Semantic Rank, Traversal, and Analysis of Complex Graph Structures

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Daniel Verdejo

School of Computer Science

University of Galway

Supervisor(s)

Frank Glavin

In partial fulfillment of the requirements for the degree of

MSc in Computer Science (Artificial Intelligence)

August 22, 2024

DECLARATION I, Daniel Verdejo, hereby declare that this thesis, titled "Semantic Rank, Traversal, and Analysis of Complex Graph Structures", and the work presented in it are entirely my own except where explicitly stated otherwise in the text, and that this work has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: _____

Abstract

This thesis introduces a graph-based semantic rank, traversal, and analysis system that navigates a graph structure using an embedding model to calculate and prioritise connected vertices based on contextual relevance. The system aims to generalise well over varying types of relational data represented as a graph to enable a mechanism to rank, traverse, and analyse data. From an initial vertex, the algorithm encodes the given context and computes semantic similarities between the context and the properties of each vertex. These similarities guide the traversal, dynamically updating the relevance scores to refine the exploration path. The traversal continues exhaustively until all relevant vertices are visited, ensuring the discovery of the most contextually pertinent path and highest-scoring vertices. This approach demonstrates an efficient method for context-driven graph exploration, optimizing for meaningful data retrieval and analysis. It also uses a Large Language Model to analyse the results of the rankings and traversal, offering additional information based on the user-defined context.

Keywords: Graph, Semantic, search algorithm, scoring, similarity

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ability to efficiently search, visualise, and analyse complex data is essential as the amount of data available grows daily [2]. Graph Theory (GT) defines a framework for representing entities and relationships as graph structures. Fundamental elements of social networks, knowledge graphs, and recommendation systems are based on graph models, which intuitively illustrate complex information [3] [2]. In the context of search and traversal, graph visualisations and algorithms help analysts discover relationships and patterns that may be difficult to identify in traditional data structures. As data becomes more interconnected, sophisticated methods are required to derive meaningful insights. As such, graph-based approaches have improved the management and analysis of large datasets and enhanced the precision and relevance of Information Retrieval (IR) [3]. In a graph, entities are represented as vertices and relationships as edges. This is an intuitive way to model real-world scenarios, allowing for integrating various data types, and facilitating analysis and interpretation [3]. Introducing semantic similarity, a context-aware ranking mechanism, into a graph-based search system can enhance the relevance of results [4]. This is particularly

useful in understanding the nuances of data and the retrieval of information, such as:

- Understanding user interactions, influence, and community detection within social networks.
- Structuring and querying vast amounts of interconnected information in a well-organised and accessible manner.

The aim of combining concepts from Natural Language Processing (NLP) and GT is to address a gap in the existing research in the field of GT by combining the strengths of both to enhance relevance, understanding, and analysis. Traditional graph traversal methods efficiently explore data but cannot prioritise contextually relevant paths. Additionally, Large Language Models (LLM(s)) can provide insights by comprehension and explanations of complex data which may otherwise have been difficult to identify if not missed entirely. This combination could aid in turning raw data into actionable knowledge, which as a result has the potential to improve decision-making processes and uncover hidden patterns within complex datasets. The research carried out builds upon and extends existing methods, offering an alternative approach to data analysis using cutting-edge tools and technologies.

1.1 Research Questions and Objectives

The research sought to answer the following questions:

1. RQ1: Can meaningful information be understood from embedding representations of complex graph data structures?

To achieve this, an evaluation of the performance of six embedding techniques and technologies and systematically evaluate each of their performance across various similarity queries, including graphs, context (concepts and ideas present in the structure), term (specific terms which exist within the structure), and entities (things such as a person, organisation, or object).

2. RQ2: Can LLMs carry out detailed analysis of complex graph data structures?

   To achieve this, a system will be developed to provide a graph, and prompt to LLMs to analyse said graph based on various criteria. A variety of LLMs should be evaluated to understand each of their capabilities in graph analysis tasks. The results from each should be scored based on some criteria, with the perceived most capable model(s) for this task being observed.

