerous prompting techniques, the field of prompt engineering involves considerable experimentation. Despite achieving superior empirical outcomes (Wei et al., 2022), there is limited theoretical insight into why certain task phrasing is more effective, aside from achieving superior empirical outcomes. The next section will provide a brief overview of the current state of prompt engineering.

2.3.1 Overview of Prompting Frameworks

Prompt engineering frameworks have advanced to include a diverse array of techniques designed to tailor inputs to large language models (LLMs) for optimal results. In this section, we categorise these frameworks into different sub-categories.

Simple Frameworks - These frameworks rely on minimal input to guide the model, either leveraging the model's inherent knowledge or providing a few examples to improve performance. For example, see Figure 3:

i) *Zero-Shot Prompting:* Relies on the model's pre-existing knowledge to generate responses without providing specific examples during the task prompt. The model is expected to understand the task and generate appropriate outputs solely based on the given instructions or query (Radford et al., 2019).

ii) *Few-Shot Prompting:* Provides the model with a few examples to help it understand the task (Brown et al., 2020). This method enhances the model's ability to generalize from a small number of provided examples, improving its performance on the task by learning from these demonstrations.

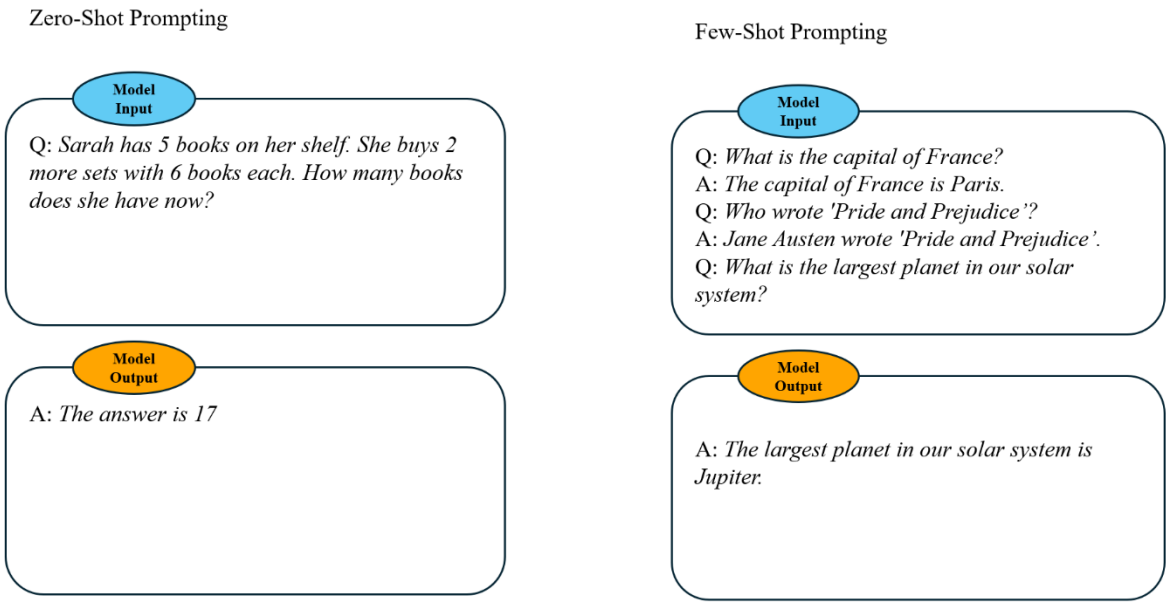


Figure 3. Example inputs and outputs from simple frameworks.

Reasoning and Logic - These frameworks rely on minimal input to guide the model, either leveraging the model's inherent knowledge or providing a few examples to improve performance. For example, see Figure 4:

i) *Chain-of-Thought (CoT) Prompting:* Encourages models to produce intermediate reasoning steps before arriving at a final answer (Wei et al., 2022). This approach helps in breaking down complex problems into manageable steps, making the reasoning process more transparent and improving the model's accuracy in tasks requiring logical deductions.

ii) *Self-Consistency:* Generates multiple reasoning paths and selects the most consistent answer through majority voting (X. Wang et al., 2022). This technique ensures that the model's eventual answer is the most reliable one by comparing various generated responses and choosing the one that appears most frequently.

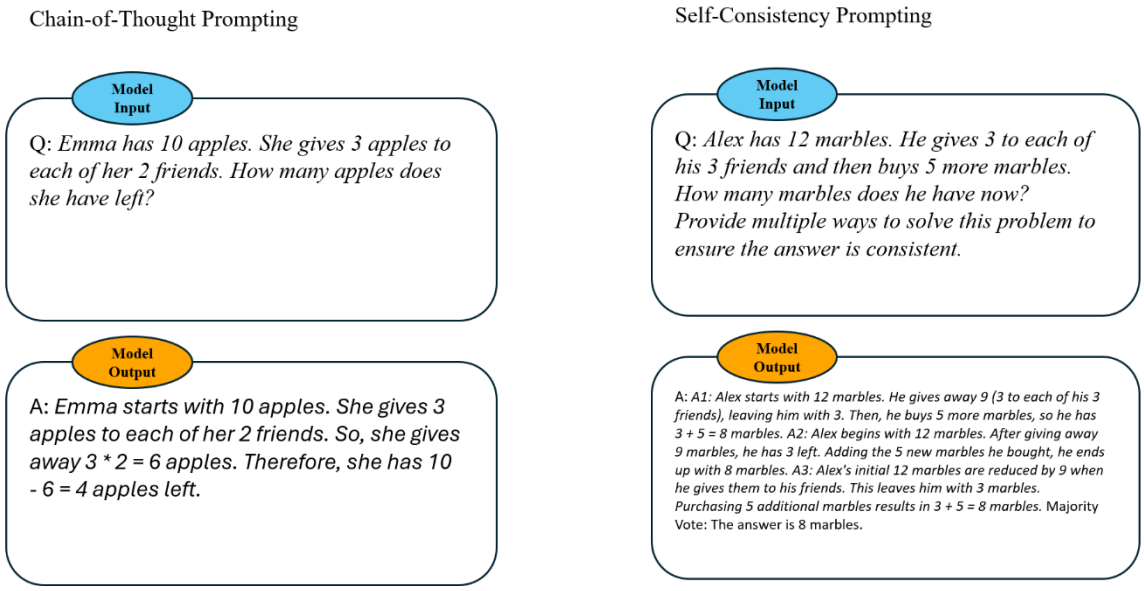


Figure 4. Example inputs and outputs from Reasoning and Logic frameworks.

Reduce Hallucination - These frameworks are designed to minimise the occurrence of false or fabricated information in the model's outputs, ensuring more accurate and reliable responses. For example, see Figure 5:

i) ReAct Prompting: Combines reasoning and action generation to enhance model performance in decision-making tasks. This method allows the model to generate reasoning traces and corresponding actions concurrently, facilitating more effective decision-making processes (Yao et al., 2022).

ii) Chain-of-Verification (CoVe) Prompting: Dhuliawala et al. (2023) Proposed a system whereby each step of the model's reasoning is systematically verified to reduce logical errors and hallucinations. This approach involves a multi-step verification process that ensures the coherence and accuracy of the model's final output.

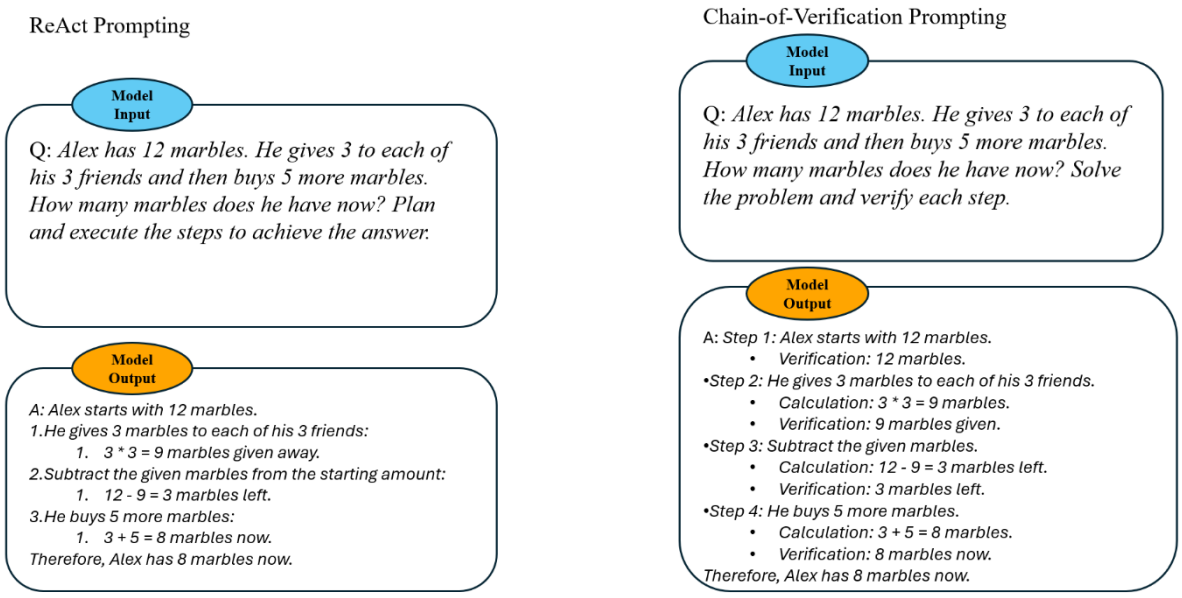


Figure 5. Example inputs and outputs from frameworks aiming to reduce hallucination

The quality of a prompt plays a crucial role in determining the output quality and reliability of LLMs. The structure of a prompt can significantly influence the model's response. Zhu et al. (2023) demonstrated that slight variations in prompts, such as minor typos or synonym replacements, can cause large variations in LLM output. Their research introduced PromptBench, a robustness benchmark designed to measure how LLMs respond to adversarial prompts. They investigated how an LLM would respond if there was an incorrect word within a prompt. They found that even a single incorrect word can lead to a 33% performance drop across various tasks. This sensitivity highlights the importance of precise prompt design to ensure consistent and accurate model performance. The research showed that adversarial prompts would redirect the model's attention away from the most relevant parts of the input resulting in incorrect or less accurate outputs. This concept is referred to as Prompt Brittleness, where 'Variations of the prompt syntax, often occurring in different ways unintuitive to humans, can result in dramatic output changes.' (Kaddour et al., 2023). Creating LLMs that are resilient to various prompting styles and formats remains an unresolved challenge, leading individuals to design prompts through experimentation. The

impact of prompting on LLM output cannot be overstated. Effective prompt engineering is critical for harnessing the full potential of LLMs, ensuring that they generate reliable, accurate, and contextually appropriate responses. As research continues to uncover the nuances of prompt sensitivity and robustness, the development of more sophisticated prompting strategies will be key to advancing the capabilities and applications of LLMs. The first aspect of this paper will look to establish which prompting framework delivers the most reliable results in generating a data-to-text marketing insight.

2.4 LLMs as text evaluators

Having assurances over the quality of the text being produced is an important aspect of developing an adequate data to text system. One way to do this is to use the LLM as an evaluator to give a judgement on the text that has been produced. However, using traditional NLG evaluation metrics such as BLEU (Papineni et al., 2001) and ROGUE (Lin, 2004) have been shown to have low correlation with human judgments (Sulem et al. 2018). So just passing these frameworks to LLMs and expecting it to produce evaluations similar to humans is not likely to work.

Furthermore, research on the validity and reliability of LLMs as NLG evaluators is still sparse and, in some cases, have been shown to have low correlation with human judgments (Zhong et al., 2022). Therefore, there is a need to develop more effective and reliable frameworks to use LLMs as successful NLG evaluators (Liu et al., 2023). This will be the second aspect of this paper, creating a framework for the LLMs to act as evaluators, then we will take these evaluations and compare it with a human judgment to establish if using LLMs to evaluate the text produced is a viable solution.