3. RQ3: Can significant or relevant paths be identified between a starting vertex and contextually relevant vertex?

   To achieve this, an algorithm which uses the most capable embedding technique discovered in RQ1 to be used to guide graph traversal will be defined. Following this, the algorithm will be compared against existing algorithms used for graph-based traversal, and ranking.

4. RQ4: Does the LLMs analysis of the graph improve as result of supplementing semantic graph traversal data?

   To achieve this, the same system used in RQ2 will be used with minor changes and additional data supplemented. The results from RQ2 will be compared against the new set of results generated from the

updates for evaluation and analysis. The comparison will assess if an improvement in the LLMs analysis of the graph structure is observed.

1.2 Thesis Structure

In the following chapters, we will explore the key concepts and findings of the research. Chapter Two, Background, serves as an introduction to the various aspects of the research, providing essential academic knowledge on each topic. Chapter Three, Related Work, examines the advancements in the concepts relevant to this thesis.

Methodology (Chapter Four) offers insight into the dataset and data handling techniques, along with a detailed description of the system developed through this research. It concludes by outlining the system's approach.

Chapter Five, Experiments, presents a series of experiments, with their results discussed in the subsequent chapter, Results (Chapter Six). In this chapter, the results are analysed and linked back to the research questions to provide insights or answers.

The thesis concludes with Chapter Seven, Conclusion, which provides high-level commentary on the findings, discusses future work, and highlights the limitations of the research.

The rationale behind the decisions made in this research is rooted in the need for a comprehensive analysis of graph-based information, whether that be for social-network analysis or IR. The integration of LLM analysis, NLP techniques, and GT methods allows for a nuanced evaluation of the strengths and limitations of different approaches. These choices are justified by their relevance to the research objectives, aiming to enhance understanding when applying the combination of GT with NLP and LLMs to

real-world graph analysis scenarios.

Chapter 2

Background

Graph theories origins can be traced back to Leonhard Euler's solving of the "Seven Bridges of Königsberg" problem. The problem explored the feasibility of traversing four regions of a city, linked by seven bridges, without retracing any bridge. Euler found that it was not possible to find such a path, emphasizing the significance of connections and laying the groundwork for the principles of GT [3]. The evolution of computer science has been shaped by mathematicians, and as computers became more integral to society, their needs began to guide research in GT [5]. Beyond its mathematical roots, graphs serve as a practical and precise method for modelling and analysing data. In GT, vertices represent entities, and edges represent their relationships between one another. Graph analytics uses graph-based methods to examine interconnected data through querying graph data, applying basic statistics, visually exploring graphs, or integrating graphs into machine learning (ML) tasks [3] [5]. Graph algorithms are crucial in understanding connected data across a wide range of real-world systems, from protein interactions to social networks and communication systems. These algorithms enable the discovery of patterns within highly

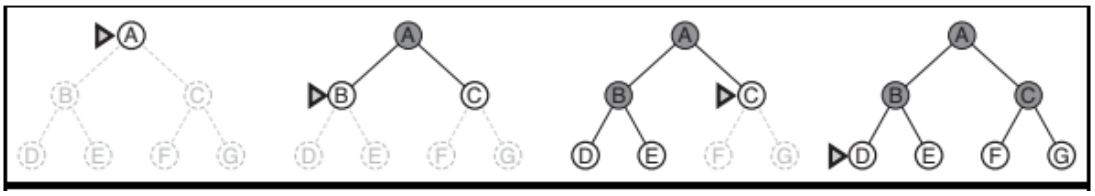


Figure 2.1: BFS illustration. Reference: Artificial Intelligence: A modern approach (3rd edition), Figure 3.12, page 82 [1]

connected datasets, offering significant potential for insight and innovation [3]. Traversal is a fundamental part to many graph algorithms, it allows for the evaluation of the graph's structure and the solving of computational problems. Traditional graph traversal algorithms, like Depth-First Search (DFS) and Breadth-First Search (BFS), efficiently query and traverse large datasets exhaustively. Other search algorithms such as Dijkstra's shortest path (DSP) and A-star (A*) offer more specialised approaches to finding the shortest or most optimal paths between two vertices by incorporating heuristic evaluations to enhance efficiency and precision in navigation and path-finding tasks.