2.5 Traditional NLG systems vs LLMs

While traditional NLG systems have the benefit of not needing exceptionally large datasets to be trained on, they fall short in handling the complexities and scale of modern applications. They rely heavily on skilled experts to create and maintain extensive rule sets, struggle with scalability, and face significant challenges with the inherent ambiguity of natural language. Given the technological advancements in recent years, developing a data-to-text system using LLMs is the only logical solution. LLMs offer versatility and automation, excelling in tasks such as content creation, translation and summarisation. Their ability to comprehend and maintain context over long sequences of text ensures coherent and relevant outputs, a crucial feature for generating meaningful and accurate text from data. LLMs are highly adaptable and can be fine-tuned for specific use cases, significantly enhancing their performance across various applications. Additionally, LLMs possess multilingual capabilities, enabling them to process multiple languages, which broadens their applicability on a global scale. Their rapid response times make them suitable for real-time applications. Despite concerns about biases and resource intensity, advancements in AI research are continually addressing these issues, making LLMs more ethical and efficient.

3 Methodology

This section outlines the steps taken to develop a data-to-text framework. Firstly, we will discuss the data used within this project, then, the project management approach taken in this project and how it was used to achieve the goals set out by the company. Provide a brief overview of the systems in place before the commencement of this project, setting the scene for how the new framework integrates and enhances the existing infrastructure.

3.1 Data

The data involved in this project comes from various online providers through the usage of their APIs. The data then transformed into a format that is suitable to the tables set up by the company, and then aggregated into different time periods, be it daily, weekly or monthly. These data engineering steps were handled by a separate team within the company. For the data-to-text pipeline, the main task was to have correct queries that were suitable for the insight being produced.

3.2 Project Management and Implementation Strategy

This project was conducted with a company that had an established software platform, with customer paying license fees. The primary focus was to create a production-ready data-to-text system that would provide actionable insights from the various metrics tracked on the platform. To achieve this, a flexible approach was necessary, balancing new functionalities with existing systems. Rather than building from scratch, the strategy involved leveraging and enhancing existing infrastructure. This approach ensured we could build on a solid foundation, efficiently meet deadlines, and deliver a practical solution that met both technical and business objectives.

The development process of this project used the Cross Industry Standard Process for Data Mining (CRISP) approach (Wirth & Hipp, 2000), recognised as the industry standard for developing data products. The CRISP-DM framework, consisting of six stages, it was adapted to suit the needs of this project, which focused on prompt engineering rather than model building. This modified methodology included the following phases:

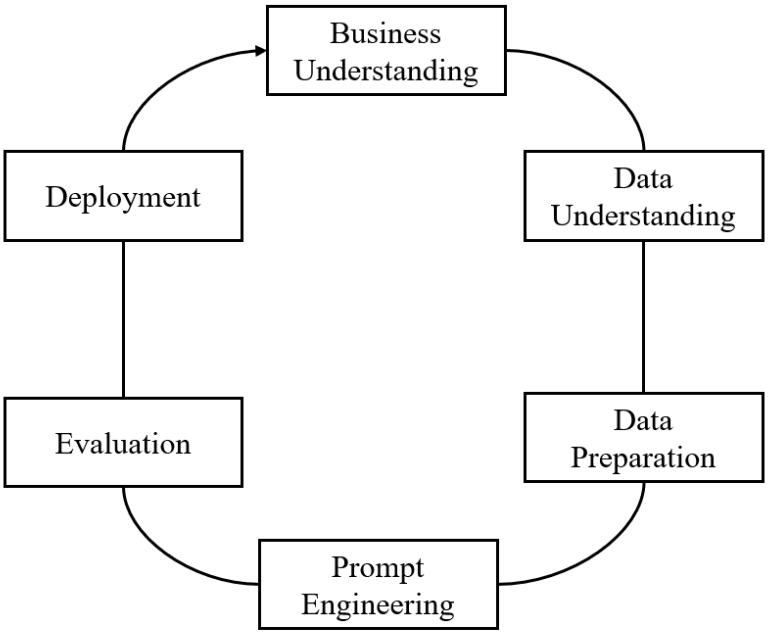


Figure 6. (Adapted) CRISP-DM Framework

In the initial phase, *business understanding*, the focus was on understanding project objectives and requirements from a business perspective. This involved defining the problem, setting project goals, and determining success criteria. Once business objectives were clear, the next step was *data understanding,* which ensured a comprehensive grasp of the available data. Following this, the *data preparation* phase began, where the data was transformed and formatted in a way that was suitable to generate the insights at either a weekly or a monthly level.

The core phase, *prompt engineering*, replaced the traditional modelling phase. Various prompt strategies were designed and evaluated, involving the creation of different prompt configurations, experimenting with different frameworks, iteratively refining the prompts to improve performance. The goal was to identify the most effective prompts that produced accurate and coherent text outputs. Following prompt engineering, the *evaluation* phase, where we assessed the results to establish which framework met the business objectives most convincingly. This involves testing and validation to gauge the effectiveness of the prompts. Evaluation criteria may include accuracy, coherence, and relevance of the generated text.

Finally, the *deployment* phase involves implementing the engineered prompts in a production environment. This includes integrating the prompts within business processes and monitoring their performance over time. The deployment phase ensures that the insights gained from the prompt engineering are actionable and beneficial to the organisation. The iterative nature of this adapted framework allows for revisiting and refining previous steps, ensuring flexibility and continuous improvement throughout the project lifecycle. This comprehensive approach makes it well-suited for developing a data-to-text system through prompt engineering.

3.2.1 Key Requirements

To ensure that the automated insights meet the needs of the business and its clients, several key requirements must be addressed. These requirements will form the base of the evaluation criteria that will be introduced later. The insights must be accurate, containing correct data and reflecting true metrics without errors, necessitating reliable data sources and a robust data integration process. The insights should be highly informative, presenting data and providing context, helping users understand the broader implications of changing metrics. This means interpreting trends and offering insights that explain the data’s significance for the business.

The system should be adaptable, capable of generating insights over various time periods (weekly, monthly, or annually), and versatile enough to cover several types of analysis, including brand-specific performance and brand versus competitor comparisons. Timeliness is also essential; the system must produce insights promptly to ensure clients have access to the latest information for decision-making. They should be presented in a clear, user-friendly format, allowing users to quickly extract necessary information. Scalability is vital, ensuring the system can handle varying volumes of data and generate insights effectively, regardless of the dataset size. By meeting these requirements, the automated insight generation system will provide valuable, accurate, and timely insights, enhancing the decision-making process for clients and helping their businesses maintain its competitive edge.

3.3 Initial Development Process

The collaborating company emphasised the necessity of identifying a solution quickly, which guided the initial phase of work. Preliminary testing by the company to develop an automated insight generation system revealed significant deficiencies, including incorrect data, mislabelled metric changes, and inappropriate language. These issues necessitated substantial manual intervention, thereby negating the benefits of automation. Initially a thorough evaluation of the existing system to pinpoint areas for improvement was required. To achieve this, it was essential to conduct a full cycle of the CRISP-DM framework. This approach facilitated a structured assessment of the project's current state and identified critical areas needing enhancement.

During the *business understanding* phase, it was observed that the company requested weekly summaries but were providing the incorrect level of data, resulting in confusion and the selection of incorrect metrics. In the *data understanding* phase, the model demonstrated substantial difficulties in contextual interpretation, misinterpreting metrics. Additionally, the

model occasionally generated sentences based on data points with missing values, leading to inaccurate reports. In the *data preparation* phase, efforts were made to restructure data input, ensuring the model received only the relevant week's data for generating summaries, thereby mitigating confusion caused by data overload.

Having addressed deficiencies in the automated report generation system, a functional weekly commentary for each client was achieved. The reports generated were satisfactory and met the requirements of our clients. However, we wanted to explore the possibility of making further improvements to enhance the overall quality and efficiency of the reports. This is where we entered the prompt engineering phase of the cycle where we tried to improve the system further by systematically examining the effect of different prompting techniques. The reports were being generated using GPT-4o, one of the most expensive LLMs available. Given the high costs, it was crucial to determine whether the expense was justified by the higher quality of the reports. To find a more cost-effective solution while maintaining or improving report quality, we conducted a series of systematic steps in prompt engineering and model evaluation.

3.4 Prompt Engineering Phase

To optimise the report generation process and achieve better results whilst attempting to achieve minimal computational expense, we ran a systematic prompt engineering experiment, focusing on testing different frameworks for prompting. The Chain of Thought framework was chosen for its ability to guide the model through a logical reasoning process, enhancing its capacity to generate coherent and contextually accurate outputs (Wei et al., 2022). Few-Shot prompting was implemented to provide the model with examples, improving its performance by demonstrating the desired output format and reducing the need for extensive training data (Brown et al., 2020). ReAct prompting fostered a more dynamic and interactive engagement with the model, allowing for adaptive responses based on real-time feedback (Yao et al., 2022). These methods were iteratively tested and refined through a continuous feedback loop, ensuring that the prompts were systematically optimised to deliver high-quality outputs from cost-effective models.

In our experiment, we tackled both a simple task and a complex task to fully assess the capabilities and robustness of our prompting strategies. The simple task involved generating weekly insights, which provided a baseline for measuring the model's performance on routine and repetitive data synthesis. The complex task entailed generating competitor insights over a monthly period, which required a more in-depth analysis and the ability to synthesize diverse data points into comprehensive reports.

Analysing larger quantities of data for the complex task introduced additional challenges, as the increased volume and variety of data heightened the potential for errors and inconsistencies. This necessitated more sophisticated prompting strategies to manage these intricacies and ensure accurate, reliable outputs. By addressing both tasks, we validated that our methods could effectively handle varying levels of complexity while maintaining high-quality results. A detailed depiction of the experiment can be seen in Figure 8.

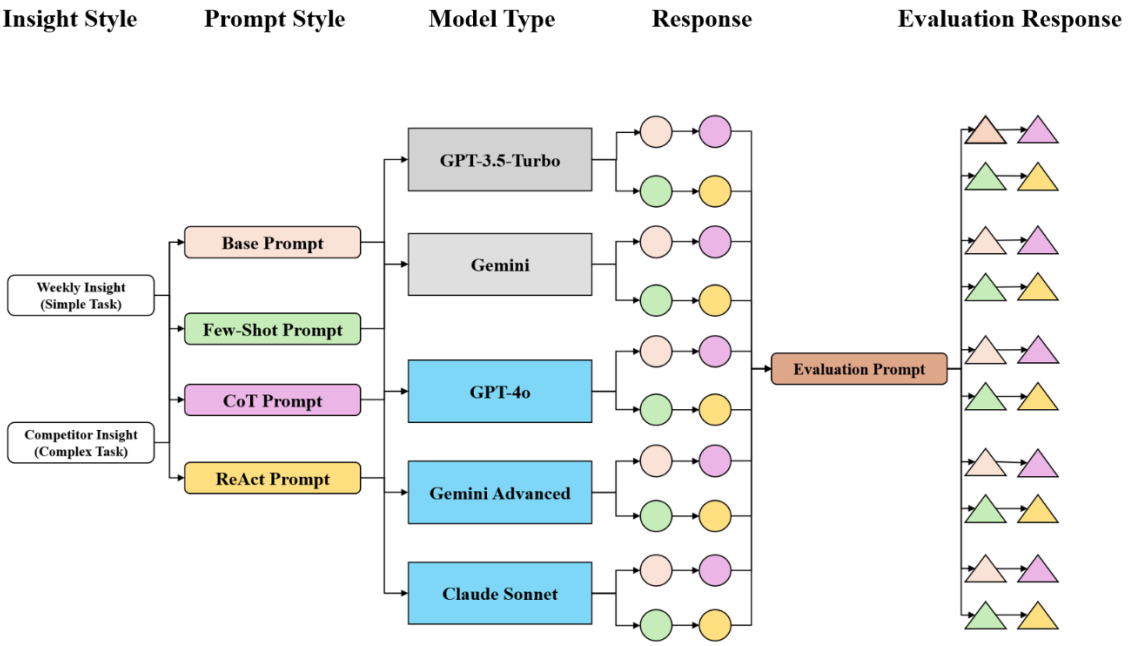


Figure 7. Visual breakdown of the experimental process, showing the flow from framework, to model, to response, to overall score generated.

3.4.1 Model Comparison

The initial model selected for this project was GPT-4o, which is recognised as one of the most advanced LLMs currently available. However, its operational cost is higher than alternatives. To explore more cost-effective alternatives without compromising performance, an evaluation of the alternative models was conducted. As shown in Figure 8. The models selected for comparison were GPT-3.5-Turbo and Gemini, classified as Base models

(indicated by the grey box) because they are the core LLM offerings from OpenAI and Google.

We then compared these to their most advanced alternatives GPT-4o and Gemini Advanced. Additionally, we included Claude Sonnet in the comparison. These three LLMs: GPT-4o, Gemini Advanced, and Claude Sonnet constitute what we refer to as the ‘Advanced models,’ indicated by the blue boxes in Figure 8., representing the current state-of-the-art offerings. This assessment aimed to determine whether optimised prompting techniques could enable smaller, less expensive models to perform comparably to the larger, costlier ones. By systematically comparing these models, we sought to identify a balance between performance and operational cost, potentially allowing for more efficient resource utilization without significant loss of functionality.

3.5 Evaluation Metrics

To ensure the accuracy and quality of the generated reports, an additional evaluation step was integrated into the pipeline. Traditionally, reports were generated and then manually checked before publication. To streamline this process and assess whether LLMs could function as effective NLG evaluators, we introduced a secondary evaluation phase utilising an LLM API as can be seen in Figure 9. For this purpose, we developed the CLEAR framework, a set of criteria specifically designed to evaluate the generated insights. The evaluation prompt included the initial prompt and response, along with the CLEAR criteria to guide the secondary model in its assessment. This step also aimed to determine if the evaluations made by LLMs would align with human evaluations.

The CLEAR framework encompasses **C**ompleteness, **L**ogical Flow & Consistency, **E**ngaging & Insightful, **A**ccuracy and **R**elevance & Formatting. Completeness and Accuracy involve

verifying that each data point aligns with the input data and is organised so that it fits into the narrative being created. Logical Flow and Consistency ensure clear reporting, sequential comparison of metrics, maintenance of a logical progression and consistent use of terminology. Engagement and Insights focus on the depth of insights provided, highlighting significant points, employing professional language, engaging the reader, and maintaining a consistent tone throughout the report. Comparison of Changes emphasizes the accurate labelling of positive and negative changes, describing the magnitude of changes accurately as well as using appropriate numerical abbreviations. Lastly, Relevance and Formatting ensure that the report is well-formatted and contextually relevant.

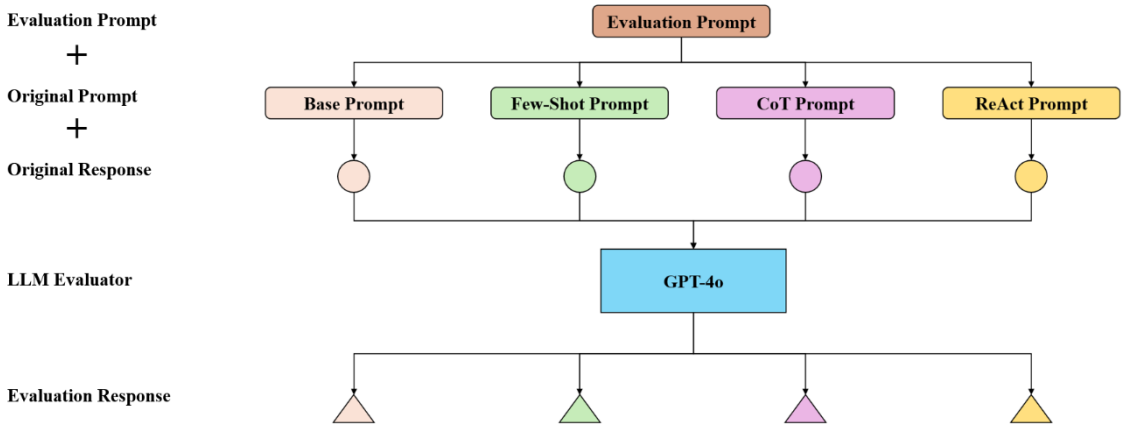


Figure 8. Visual representation of the evaluation process.

As shown in Figure 9. the second LLM call acted as an 'evaluator,' it would be sent the evaluation prompt, the original prompt and the original response and score each aspect of the CLEAR framework. The combined scores from the CLEAR framework would be given an ‘Overall Score’, with a maximum of 10 points. If the insight met the publishing threshold of 9 out of 10, it would recommend it be published; if not, it would provide a recommendation to be regenerated. Developing the evaluation prompt and the CLEAR framework required careful consideration to ensure they effectively guided the LLM in its assessment role. This

automated evaluation step aimed to reduce the reliance on manual checking, increase efficiency, and maintain high standards of insight quality.

3.6 Evaluation Verification

As previously mentioned, the use of LLMs as evaluators for NLG tasks is relatively unproven (Liu et al., 2023), however, they potentially offer innovative methods to assess the quality of generated content. To test the reliability of the LLM evaluations, we incorporated a step involving human evaluators. This step was designed to see if LLM and humans were aligned in what they judged as a good quality insight.

To achieve this, we employed the Analytic Hierarchy Process (AHP), a structured technique for organising and analysing complex decisions. Participants were presented with pairwise comparisons of generated reports and asked to determine which response was superior based on criteria from the CLEAR framework. These pairwise comparisons allowed us to establish a rank order of the responses, which was then compared to the LLM evaluations.

The AHP process began with defining the evaluation criteria to ensure alignment with the CLEAR framework used by the LLMs. Participants were then presented with pairs of reports and asked to choose the better one based on these criteria. The results of these comparisons were analysed to establish a rank order of the responses, providing a robust method to validate the LLMs' evaluations. This approach not only confirmed the accuracy of the LLMs' assessments but also offered insights into the alignment between LLM evaluations and human judgments. Incorporating human verification added rigor to the evaluation process by

determining whether LLMs' assessments were aligned with human evaluations of generated insights.

3.6.1 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) is a structured decision-making framework developed by Saaty (1987). It is designed to handle complex decision problems by breaking them down into a hierarchy of simpler sub-problems. AHP is particularly effective in situations where decisions need to be made based on multiple criteria, making it an ideal method for evaluating and comparing generated reports in this study. When asking participants to compare multiple outputs, it can be challenging for them to retain and evaluate numerous options simultaneously. AHP mitigates this difficulty by breaking down the decision-making process, reducing the cognitive load (Bramley & Oates, 2010) into manageable pairwise comparisons. By asking participants to compare only two reports at a time, AHP facilitates a more focused and accurate assessment (Bozóki et al., 2013), allowing us to determine their rank order systematically. AHP is a measurement used to derive preference scales from paired comparisons. It is very useful in multi-criteria decision-making decisions. It allows for a comparison between many options, making it a versatile tool in diverse decision-making scenarios.

3.6.2 How AHP Works

AHP begins with a hierarchy construction, where the decision problem is broken down into a hierarchy as shown in Figure 10. consisting of an overall goal at the top, followed by criteria and sub-criteria, and finally the decision options at the bottom. This hierarchical structure simplifies the problem into manageable parts. Pairwise comparisons are conducted to evaluate the relative importance of each option and the performance of each alternative with

respect to these criteria. Participants express their preferences using a scale of relative importance, generating matrices that are then used to derive priority scales.

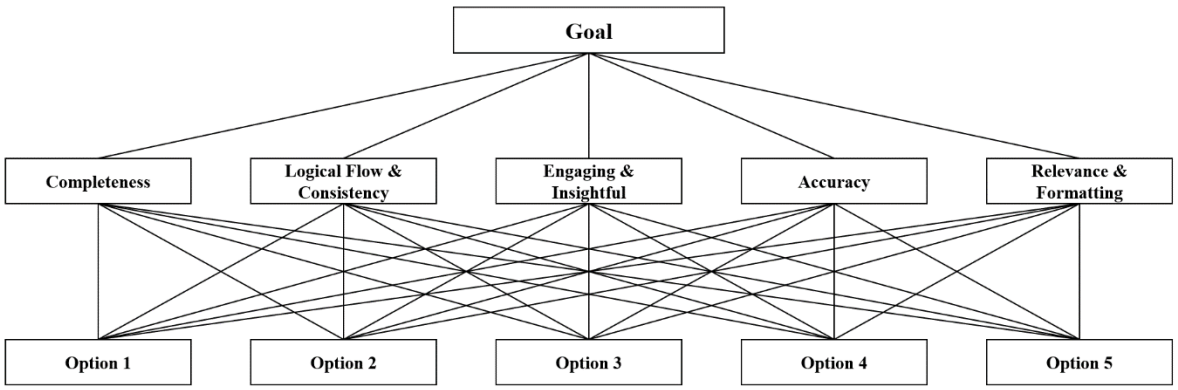


Figure 9. Visual representation of the Analytic Hierarchy Process

The pairwise comparisons, illustrated in Figure 11, demonstrate how you would organise the comparison of 5 alternatives, allowing each option to be evaluated against the others, enabling the calculation of priority weights. These weights are derived from the principal eigenvector of the comparison matrix. By aggregating these weights across all criteria, an overall score for each alternative is determined, resulting in a rank order based on participant preferences. This systematic approach allows participants to establish a rank order by considering only two options at a time.

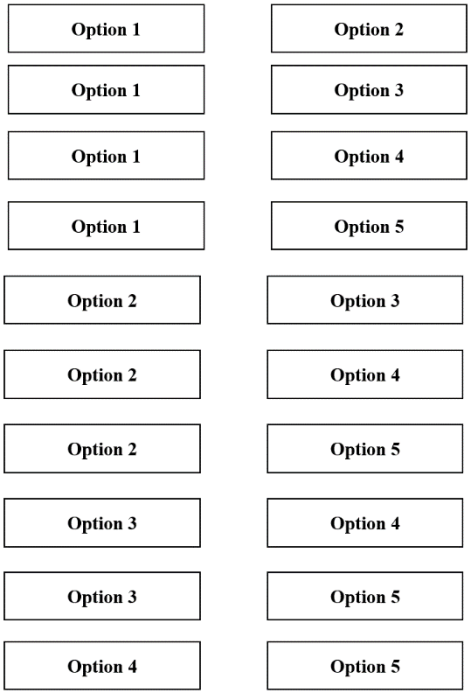


Figure 10. Visual representation of pairwise comparisons

In this experiment, AHP was used to corroborate the evaluations from LLMs by involving human evaluators in the assessment process. Participants were presented with pairwise comparisons of generated reports, and through these comparisons, we were able to establish a rank order of the responses. This approach not only provided a robust method to validate the LLMs' evaluations but also offered insights into the alignment between machine-generated evaluations and human judgments, thereby ensuring high standards of quality and reliability in the automated evaluation process.

4 Results

The results of this study are presented in three sections, each addressing a critical aspect of the evaluation process. This analysis aims to understand the performance and effectiveness of various prompt engineering techniques and language models in automated insight generation. First, we assess the overall quality of the generated reports based on key metrics: overall score, response length, lexical diversity, and consistency. We also analyse the correlation between prompt/response length and quality score. Next, we compare LLM evaluations with human evaluations to validate the reliability of automated evaluations, using the Analytic Hierarchy Process (AHP) for detailed comparison. Finally, we address the cost differences between different prompt/model combinations, highlighting the trade-off between output quality and financial implications to offer insights into the cost-effectiveness of each approach.

4.1 LLM Evaluation of the Output

In this section, we present the average overall scores assigned by the LLM (GPT-4o). The average overall score represents the results of the LLM evaluating each response based on the criteria outlined in the CLEAR framework.

4.1.1 Weekly Insight Analysis

The analysis reveals several key insights into the performance of different models and prompting frameworks when generating weekly insights as shown in Figure 12. For all advanced models, (GPT-4o, Claude and Gemini Advanced), the ReAct framework led to the lowest standard deviation compared to other frameworks see Table 3., indicating a higher level of consistency in the generated outputs. However, for the simpler models like GPT-3.5

Turbo and Gemini, the ReAct framework resulted in a slightly higher standard deviation. Suggesting that the simpler models occasionally struggled to fully comprehend the more complex ReAct framework.