- BFS, shown in Figure 2.1, is a simple approach where the root (starting vertex) is explored first, followed by all of its children (the vertices at the next level to it), and then all of their children. In essence, all vertices at a specific depth in the search tree or graph are fully expanded before moving on to vertices at the next depth level [1].

- DFS, shown in Figure 2.2, starting from a vertex, referred to as the root, and explores as far along each branch as possible before backtracking. This approach ensures that into a path before moving on to explore other paths [1].

- DSP finds the shortest path from a single source in weighted graphs

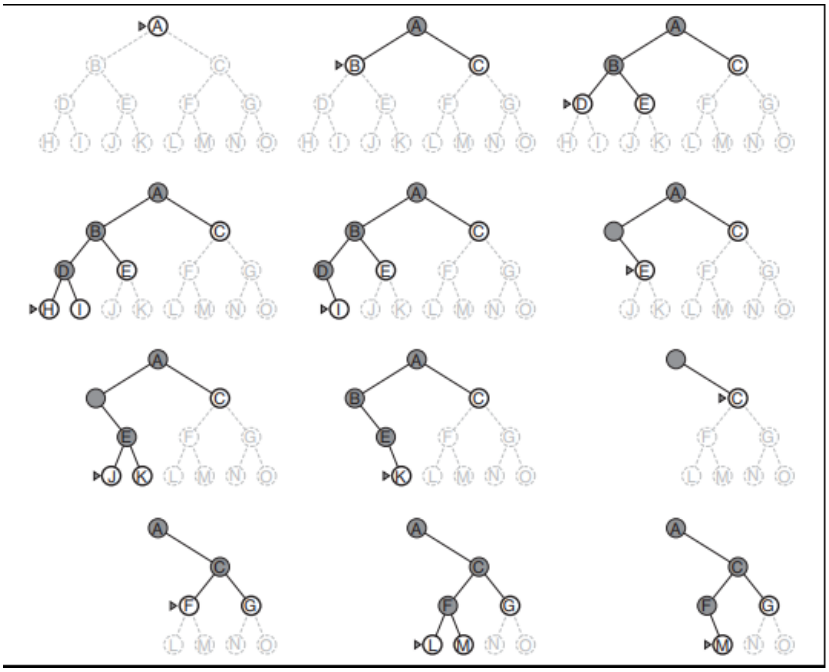


Figure 2.2: DFS illustration. Reference: Artificial Intelligence: A modern approach (3rd edition), Figure 3.16, page 86 [1]

with non-negative weights by iteratively selecting the smallest known distance vertex, using a priority queue updating the distances to its neighbouring vertices, and marking the node as “visited”, continuing this process until the shortest paths to all vertices are calculated.[6].

- A* extends DSP and Greedy Best-First Search by using a heuristic cost function to guide the search towards the goal efficiently. It ensures that an optimal and complete path is found consistently, provided the heuristic never overestimates the true cost [7].

Another algorithm for ranking vertices in a graph, relevant to this research, is PageRank (PR) [8]. It’s an iterative method for ranking web pages based on their importance, operating on a graph of vertices (representing web pages) and edges (representing links between pages) [8]. In each it-

eration, a vertex's score is calculated by summing the PR scores from all vertices linking to it, divided by the number of outlinks from that vertex [8]. Traditionally, all vertices start with equal scores, but an alternative approach initialises scores based on the ratio of in links received. Optional weightings can be applied to adjust the algorithm, although this functionality remains unused. When integrating PR into a search engine, ranking parameters can be customized to bias results according to user preferences [9]. Resource Description Framework (RDF) is a graph structure used to represent information about web resources. It consists of triples, each comprising a subject, predicate, and object, forming a directed, labelled graph [10]. Previous research by Hogan, Harth and Decker sought to propose and evaluate context-aware ranking functions for RDF data, to enhance the ranking quality of RDF data collected from the Web by considering both the data graph and the data provenance graph [9]. In a similar vein, we use context as a mechanism to rank and guide traversal in this research.