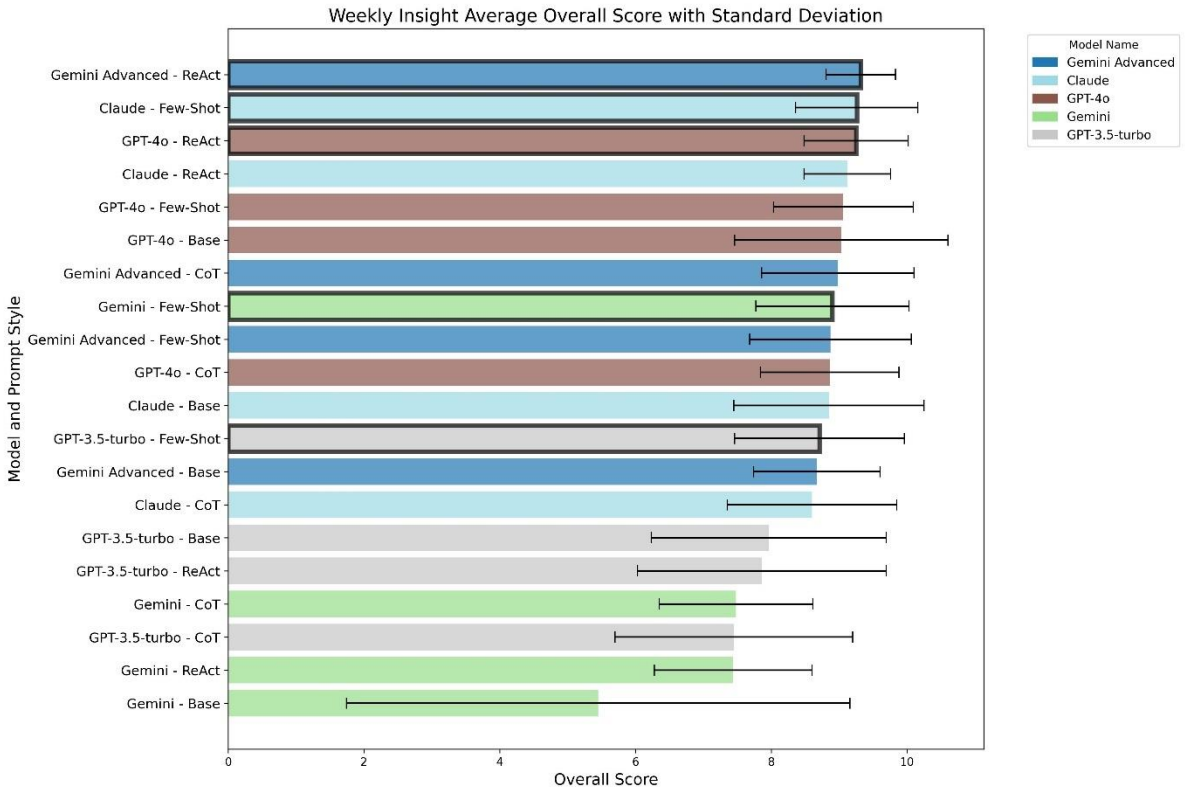


Figure 11. Weekly Insight Average Overall Score with Standard Deviation

For the Base models, the base prompt also had mixed results, possibly due to it being overly simplistic, resulting in lower average scores and higher standard deviations. However, there appears to be a sweet spot in prompting frameworks for these models, with Few-Shot prompting leading to improvements in both average overall score and standard deviation in GPT-3.5-turbo (8.71) and Gemini (8.9). Indicating that simpler models may benefitted most by receiving additional guidance by being shown an example of the expected output.

Table 3. Evaluation metrics for weekly insights

Model Name	*Prompt Style*	*Average Overall Score*	*Standard Deviation Overall Score*	*Average Response Length (Words)*	*Standard Deviation Response Length (Words)*	*Average Lexical Diversity*	*Standard Deviation Lexical Diversity*
GPT-4o	*Base*	9.03	1.57	178.30	38.01	0.75	0.04
GPT-4o	*CoT*	8.86	1.02	205.53	50.84	0.74	0.05
GPT-4o	*Few-Shot*	9.06	1.03	166.20	18.70	0.73	0.03
GPT-4o	*ReAct*	9.25	0.77	323.61	67.01	0.67	0.05
Claude	*Base*	8.85	1.40	133.55	26.15	0.79	0.04
Claude	*CoT*	8.60	1.25	115.90	23.89	0.79	0.04
Claude	*Few-Shot*	9.26	0.90	170.25	24.96	0.75	0.04
Claude	*ReAct*	9.12	0.64	166.80	41.44	0.78	0.06
Gemini Advanced	*Base*	8.67	0.93	111.14	24.12	0.78	0.05
Gemini Advanced	*CoT*	8.98	1.12	101.72	33.54	0.81	0.05
Gemini Advanced	*Few-Shot*	8.87	1.19	127.05	30.64	0.77	0.05
Gemini Advanced	*ReAct*	9.32	0.51	147.24	42.13	0.77	0.06
GPT-3.5-turbo	*Base*	7.96	1.73	129.45	31.96	0.76	0.06
GPT-3.5-turbo	*CoT*	7.45	1.75	103.10	30.31	0.78	0.06
GPT-3.5-turbo	*Few-Shot*	8.71	1.25	142.90	18.11	0.74	0.04
GPT-3.5-turbo	*ReAct*	7.86	1.83	151.40	72.32	0.74	0.11
Gemini	*Base*	5.45	3.71	61.90	36.77	0.9	0.07
Gemini	*CoT*	7.48	1.13	66.78	14.82	0.91	0.05
Gemini	*Few-Shot*	8.90	1.13	159.52	29.40	0.74	0.05
Gemini	*ReAct*	7.44	1.16	68.19	29.33	0.89	0.07

When examining average response length, the overall average response length was 141.53 words, which is where most model/framework combinations had response lengths. However, Gemini consistently produced shorter responses across three of the frameworks with lengths of 61.9 (Base), 66.78 (CoT), and 69.19 (ReAct). Interestingly, only the Few-Shot framework brought the response length for Gemini close to the average at 159.52 words. This reinforces the beneficial impact of Few-Shot prompting in enabling simpler models to generate more comprehensive responses.

Most model/framework combinations demonstrated consistent lexical diversity, with scores ranging from 0.67 to 0.81. Notably exceptions occurred in cases where models produced significantly shorter responses, such as the Base, CoT, and ReAct frameworks used with Gemini, which showed higher lexical diversity. This is expected with lower word counts, as shorter texts often have a relatively higher variety of words. Overall, lexical diversity remained consistent across different models and frameworks, indicating that prompt engineering techniques did not significantly impact the variety of language used.

The correlation matrix in Table 4. reveals several insightful relationships between various metrics. A positive correlation of 0.36 between overall score and response length indicates that longer responses tend to receive higher overall scores, suggesting that more detailed and comprehensive responses are perceived as higher quality. Conversely, a moderate negative correlation of -0.37 between overall score and lexical diversity implies that higher lexical diversity does not necessarily correlate with better performance. This is particularly evident in the Gemini model, which produced shorter responses with higher lexical diversity, leading to poorer overall scores. Additionally, the strong negative correlation of -0.79 between

response length and lexical diversity is expected, as shorter texts often exhibit a higher variety of words relative to their length.

Table 4. Correlation between Overall Score, Response Length (Words), Lexical Diversity and Total Cost

	Overall Score	*Response Length (Words)*	*Lexical Diversity*	*Total Cost*
Overall Score	1.00	0.36	-0.37	0.36
Response Length (Words)	0.36	1.00	-0.79	0.73
Lexical Diversity	-0.37	-0.79	1.00	-0.52
Total Cost	0.36	0.73	-0.52	1.00

The correlation between total cost and overall score 0.36 highlights that more expensive models tend to produce higher quality outputs, though the correlation is not very strong. The positive correlation between total cost and response length 0.73 further suggests that more costly models generate longer, more detailed responses. Finally, the negative correlation between total cost and lexical diversity -0.52 indicates that higher-cost models may produce responses with less varied vocabulary, possibly due to their focus on coherence and detail.

4.1.2 Competitor Insight Analysis

Analysing responses from the competitor insights provides an interesting insight in understanding how different LLMs coped with larger amounts of data. It allows us to understand how advanced prompting techniques could assist models in managing complex data more effectively. Interestingly, across all models apart from Claude, Few-shot prompting resulted in the highest average overall scores as seen in Figure 13. Suggesting that when producing complex outputs, LLMs receive benefit from being provided with examples of the desired output. Moreover, for all models except Claude, few-shot prompting also resulted in the lowest standard deviation across the four prompting frameworks (see Table. 5). This indicates that few-shot prompting not only improves average performance but also enhances consistency in the generated responses.

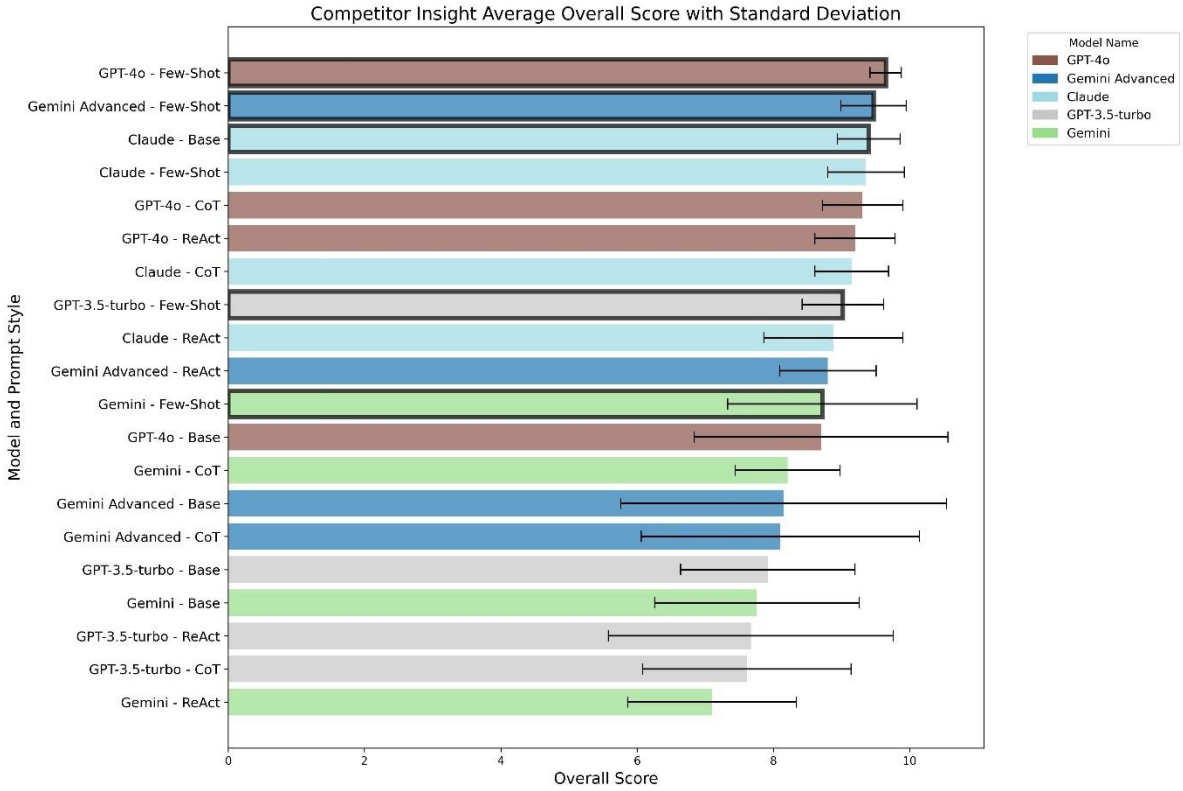


Figure 12. Competitor Insight Average Overall Score with Standard Deviation

When comparing the other prompting frameworks (Base, CoT, and ReAct), their performance within the models was quite similar. The range between the lowest and highest average overall scores for these frameworks was 0.61 for GPT-4o, 0.52 for Claude, 0.65 for Gemini Advanced, 0.31 for GPT-3.5 Turbo, and 0.66 for Gemini, demonstrating relatively minor variations in performance across different frameworks.