Semantic similarity in NLP is critical for assessing how closely related two pieces of text are in terms of their meaning, providing deep insights into content comprehension [11] [12]. Techniques include Vector Space Models (VSM(s)) which represent text as vectors [13], and calculating cosine similarity [14], contextual embeddings using models such as BERT [15] or generative pre-trained transformers (GPT) models like text-embedding-ada-002. Semantic similarity enhances the performance of tasks such as IR and text classification by enabling more effective interpretation of language [16]. It helps prioritise relevant documents in IR tasks and improves context understanding in machine translation. By assessing meaning beyond exact wording, systems can provide more accurate and context-aware results.

LLMs are a type of generative artificial intelligence (Gen AI) based on

transformer models, which will be discussed later. LLMs are trained on large amounts of data so that they are capable of understanding the complex nuances of language, such as the meaning of words and how each word relates to each other in a given sequence. This allows them to be able to carry out a variety of tasks like text generation, summarisation, translation, etc. With this understanding, they can generate output sequences that are on par with humans. For example, models such Llama3 has 70 billion parameters and was pretrained over 15 trillion tokens of publicly available data, which included multilingual data of 30 different non-English languages as well as code [17]. In November 2022, OpenAI announced the public availability of ChatGPT based on GPT-3.5-turbo. The raw capabilities of this model attracted enormous attention, with Reuters reporting that OpenAI had one-hundred-million monthly active users in January 2023, representing the fastest-growing user base to date [18]. There were also sensationalised narratives suggesting that this kind of Generative AI would cause significant job losses. However, actual analysis on the topic revealed that while there is potential for job displacement, the level of such effects had been exaggerated. Instead, these technologies are expected to have a transformative effect on productivity in the workplace [19] [20]. This research did not explore the economic impact of these systems, though it would be an interesting read once more statistics are published on the matter.

Chapter 3

Related Work

3.1 Natural Language Processing

Semantic similarity offers a measure of similarity between two pieces of text or objects, which can represent entities, concepts, paragraphs in a book, etc. [21]. This concept is critical in IR, as it gives systems the ability to identify relevant results to that of the user query [22]. Semantic similarity offers more than just simple surface-level syntactic similarity (word matching), giving deeper insight into the relationship between words. This involves understanding the meaning, context and relationships between concepts, and entities.

Various approaches have been designed to measure semantic similarity, including VSMs, word embeddings like Word2Vec [11] or GloVe [12], knowledge graphs, and latent semantic analysis (LSA). Language models such as BERT, RoBERTa, and their variants demonstrate exceptional performance in various NLP tasks such as sentiment analysis, and text classification, for example [23]. Breakthroughs in language understanding have shown that unsupervised pre-training with rich language models is a criti-

cal component of many AI systems. These advancements have enabled even resource-constrained tasks to leverage the power of deep unidirectional architectures, enabling models to tackle a diverse range of NLP challenges with success [15]. Embedding models like OpenAI's text-embedding-ada-002 offer a deep understanding of text for the generation of text embeddings, enabling excellent natural language understanding. These models can capture subtle linguistic nuances, such as polysemy and figurative language. This has led to significant advances in measuring semantic similarity, as these embedding models can be fine-tuned for specific tasks or domains, allowing them to adapt to the unique characteristics of various languages [23]. Previous research has shown that language models and LLMs like GPT-3 have strong performance in many NLP tasks in zero, one, and few-show scenarios, sometimes matching the performance of fine-tuned systems as well as "strong qualitative performance at tasks defined on-the-fly" [24], thus making them a suitable choice for generating embeddings and potentially a capable tool to carry out analysis on complex data structures.

The paper titled "Attention is All You Need" [25] introduced the transformer model, showed several advantages over its counterparts, including faster training times, better handling of long-range dependencies, and superior performance across various NLP tasks [25]. The Transformer architecture removed the need for recurrence in recurrent neural networks (RNN) based models by using positional encodings so that models could efficiently process sequential data, making them faster and self-attention mechanisms to process input sequences in parallel [25]. This approach enabled models to learn long-range dependencies between words of an input sequences, which is a key challenge in many sequence transduction

tasks [25]. As a result of these innovations, the transformer demonstrated superior performance on various NLP benchmarks (BLEU, Training Cost (FLOPs)), including machine translation tasks (WMT 2014 English to German, WMT 2014 English to French), and set a new standard for NLP model performance [25]. Transformers were originally developed for machine translation tasks (text-to-text) but have since been found to be applicable in a variety of other tasks like text-to-speech, text-to-image, etc.