Table 5. Evaluation metrics for competitor insights

Model Name	*Prompt Style*	*Average Overall Score*	*Standard Deviation Overall Score*	*Average Response Length (Words)*	*Standard Deviation Response Length (Words)*	*Average Lexical Diversity*	*Standard Deviation Lexical Diversity*
GPT-4o	*Base*	8.70	1.86	207.82	27.37	0.72	0.04
GPT-4o	*CoT*	9.31	0.59	209.14	37.81	0.72	0.05
GPT-4o	*Few-Shot*	9.65	0.23	263.95	67.55	0.64	0.15
GPT-4o	*ReAct*	9.20	0.59	285.77	106.72	0.70	0.09
Claude	*Base*	9.40	0.46	195.29	32.12	0.73	0.05
Claude	*CoT*	9.15	0.54	165.38	61.5	0.67	0.23
Claude	*Few-Shot*	9.46	0.56	276.33	31.71	0.66	0.04
Claude	*ReAct*	8.88	1.02	151.88	60.98	0.75	0.21
Gemini Advanced	*Base*	8.15	2.39	132.12	44.07	0.74	0.20
Gemini Advanced	*CoT*	8.10	2.04	188.24	77.30	0.66	0.18
Gemini Advanced	*Few-Shot*	9.47	0.48	247.38	42.46	0.70	0.05
Gemini Advanced	*ReAct*	8.80	0.71	189.33	51.75	0.77	0.05
GPT-3.5 Turbo	*Base*	7.92	1.28	124.45	25.51	0.76	0.06
GPT-3.5 Turbo	*CoT*	7.61	1.53	97.14	25.76	0.79	0.06
GPT-3.5 Turbo	*Few-Shot*	9.02	0.60	205.41	29.74	0.68	0.05
GPT-3.5 Turbo	*ReAct*	7.67	2.09	109.28	62.01	0.76	0.15
Gemini	*Base*	7.76	1.50	75.75	16.93	0.90	0.05
Gemini	*CoT*	8.21	0.77	84.48	24.98	0.86	0.07

Gemini	*Few-Shot*	8.72	0.39	226.95	62.42	0.66	0.15
Gemini	*ReAct*	7.10	1.24	63.38	20.50	0.89	0.07

For average word length, the overall average response length across all model and framework combinations was 174.97 words. With the Advanced models being clustered around this average. Whereas the Base models typically produced shorter responses, however, just like with the weekly insights, few-shot prompting solved this. With the Base models (GPT-3.5 Turbo and Gemini) nearly doubling their average word count, aligning closer to the advanced models' outputs. A similar pattern was observed with lexical diversity. Most models and frameworks clustered around a lexical diversity range of 0.65 to 0.75, except for those responses where the word count was unusually low.

What can be seen from these results is that across both types of insight generation, Base models consistently performed better when provided with examples through Few-Shot prompting. Interestingly, as the task became more complex with competitor insights, the Advanced models also benefited from the Few-Shot as appose to the ReAct technique. Suggesting that being shown what is expected leads to better results than when breaking down the problem through CoT or ReAct. These findings demonstrate the effectiveness of Few-Shot prompting in enhancing performance and consistency across varying levels of model complexity for NLG tasks.

4.2 Analytic Hierarchy Process Results

In this section, we present the results of the Analytic Hierarchy Process (AHP), which evaluates the alignment between human and LLM assessments of the 'quality' of NLG-generated texts. The goal of this analysis was to determine if humans would assess the quality of insights similarly to LLMs, thereby evaluating the effectiveness of LLMs as NLG evaluators. Participants were shown five different insights and asked to evaluate ten pairs of these insights through pairwise comparisons.

Table 6. Details of Options for Analytic Hierarchy Process

Option Number	*Model*	*Framework*	*Overall Score*
1	Gemini Advanced	ReAct	8.4
2	Claude Few-Shot	Few-Shot	8.8
3	GPT-3.5 CoT	CoT	7.6
4	Gemini ReAct	ReAct	6.4
5	Gemini Few-Shot	Few-Shot	8.8

Options 1 and 2 represented Advanced models with high performance, Options 3 and 4 were Base models with lower performance, and Option 5, while coming from a Base, provided a good example of a Base model judged by the LLM to be high-quality. The AHP survey was sent to 20 people within the company and the results are presented in Table 7. Preference counts indicate how often each option was chosen by participants during pairwise comparisons, providing insight into participant preferences. Final scores, calculated based on these preference counts, offer a quantitative measure of each option's relative performance, allowing for ranking from highest to lowest performance accordingly.

Table 7. Preference counts and scores from AHP

Chosen Option	*Counts*	*Score*
2	66	0.825
1	60	0.750
5	53	0.650
3	18	0.225
4	4	0.050

The AHP analysis reveals a promising alignment between human assessments and LLM-generated scores in determining the quality of NLG outputs. Option 2 (Claude) received the highest preference from participants, with 33 counts and a score of 0.825, closely followed by Option 1 (Gemini Advanced) with 30 counts and a score of 0.750. These preferences align well with the LLM-assigned scores, where Claude and Gemini Advanced received high scores of 8.8 and 8.4, respectively. Option 5, despite coming from a Base model still provided a good response. It was the third ranked insight with 26 counts and a score of 0.650, and it also received a high score of 8.8 from the LLMs. This consistency indicates that both humans and LLMs similarly assess the quality of NLG insights, demonstrating the effectiveness of LLMs as evaluators of NLG outputs. Insights that received lower scores from the LLMs were also judged as lower quality by human evaluators. Option 3 (GPT-3.5) received only 9 counts and a score of 0.225, while Option 4 (Gemini) received just 2 counts and a score of 0.050. Correspondingly, the LLMs scored these options 7.6 and 6.4, respectively. This alignment in evaluating poor outputs further demonstrates potential reliability of using LLMs to assess generated insight quality.

Overall, these results suggest there is promising alignment between human assessments and LLM-generated scores in evaluating NLG outputs. This suggests that LLMs could be effectively used as NLG evaluators, providing reliable assessments of generated text quality. This has significant implications for the development and optimisation of NLG systems,

emphasising the potential of LLMs in both generating and evaluating high-quality textual outputs.

4.3 Cost Analysis

In this section, we begin our cost analysis by examining the differences in the costs associated with the Base models versus Advanced models. Figure 14. illustrates these cost differences, providing a clear comparison between the two types of models. This section attempts to determine whether the increased cost of the Advanced models is justified with higher quality output. By evaluating the cost-effectiveness of each model, we aim to provide insights into the economic implications of using different models for NLG tasks.

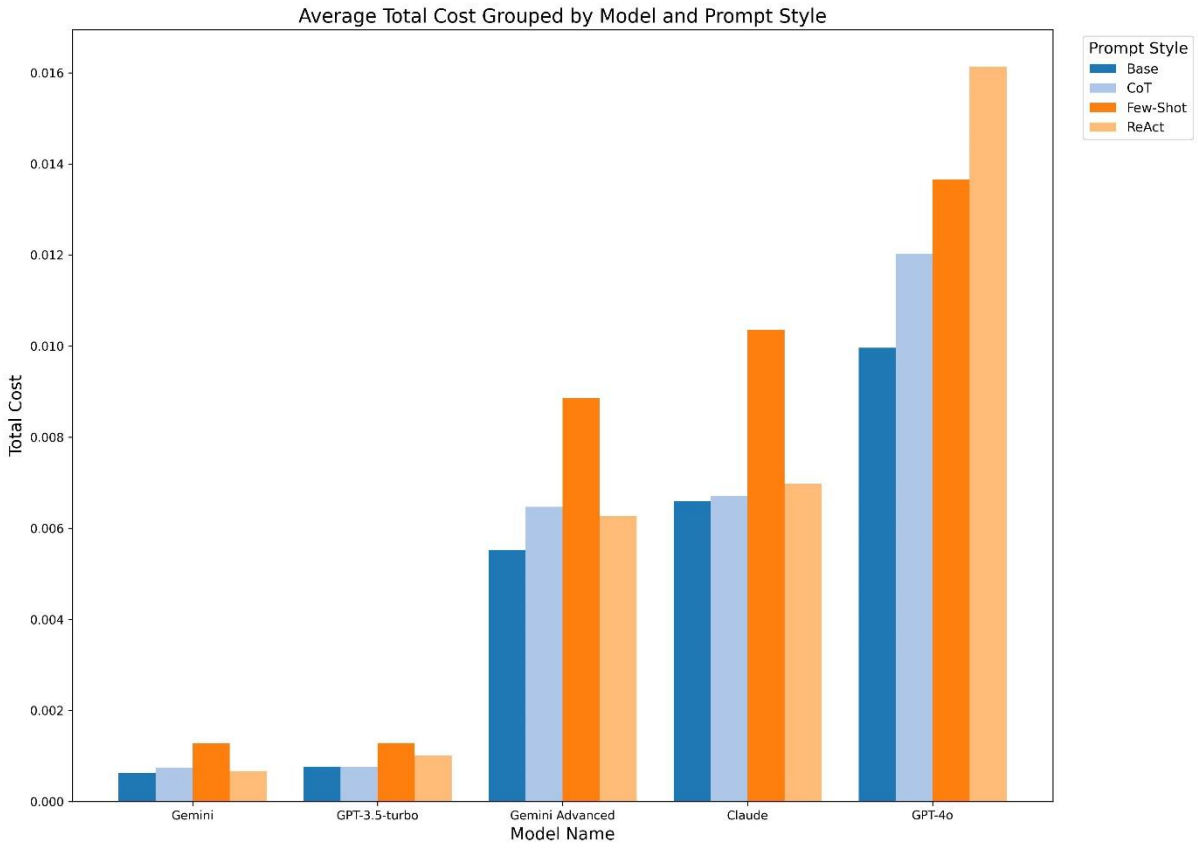


Figure 13. Average Total Cost Grouped by Model and Prompt Style

4.3.1 Cost Analysis by Model

Whilst it is unsurprising that GPT-4o emerges as the most expensive model, Table 8. shows that the two next cheapest models, Claude and Gemini Advanced are around, 30-40% cheaper. Furthermore, there are even cheaper alternatives such as Gemini and GPT 3.5-turbo which are the most cost-effective options analysed, they are around 90% cheaper than GPT-4o. The wide range is costs suggests there should be a critical evaluation of whether using the more expensive advanced models results in proportionately better marketing insights.

Table 8. Average Total Cost and Percentage Savings by Model

Model Name	*Average Total Cost ($)*	*Percentage Cheaper than GPT-4o*
GPT-4o	0.0304	0.00%
Claude	0.0200	34.15%
Gemini Advanced	0.0186	38.90%
GPT-3.5-turbo	0.0023	92.40%
Gemini	0.0022	92.65%

The scatter plots illustrate the relationship between total cost and overall score, separated by model type, for both Weekly (Figure. 15) and Competitor (Figure. 16) insights. In both plots, Advanced Models cluster at higher total costs and generally achieve higher overall scores more consistently. In contrast, Base Models exhibit lower total costs but show less consistency in their overall scores. While some Base Models achieve performance levels like Advanced Models, a notable cluster of their scores falls between 7 and 8.

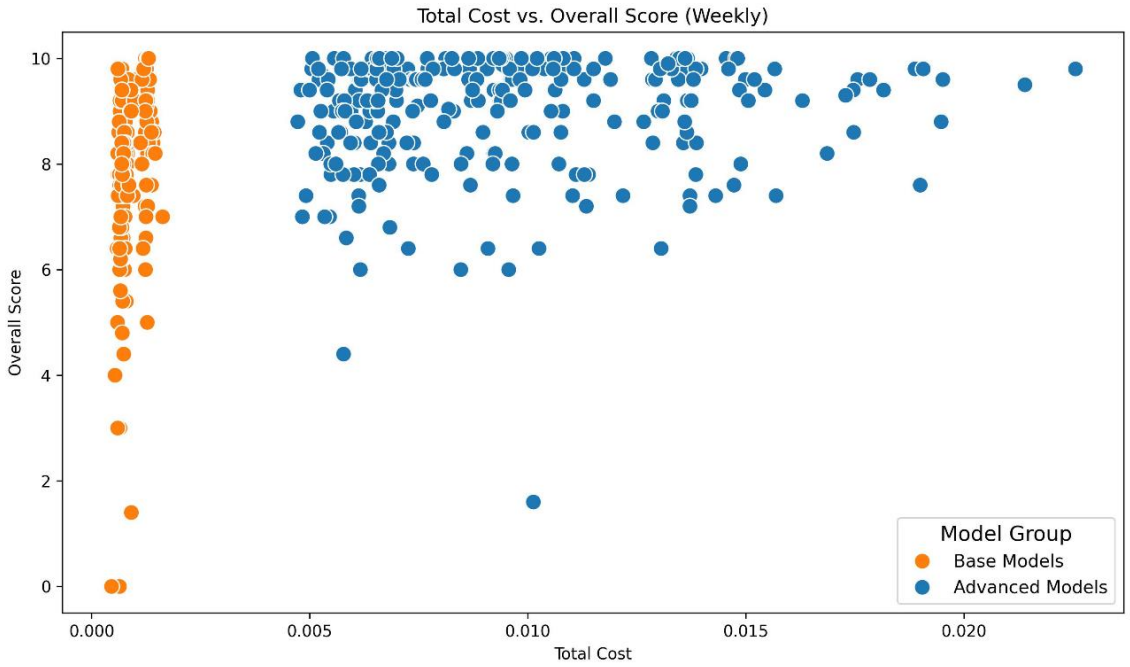


Figure 14. Scatter plot showing Total Cost against Overall Score (Weekly)