In light of this, the choice to use embedding models for ranking vertices is based on their ability to capture nuanced semantic relationships and contextual dependencies within large-scale datasets. These powerful models can harness their advanced language understanding capabilities to generate more accurate and informative rankings that reflect the nuances and complexities of language and entities. LLMs could be used for analysing complex datasets due to their ability to capture subtle patterns and relationships found within the data. This could be particularly valuable when working with large-scale graph structures to assist in uncovering meaningful insights, or when seeking to identify patterns and relationships between vertices.

3.2 Graph Theory

The analysis of paths between vertices in graphs is a fundamental aspect of GT with numerous applications across various domains [3]. This includes shortest path queries, which can be used directly or aggregated to identify small-world properties and communication efficiency. Watts and Strogatz proposed that many real-world networks have small-world properties, this means identifying if a network has the same characteristics, such as effi-

cient connections between distant vertices, or local clusters of connected vertices [26]. Similarities between the shortest path calculations and short paths in small-world networks (SWNs) reveal a connection between these two concepts. On one hand, the shortest path algorithms like DSP find the most efficient route from a source vertex to a target vertex by exploiting the underlying network topology [27] [28]. Optimal routing mechanisms share similarities to how shortcuts in SWNs enable efficient traversal of the network, allowing for swift and direct passage between vertices. Efficient connectivity and optimal routes are achieved through the strategic use of intermediate vertices or edges that bypass less relevant or more distant connections. Furthermore, the shortest path calculation and SWN structure are both influenced by the underlying network topology, highlighting a common fundamental mechanism driving these events [26] [28]. These similarities suggest that the properties of short paths in SWNs, such as high clustering coefficient and short average path length, may be related to or even predictive of the outcomes of shortest path calculations in other networks [26].

Social network analysis has long been used to study human communication patterns [29]. Social interactions can be modelled as graphs, where vertices represent individuals and edges represent connections between them. Researchers have analysed various social media platforms, such as X (formerly known as Twitter), using techniques such as clustering and ranking users based on authority scores. For example, studies have identified influential users on X by applying PR or Hyperlink-Induced Topic Search algorithms to user-tweet graphs, considering factors like follow relationships, post relationships, and retweets [29]. These analyses have helped researchers understand shared interests, community structures, and key

users who drive the conversation on social media platforms. In social network analysis, two types of graphs can be made: real graphs and virtual graphs where real graphs represent direct data connections, and virtual graphs are derived from real graphs to analyse patterns or processes [30].

One of the most significant contributions to the field of IR and search engines was made in the paper "The Anatomy of a Large-Scale Hypertextual Web Search Engine" by Brin and Page [8] established the foundation for many search engine algorithms and techniques still seen in use today. The authors described their experience of searching with search engines with and without the use of PR, highlighting the importance of this algorithm in ranking web pages based on relevance and popularity [8]. Previous research has built upon the foundation of PR, with notable extensions such as Topic-Sensitive PageRank (TSPR), Context-aware Personalized PageRank (CPPR), and more. The TSPR algorithm generated topic-specific graphs to improve relevance for queries by combining relevant vertices from multiple versions of the graph. Experimental results showed that TSPR outperformed traditional PR and other methods in terms of relevance and user satisfaction [31]. Beyond web search, PR has been applied to various domains like social networks, recommendation systems, etc. but its adaptation to non-web-based networks is hindered by challenges like data sparsity, scalability, and domain knowledge [31]. Notably, recent research has proposed similar concepts, such as CPPR, which generate personalized recommendations by building graph models representing user preferences and context features [31].

3.3 Combining Natural Language Processing and Graph Theory

To overcome the limitations of traditional pathing algorithms in capturing complex semantic relationships between vertices, previous research has explored various semantic graph traversal techniques incorporating machine learning models to capture nuanced vertex similarity and connectivity patterns [32]. The integration of NLP and GT concepts has been studied in numerous papers which align with the research and experiments in this paper. For example, Node2vec [33] defined a semi-supervised algorithm for scalable feature learning in networks by optimising a graph-based objective function using Stochastic Gradient Descent (SGD), inspired by NLP techniques [33]. Node2vec's effectiveness was tested on various real-world networks, showing it outperformed algorithms such as DeepWalk and LINE in multi-label classification and link prediction tasks [33]. Similarly, an embedding model is used in this research to encode vertices' properties and compute their semantic similarities against a given context. Other research outlined an unsupervised method named RaDE (Rank Diffusion Embedding) and its variant RaDE+ to generate low-dimensional vector representations based on similarities between common vertices and highly effective representative vertices in a network, with RaDE+ capable of considering multiple representative vertices for each class [4]. RaDE and RaDE+ achieved favourable results in most cases, generating highly effective representations that were denser and smaller while significantly improving overall effectiveness according to the research [4]. An example provided found that in the ALOI dataset, the output was 35.15 times smaller than the original with a slight improvement in effectiveness (MAP relative gain) [4].

Another notable combination of semantic similarity with graphs is detailed in the paper “Knowledge Graph-Based Semantic Ranking for Efficient Semantic Query,” which proposes a knowledge graph-based semantic ranking method to mine latent semantic knowledge behind queries. This method enriches syntactical queries semantically using the keywords’ latent semantic knowledge. Experimental results using Sogou search engine query logs and user click and browsing data show that this method effectively improves ranking performance [34]. Additional research on knowledge graphs used LLMs to propose an approach for multi-hop link prediction tasks [35]. The system used LLMs processed by the “Knowledge Graph Large Language Model” (KG-LLM) framework, which resulted in a significant improvement in prediction accuracy according to the research [35].

The system presented as part of this research aligns most with elements of these papers. It uses an embedding model to encode vertices’ properties and compute their semantic similarities against a given context, similarly to Node2vec. It also uses LLMs for their capabilities in understanding complex graph-based structured data, which somewhat aligns with experimentation carried out for the KG-LLM framework. The combination of these elements allows the system to prioritise vertices with higher semantic scores during traversal, and effectively analyse complex relationships between vertices in real-world graph structures and applications, ultimately informing more effective solution for social network analysis, in this case.

Chapter 4

Methodology

4.1 Data and the Ontology

Data Selection and Suitability

When selecting data for graph-based social network analysis, it's important to choose datasets with natural text elements and highly connected relational data to form dense, diverse networks. This enables the uncovering of patterns, trends, and insights not easily found through traditional methods.

The IMDB ontology can analyse social networks within the film industry by examining connections between actors, directors, and movies, revealing insights such as how people are connected, and preferences for genres, for example. By applying the semantic rank and traversal algorithm to this data, observations such as relevant connections and associations can be identified.

For LLM analysis tasks, JSON-LD will be used due to its flexible and interoperable format, suitable for representing complex relationships in datasets [36]. The ontology involves entities like Movies, Directors, and

Actors, capturing attributes and intricate relationships necessary for analysis.

Data Acquisition

The dataset utilised in this research was sourced from Kaggle, specifically the IMDB-top-1000.csv dataset[37]. It's essential to note that the primary objective of this research was not solely centred around constructing a system tailored for movie data analysis. Instead, the goal was to develop a methodology with the versatility to be applied to various types of structured graph data. The dataset used comprises rows detailing the top 1000 movies on IMDB from 1930 through to 2019. The transformation process involved converting the rows from the CSV file into separate entities in JSON files, these were used to generate JSON-LD graphs, effectively creating an ontology.

Data Transformation

A comprehensive data transformation procedure was used to create an ontology based on IMDb movie data. Initially, the script reads the IMDb movie data from a CSV file (`imdb_top_1000.csv`) using the Pandas library, organising it into a DataFrame. Subsequently, each row of the DataFrame is iterated, and the movie-related details are transformed into a structured ontology format. This includes extracting attributes like title, release year, certificate, runtime, genre, ratings, overview, and cast members, which are then organised into a JSON format representing the "Movie" entity. Additionally, directors and actors are identified and stored separately, establishing relationships between them and the movies.