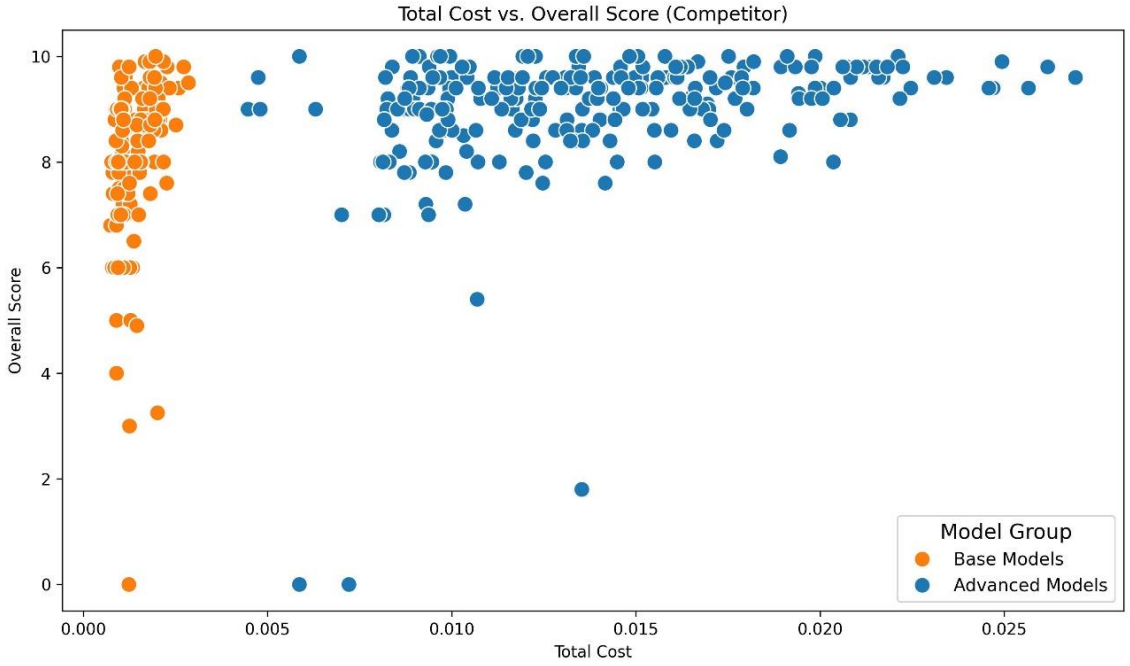


Figure 15. Scatter plot showing Total Cost against Overall Score (Competitor)

To determine if the difference in overall scores between model types is statistically significant, independent samples t-tests were conducted to compare the overall scores of Advanced Models and Base Models in both the Weekly and Competitor datasets. For the Weekly models, assuming equal variances, the t-test yielded a statistically significant difference in overall scores between the two groups, $t(198) = 8.71, p < .001$, $d = 0.86$. Similarly, for the Competitor models, the t-test also indicated a statistically significant difference, $t(198) = 7.79$, $p < .001$, $d = 0.79$. These results show that Advanced Models had significantly higher scores than Base Models in both weekly and competitor insights. Although the cost of Advanced Models is higher than that of the Base Models, this increased cost is accompanied by a noticeable benefit, as indicated by their significantly better performance and the large effect sizes.

4.3.2 Assessing the Impact of Prompt Engineering

Discovering that Advanced Models outperform Base Models is not a novel finding; however, it gives reliability to the evaluation process, as it confirms what we were expecting. With this confirmation, we shift focus to evaluating whether prompt engineering can mitigate this disparity. By analysing the impact of prompt engineering on each model's performance and comparing the overall scores between different prompt engineering techniques, we can determine if these techniques enhance performance significantly and narrow the gap between Advanced and Base Models. This step is crucial for understanding the potential of prompt engineering to improve cost-effectiveness. First, we examine the effect of prompt engineering on model performance for weekly insights, see Figure 17.

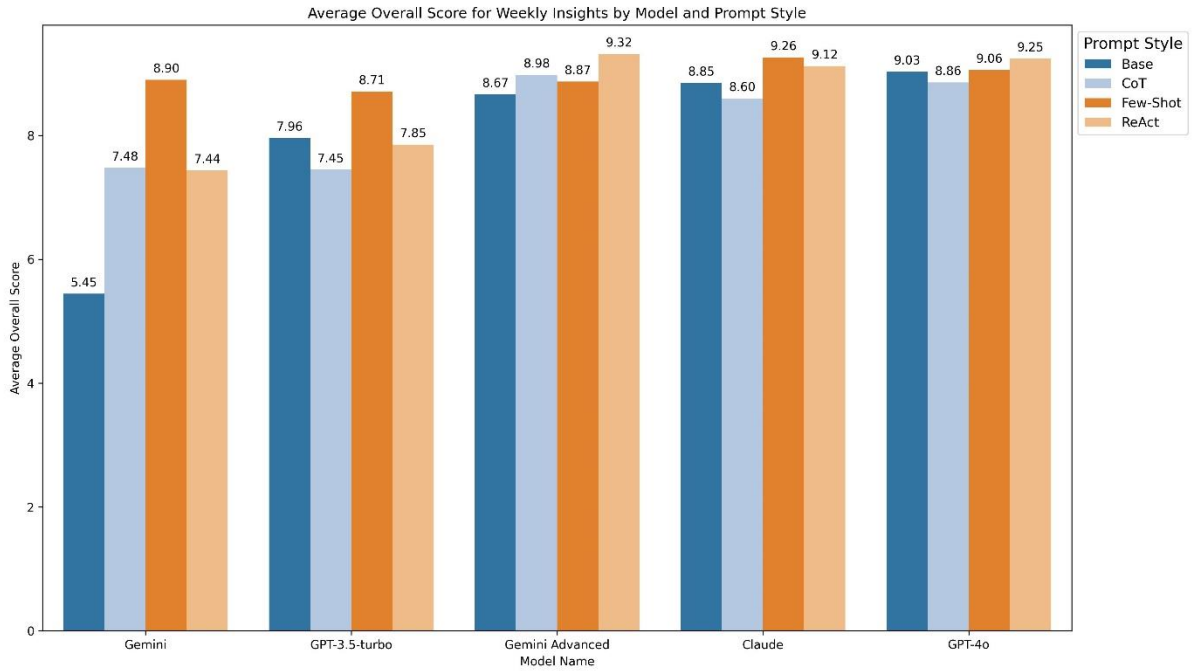


Figure 16. Average Overall Score for Weekly Insights by Model and Prompt Style

From Figure 17. we observe that for some models there is a noticeable difference in average overall scores in weekly insights within the models when they use different prompting techniques, with the base models showing more variation compared to the advanced models. To evaluate if different prompt frameworks led to statistically significant differences in overall scores within each model, an Analysis of Variance (ANOVA) was conducted. This analysis aimed to determine if variations in prompting techniques lead to statistically significant differences in model performance. By identifying these significant differences, the results could highlight which prompting technique is most effective in enhancing model performance.

For each model, a one-way ANOVA was performed to compare the overall scores across different prompt styles. If the ANOVA indicated significant differences (p-value < 0.05), Tukey's Honest Significant Difference (HSD) test was conducted to identify which specific prompt styles differed significantly from each other and the direction of these differences.

The results in Table 9. showed that for GPT-4o, Claude, Gemini Advanced, and GPT-3.5-turbo, the differences in overall scores among prompt styles were not statistically significant. However, for Gemini, significant differences were found among prompt styles $F(9.3170), p < .001$. Tukey's HSD test further identified the specific differences: the Base prompt scored significantly lower than CoT, Few-Shot, and ReAct styles, with mean differences of -2.03, -3.45, and -1.99 respectively. No significant differences were found between cot, few-shot, and react styles.

Table 9. ANOVA Results and Average Scores for Different Models (Weekly)

Model Name	*Average Overall Score (all prompt styles)*	*F-value*	*P-value*
GPT-4o	9.07	0.4354	0.7281
Claude	8.96	1.4453	0.2363
Gemini Advanced	8.96	1.6654	0.1814
GPT-3.5-turbo	7.99	2.0220	0.1179
Gemini	7.31	9.3170	< 0.001*

** Indicating statistical significance at the $p < 0.05$ level*

Surprisingly, prompt engineering only had a significant impact on the overall scores for weekly insights when using Gemini. However if we lookbat Figure 18. for competitor insights, we observe a larger disparity between the average scores of different prompt styles across multiple models.

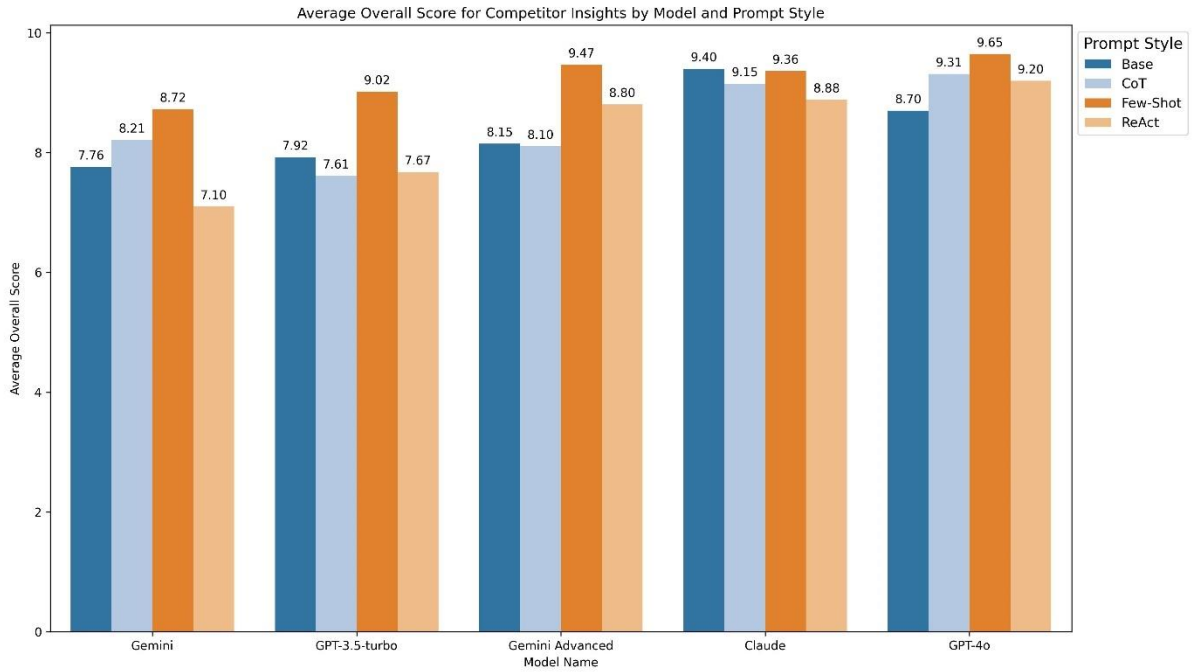


Figure 17. Average Overall Score for Competitor Insights by Model and Prompt Style

The ANOVA results for competitor insights as shown in Table 10. confirmed significant differences in overall scores between different prompt styles for several models. For Gemini, the ANOVA displayed multiple significant differences $F(6.4488)$, $p < .001$: the CoT style outperformed the ReAct style by 1.11 points, and the Few-Shot style exceeded the ReAct style by 1.62 points, indicating that the ReAct style consistently underperformed against alternatives. In the case of GPT-3.5-turbo, $F(4.4512)$, $p = 0.0061$, the Few-Shot style also proved superior to the CoT style, with a mean difference of 1.40. Similarly, Gemini Advanced, $F(3.3782)$, $p = 0.0226$, showed that the Few-Shot style significantly surpassed the CoT style by 1.36 points, demonstrating the effectiveness of the Few-Shot approach. For Claude, no significant differences were observed among the prompt styles, $F(2.4109)$, $p = 0.0734$. Finally, for GPT-4o, $F(3.2253)$, $p = 0.0269$, Tukey's HSD test revealed that the Few-Shot prompt style outperformed the Base style with a mean difference of 0.95.

Table 10. ANOVA Results (Competitor) and Average Scores for each model

Model Name	*Average Score*	*F-value*	*P-value*
GPT-4o	9.07	3.2253	0.0269*
Claude	8.96	2.4109	0.0734
Gemini Advanced	8.96	3.3782	0.0226*
GPT-3.5-turbo	7.99	4.4512	0.0061*
Gemini	7.31	6.4488	< .001*

** Indicating significance at the $p < 0.05$ level*

These findings suggest that prompt engineering significantly enhances overall scores, particularly when the task is more complex tasks such as when writing a competitor report. The differences in scores across various models and prompt styles indicate that task complexity magnifies the benefits of effective prompt engineering.

4.3.3 T-Test Comparison for top performing Base model frameworks

For both Weekly and Competitor insights, the Few-shot prompting framework yielded the highest average overall score for the base models, as shown in Figure 17. and Figure 18. After identifying this, we needed to evaluate whether the Base models using their highest-performing prompting framework produced significantly different overall scores compared to the Advanced models.

To evaluate the performance differences, t-tests were conducted to compare overall scores between the top-performing Base frameworks and the Advanced models. These tests aimed to determine whether the differences in overall score were statistically significant, thus justifying the additional cost of the advanced models. Additionally, Cohen's d was calculated to measure the effect size, indicating the magnitude of the observed differences. A positive Cohen's d indicates that the Base model outperformed the framework on the Advanced model, while a negative Cohen's d indicates that the framework used with the Advanced

model outperformed the Base model. By performing these t-tests and calculating Cohen's d, we can quantify performance differences between model type and assess the practical significance of selecting different models.

The results from the Weekly insights are presented in Table 11. for GPT-3.5-Turbo and in Table 12. for Gemini. Of the 24 comparisons made, there was only one instance of the Advanced model producing statistically significant better outputs than Few-shot framework on the Base models, that was when comparing GPT-3.5-Turbo to Gemini Advanced (React), other than that there were no other statistically significant differences.

Table 11. GPT-3.5-turbo Few-Shot (Weekly) T-test comparison with Advanced models

Advanced Model	*Framework*	*T-statistic*	*P-value*	*Mean Difference*	*Cohen's d*
GPT-4o	*Base*	-0.784	0.437	0.319	-0.217
GPT-4o	*CoT*	-0.393	0.697	0.149	-0.130
GPT-4o	*Few-shot*	-0.970	0.338	0.350	-0.307
GPT-4o	*React*	-1.936	0.058	0.535	-0.549
Gemini Advanced	*Base*	0.127	0.900	-0.043	0.040^
Gemini Advanced	*CoT*	-0.693	0.493	0.268	-0.225
Gemini Advanced	*Few-shot*	-0.424	0.674	0.161	-0.133
Gemini Advanced	*React*	-2.069	0.045*	0.609	-0.646
Claude	*Base*	-0.334	0.740	0.140	-0.106
Claude	*CoT*	0.279	0.782	-0.110	0.088^
Claude	*Few-shot*	-1.598	0.118	0.550	-0.505
Claude	*React*	-1.316	0.196	0.412	-0.416

** Advanced model outperforms base model ** Base model outperforms advanced model ^Base model outperforms advanced model without statistical significance*

However, there were instances as shown in Table 11. and Table 12. where the Base models outperformed the Advanced models, but these differences were not statistically significant, indicating that the Base models could occasionally produce better results. Overall, the Advanced models did not consistently provide significantly better weekly insights than the Base models when using the Few-shot framework.

Table 12. Gemini Few-Shot (Weekly) – T-test comparison with Advanced models

Advanced Model	*Framework*	*T-statistic*	*P-value*	*Mean Difference*	*Cohen's d*
GPT-4o	*Base*	-0.331	0.742	0.129	-0.090
GPT-4o	*CoT*	0.117	0.908	-0.041	0.038^
GPT-4o	*Few-shot*	-0.474	0.638	0.160	-0.148
GPT-4o	*React*	-1.335	0.188	0.345	-0.373
Gemini Advanced	*Base*	0.731	0.469	-0.233	0.226^
Gemini Advanced	*CoT*	-0.215	0.831	0.078	-0.069
Gemini Advanced	*Few-shot*	0.080	0.937	-0.029	0.025^
Gemini Advanced	*React*	-1.549	0.129	0.419	-0.478
Claude	*Base*	0.126	0.900	-0.050	0.039^
Claude	*CoT*	0.806	0.425	-0.300	0.252^
Claude	*Few-shot*	-1.122	0.269	0.360	-0.351
Claude	*React*	-0.805	0.900	0.222	-0.239

* Advanced model outperforms base model ** Base model outperforms advanced model ^Base model outperforms advanced model without statistical significance

These results suggest that, for the weekly insights (simple task) adequate prompt engineering can bridge the gap in performance between Base models and Advanced models, meaning that the additional cost of the advanced models is not necessary in this case. If we look at the results for the competitor insights (complex task), Table 13. reveals some interesting findings when evaluating the outputs for Competitor insights. Notably, the Few-shot technique on GPT-3.5-turbo outperformed Gemini Advanced when using the CoT technique. The performance difference was significant ($t = 2.015$, $p = 0.050$, $d = 0.615$), indicating that GPT-3.5-Turbo with the Few-shot technique achieved better results than Gemini Advanced with the CoT technique.

Table 13. GPT-3.5-Turbo Few-shot (Competitor) vs Advanced Models

Advanced Model	*Framework*	*T-statistic*	*P-value*	*Mean Difference*	*Cohen's d*
GPT-4o	*Base*	0.754	0.455	-0.318	0.244^
GPT-4o	*CoT*	-1.642	0.108	0.295	-0.495
GPT-4o	*Few-shot*	-4.543	0.000*	0.627	-1.370
GPT-4o	*React*	-1.007	0.320	0.182	-0.304
Gemini Advanced	*Base*	1.643	0.109	-0.868	0.540^
Gemini Advanced	*CoT*	2.015	0.050**	-0.913	0.615
Gemini Advanced	*Few-shot*	-2.688	0.010*	0.448	-0.820
Gemini Advanced	*React*	1.060	0.295	-0.213	0.324^
Claude	*Base*	-2.330	0.025*	0.382	-0.711
Claude	*CoT*	-0.768	0.447	0.134	-0.234
Claude	*Few-shot*	-1.931	0.060	0.344	-0.589
Claude	*React*	0.518	0.608	-0.136	0.167^

** Advanced model outperforms base model ** Base model outperforms advanced model ^Base model outperforms advanced model without statistical significance*

While one result showed a better output from the Base models, the Advanced models outperformed the best Base model frameworks on more occasions when generating competitor insights. Specifically, GPT-4o surpassed both Base models when using the Few-shot prompt technique. Gemini Advanced outperformed GPT-3.5 with the Few-shot prompt technique, while Claude outperformed GPT-3.5 using both the few-shot and base prompt styles. Additionally, the results in Table 14. show that Gemini Advanced performed better than Gemini when using the Few-shot prompt technique, and Claude outperformed Gemini with both the few-shot and base prompt styles. These results, indicated by p-values < 0.05, demonstrate that advanced models, particularly GPT-4o and Claude, consistently have an edge over base models in specific prompt styles.

Table 14. Gemini Few-Shot (Competitor) vs Advanced Models.

Advanced Model	*Framework*	*T-statistic*	*P-value*	*Mean Difference*	*Cohen's d*
GPT-4o	*Base*	0.041	0.968	-0.021	0.013^
GPT-4o	*CoT*	-1.835	0.074	0.592	-0.560
GPT-4o	*Few-shot*	-3.077	0.004*	0.924	-0.939
GPT-4o	*React*	-1.482	0.146	0.479	-0.452
Gemini Advanced	*Base*	0.915	0.367	-0.571	0.304^
Gemini Advanced	*CoT*	1.147	0.258	-0.617	0.354^
Gemini Advanced	*Few-shot*	-2.325	0.025*	0.745	-0.717
Gemini Advanced	*React*	-0.245	0.808	0.083	-0.075
Claude	*Base*	-2.127	0.040*	0.679	-0.656
Claude	*CoT*	-1.326	0.192	0.431	-0.409
Claude	*Few-shot*	-1.959	0.057	0.640	-0.605
Claude	*React*	-0.398	0.693	0.161	-0.130

** Advanced model outperforms base model ** Base model outperforms advanced model ^Base model outperforms advanced model without statistical significance*

Despite the Advanced models having the edge over the Base models on 6 occasions, the most common occurrence was no significant difference between them. For GPT-3.5, there was no significant difference in performance when compared to GPT-4o using the ReAct and CoT prompt styles, to Gemini Advanced using the ReAct prompt style, and to Claude using the ReAct and CoT prompt styles. Similarly, for Gemini, no significant difference in performance was observed when compared to GPT-4o using the ReAct, CoT, and Base prompt styles, to Gemini Advanced using the ReAct, CoT, and Base prompt styles, and to Claude using the ReAct and CoT prompt styles.

While these non-significant results indicate that, in many cases, the performance of Base and Advanced models is comparable. The Advanced models outperformed the Few-shot technique on the Base models 6 times more when generating competitor insights compared to weekly insights suggesting that the justification for the increased cost of Advanced models depends on the complexity of the NLG task at hand.

5 Conclusion

5.1 Concluding Remarks

The main purpose of this paper was to i) evaluate if prompt engineering could lead to improvements in the development of a data to text system using LLMs ii) analyse the cost benefit trade-off between using more expensive advanced models compared to their base model counterparts and iii) evaluate if LLMs could be successfully used as NLG evaluators.

From the results, we were able to show that in weekly insights (simple task) prompt engineering has a noticeable impact on the Base models but less of an impact on the advanced models. For the competitor insights (complex task), prompt engineering has a noticeable impact on quality of output in both types of models. After establishing the most successful prompting technique we were able to show that we can close the gap in quality of insights between base models and advanced models in 95% (23/24) of cases in a simple task and 70% (17/24) of cases in an advanced task to the point that there is no significant difference output between Base and Advanced models, or even see better outputs from the Base models. This result illustrates the benefit of adopting proper prompting techniques in and the potential in reducing costs by up to 90% when doing so.

Furthermore, after assessing the potential of using LLMs as NLG evaluators, the results suggested that LLMs align well with humans in their judgment of quality NLG output. And that when using the CLEAR framework, LLMs and humans are aligned in their assessment of NLG text.

5.2 Business Impact

A significant part of this paper was committed to assessing the cost-benefit trade-off between using Advanced models compared to the Base models, it is important to conclude and provide a recommendation based on the findings. We have calculated the estimated costs of generating full months run of insights. Assuming the company has 30 customers, it produces insights once a week for weekly insights and once a month for competitor insights. Table 15. shows the calculations for three different LLM options.

Table 15. Practical cost of LLM insight generation

Model Cost	*Cost Per Insight*	*Monthly Weekly Insight Cost*	*Monthly Competitor Insight Cost*	*Total Monthly Cost*
High (GPT-4o)	$0.0304	$3.648	$0.912	$4.56
Medium (Gemini Advanced)	$0.0186	$2.232	$0.558	$2.79
Low (Gemini)	$0.0022	$0.264	$0.066	$0.33

The total cost of the most expensive model for this run is $4.56, while the cheapest model costs $0.33. Although this cost difference may seem negligible, opting for the cheaper model results in savings of around 90%, which should not be overlooked. As companies increasingly integrate LLMs into their operations, finding ways to cut costs could prove to be crucial. For simple tasks like weekly insights, smart prompt engineering almost entirely negated the benefits of an advanced model, making the more expensive option unnecessary. However, for more complex tasks such as competitor insights, the Base models did not perform as well as the Advanced models. This suggests that as task complexity increases, the higher cost of advanced models becomes more justifiable.

To further illustrate the value of adopting AI, we compare the costs of AI-generated insights with those produced by a junior employee earning a salary of $27,000 per year. As shown in Table 16. this translates to an hourly rate of $13.80. Assuming the employee generates an insight in 15 minutes, the cost per insight is $3.45. If the process takes 45 minutes, the cost per insight increases to $10.35.

Table 16. Practical cost of employee insight generation

Human Cost	Cost Per Insight	Monthly Weekly Insight Cost	Monthly Competitor Insight Cost	Total Monthly Cost
High (45 mins)	$10.35	$1,242	$310.5	$1,552.5
Medium (30 mins)	$6.90	$828	$207	$1,035.0
Low (15 mins)	$3.45	$414	$103.5	$517.5

Even comparing the most expensive LLM option with the cheapest employee option, using an LLM to generate monthly insights costs only about 0.88% of what it would cost to employ a human. While an employee might add personal touches to insights, for simple, valuable text contributions, LLM-generated insights are the clear choice. This not only reduces costs significantly but also frees up employees to focus on tasks where they can add more value. Thus, integrating LLMs into business processes is highly recommended for efficiency and cost-effectiveness.

5.3 Limitations and Future Work

One potential limitation of our approach is the reliance on an "un-tuned" LLM, which, regardless of how well-crafted the prompt is, remains prone to hallucinations and errors. A potential way to mitigate this issue would be to create a fine-tuned version of the model, allowing for more precise guidance on output. Another benefit of this approach is the ability to provide the tuned model with additional context from previous insights, potentially leading

to more comprehensive insights Additionally, a limitation of our current work is that the model is not provided with context regarding business activities. We began addressing this by incorporating "Activity Data" to pinpoint specific instances of business activity and related social and media posts. Should the company continue to develop this feature, it would be a valuable addition to their platform. Initial testing with activity data added meaningful context to the insights.

References

Amatriain, X. (2024). Prompt Design and Engineering: Introduction and Advanced Methods. http://arxiv.org/abs/2401.14423

Appelt, D. E. (1985). Planning English Sentences. Cambridge University Press. https://doi.org/10.1017/CBO9780511624575

Arias-Pérez, J., Coronado-Medina, A., & Perdomo-Charry, G. (2022). Big data analytics capability as a mediator in the impact of open innovation on firm performance. Journal of Strategy and Management, 15(1), 1–15. https://doi.org/10.1108/JSMA-09-2020-0262

Bozóki, S., Dezső, L., Poesz, A., & Temesi, J. (2013). Analysis of pairwise comparison matrices: An empirical research. Annals of Operations Research, 211(1), 511–528. https://doi.org/10.1007/s10479-013-1328-1

Bramley, T., & Oates, T. (2010). Rank ordering and paired comparisons-the way Cambridge Assessment is using them in operational and experimental work. http://www.cambridgeassessment.org.uk/ca/collection1/digitalAssets/186333_TG_Eng_rankorder_BERA_paper_final.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. https://commoncrawl.org/the-data/

Chen, Z., & Liang, M. (2023). How do external and internal factors drive green innovation practices under the influence of big data analytics capability: Evidence from China. Journal of Cleaner Production, 404, 136862. https://doi.org/10.1016/J.JCLEPRO.2023.136862

Dale, R. (2020). Natural language generation: The commercial state of the art in 2020. Natural Language Engineering, 26(4), 481–487. https://doi.org/10.1017/S135132492000025X

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. (2023). Chain-of-Verification Reduces Hallucination in Large Language Models.

Dong, C., Li, Y., Gong, H., Chen, M., Li, J., Shen, Y., & Yang, M. (2022). A Survey of Natural Language Generation. ACM Computing Surveys, 55(8). https://doi.org/10.1145/3554727

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. In Journal of Machine Learning Research (Vol. 23). http://jmlr.org/papers/v23/21-0998.html.

Gatt, A., & Krahmer, E. (2018). Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. In Journal of Artificial Intelligence Research (Vol. 61). https://www.narrativescience.com

Goldberg, E., & Driedger, N. (1994). Using Natural-Language to Produce Weather Forecasts.

Grover, V., Chiang, R. H. L., Liang, T. P., & Zhang, D. (2018). Creating Strategic Business Value from Big Data Analytics: A Research Framework. Journal of Management Information Systems, 35(2), 388–423. https://doi.org/10.1080/07421222.2018.1451951

Gunel, B., Du, J., Conneau, A., & Stoyanov, V. (2020). Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. http://arxiv.org/abs/2011.01403

Jiwat, R., & Zhang, Z. (Leo). (2022). Adopting big data analytics (BDA) in business-to-business (B2B) organizations – Development of a model of needs. Journal of Engineering and Technology Management - JET-M, 63. https://doi.org/10.1016/j.jengtecman.2022.101676

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). Challenges and Applications of Large Language Models. http://arxiv.org/abs/2307.10169

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling Laws for Neural Language Models. http://arxiv.org/abs/2001.08361

Lample, G., & Conneau, A. (2019). Cross-lingual Language Model Pretraining. http://arxiv.org/abs/1901.07291

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. http://arxiv.org/abs/1910.13461

Lin, C.-Y. (n.d.). ROUGE: A Package for Automatic Evaluation of Summaries.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. http://arxiv.org/abs/2303.16634

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. http://arxiv.org/abs/1907.11692

Mahamood, S., & Reiter, E. (2011). Generating Affective Natural Language for Parents of Neonatal Infants.

Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R., & Reape, M. (2006). A reference architecture for natural language generation systems. Natural Language Engineering, 12(1), 1–34. https://doi.org/10.1017/S1351324906004104

Merendino, A., Dibb, S., Meadows, M., Quinn, L., Wilson, D., Simkin, L., & Canhoto, A. (2018). Big data, big decisions: The impact of big data on board level decision-making. Journal of Business Research, 93, 67–78. https://doi.org/10.1016/j.jbusres.2018.08.029

Meteer, M. W. (1995). Bridging the generation gap between text planning and linguistic realization. Computational Intelligence, 7(4), 296–304. https://doi.org/10.1111/j.1467-8640.1991.tb00402.x

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2023). GPT-4 Technical Report. http://arxiv.org/abs/2303.08774

Osuji, C. C., Ferreira, T. C., & Davis, B. (2024). A Systematic Review of Data-to-Text NLG. http://arxiv.org/abs/2402.08496

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). BLEU. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02, 311. https://doi.org/10.3115/1073083.1073135

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). Language Models are Unsupervised Multitask Learners. https://github.com/codelucas/newspaper

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. https://github.com/codelucas/newspaper

Rambow, O., & Korelsky, T. (1992). Applied Text Generation*.

Reiter, E. (1994). Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible?

Reiter, E. (1996). Building Natural Language Generation Systems.

Reiter, E., & Dale, R. (2000). Building Natural Language Generation Systems. Cambridge University Press. https://doi.org/10.1017/CBO9780511519857

Saaty, R. W. (1987). THE ANALYTIC HIERARCHY PROCESS-WHAT IT IS AND HOW IT IS USED (Vol. 9, Issue 5).

Schick, T., & Schütze, H. (2020). It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. http://arxiv.org/abs/2009.07118

Scholkmann, A. B. (2021). Resistance to (digital) change individual, systemic and learning-related perspectives. In Digital Transformation of Learning Organizations (pp. 219–236). Springer International Publishing. https://doi.org/10.1007/978-3-030-55878-9_13

Springer, S., Buta, P., & Wolf, T. C. (1991). Automatic Letter Composition for Customer Service. www.aaai.org

Theune, M., Klabbers, E., De Pijper, J. R., Krahmer, E., & Odjik, J. (2001). From data to speech: a general approach. Natural Language Engineering, 7(1), 47–86. https://doi.org/10.1017/S1351324901002625

Van Der Lee, C., Krahmer, E., & Wubben, S. (2017). PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. https://www.automatedinsights.com/

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2022). Self-Consistency Improves Chain of Thought Reasoning in Language Models. http://arxiv.org/abs/2203.11171

Webson, A., & Pavlick, E. (2021). Do Prompt-Based Models Really Understand the Meaning of their Prompts? http://arxiv.org/abs/2109.01247

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi Quoc, E. H., Le, V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models Chain-of-Thought Prompting.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. http://arxiv.org/abs/2302.11382

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing Reasoning and Acting in Language Models.

Zhong, M., Liu, Y., Yin, D., Mao, Y., Jiao, Y., Liu, P., Zhu, C., Ji, H., & Han, J. (2022). Towards a Unified Multi-Dimensional Evaluator for Text Generation.

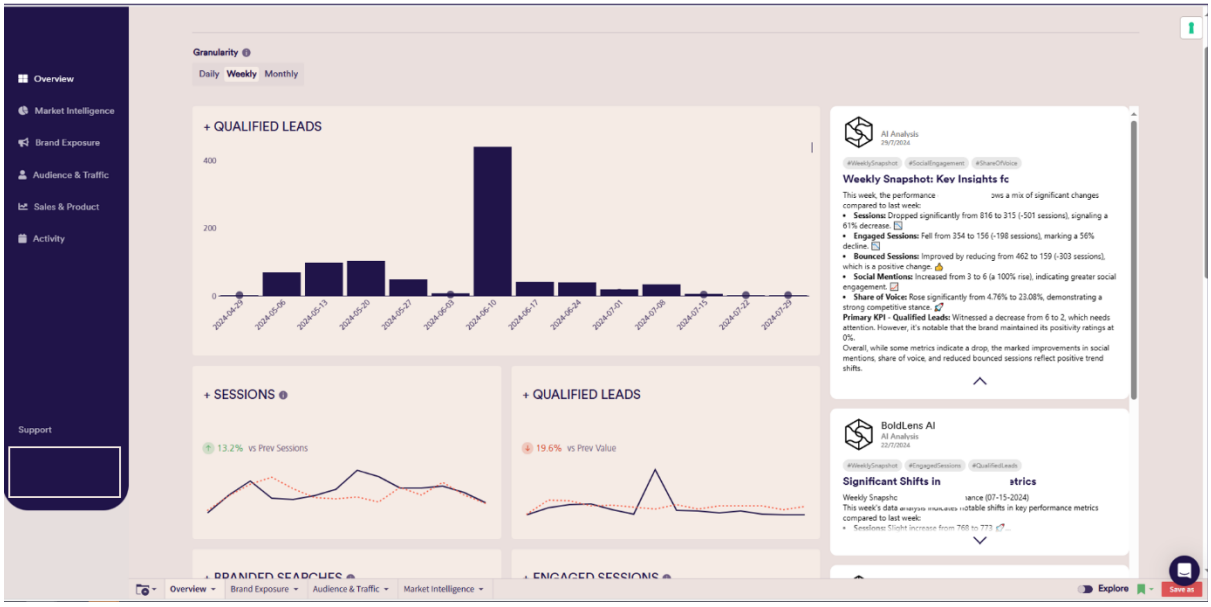
Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Zhang, Y., Gong, N. Z., & Xie, X. (2023). PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. http://arxiv.org/abs/2306.04528

Appendix

A repository containing prompts used in full as well as the code created to run the pipelines.

https://drive.google.com/drive/folders/1gzQVT66tQZBYXEMS8MXAUsKeRHKBRk_R?usp=drive_link

Screenshot where Insights have been included within the platform (Identifiable data removed)



Project Management Summary

I would consider the way that this project was managed to be a success. The collaborating company provided me with support through the form of daily stand-up meetings, where I was able to ask for any necessary help I needed. Furthermore, my company mentor was also available to help as and when I needed. As well as this I had meetings with my supervisor almost every other week, with occasional weeks missed. The goal of this project was to

create insights that would be useful, to the point that clients would use them, and from the results that we saw I would consider say the goal was achieved.

Whilst initially trying to create a system that would generate graphs for clients ad-hoc, we found that the graphs being generated were not actually adding much extra value over what was already present on the platform. It is for this reason that we pivoted to focus on the insights. Initially the insights were not of publishable standard, it would require a lot of manual intervention. But after a lengthy process of prompt engineering, we managed to get a workable system. To push this idea further the idea of competitor insights was created, after all a large idea of the platform is to provide clients with competitor information. It was tricky to get the model to give an adequate answer where it was not just listing metrics, but after considerable work we overcame this challenge. Once these systems were in working order is when I began the process of prompt engineering to try and improve these further.

We were hoping to add what we would call an Activity Insight that would include real life contextual data that would be passed to the model; however, it was not able to be completed in time. The pipeline for this step is in place for the company to continue working on it, however, due to time constraints we were unable to populate enough relevant data to get this system to work properly. In conclusion, the company tasked me with a job and in the end were happy with the solution so I would consider this process a success.