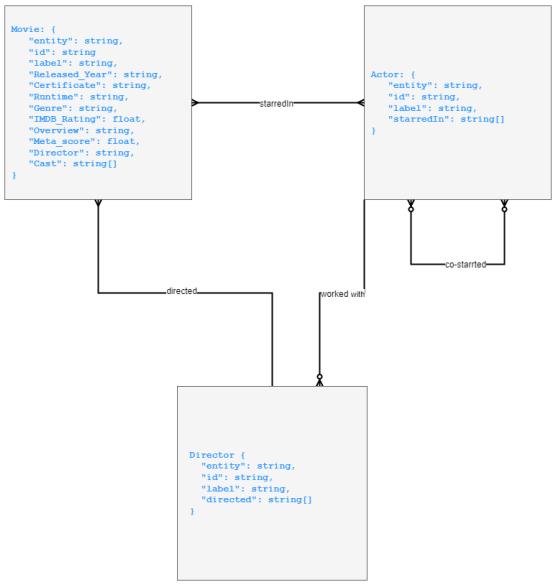


Figure 4.1: A simple ontology for representing IMDB data.

Ontology Structure and Output Files

The resulting ontology, shown in Figure 4.1, is structured into three primary components: movies, directors, and actors. Movies encapsulate detailed information and are linked to their respective directors and cast members. The script establishes relationships by creating edges in the ontology, representing connections such as "directed," "starred in," and "worked with." These edges form a network of relationships within the ontology. Each entity and relationship is assigned a unique identifier using UUIDs, ensuring uniqueness and facilitating easy referencing.

4.2 System architecture

This section examines the structural design of the system, including the organisation and interaction of data-ingestion, embedding models, traver-

sal algorithms, LLMs, the analysis, and the Representational State Transfer (REST) Application programming interface (API) which this all can be accessed via.

By detailing the system architecture, the aim is to provide a thorough understanding of how each component contributes to the overall functionality and performance of the system. Through this detailed exploration, the intention is to give a rich description of the framework that the system is built from, ensuring a robust, scalable, and efficient solution.

4.2.1 The embedding model provider

The embedding model provider system is designed to provide an interface for various embedding models. It employs a factory method approach, where a client can request an instance of a specific model using the *embedding_model_provider* function.

Each model is implemented as a property of the *EmbeddingModel* class, which provides a common interface for encoding and calculating similarities between embeddings. An abstract base class *Encoder* is defined, which is the interface for different encoder implementations. Subclasses are expected to implement the encode method, transforming input values into embeddings. The *EmbeddingModel* wraps an encoder and provides additional functionalities. A prepossessing method is defined to clean and tokenize the input text, removing stop-words (nltk english), HTML tags, and URLs. The encode method pre-processes the input using the *__preprocess_text* function and delegates the encoding task to the injected encoder. A similarity function calculates the cosine similarity between two embedding vectors using the cosine similarity function from scikit-learn.

The encoder models which can be wrapped are as follows:

1. **Mistral**: Inherits from *Encoder*: Uses the MistralAI client to encode input text into embeddings. This uses *Mistral-embed* by default, if an alternate model is available the user can specify the model during class instantiation. The *encode* method sends the input to MistralAI and retrieves the embeddings from the API response.

2. **Bert**: Inherits from *Encoder*: C the BERT model from Hugging Face's Transformers library to encode input text into embeddings. By default, *Bert-base-uncased* is used, a different model can be specified when instantiating the class.

3. **OpenAI**: Inherits from *Encoder*: Uses OpenAI's API to generate embeddings for input text using a specified model variant. Users can specify between the different embedding models provided by OpenAI, for example *text-embedding-ada-002* is used as the default model choice but users can define a new or better model such as: *text-embedding-3-small* when instantiating the class. The *encode* function sends the input value to the OpenAI API and extracts the embeddings from the API response.

4. **SentenceTransformer**: Inherits from *Encoder*: Uses the *Sentence-Transformer* model from the sentence-transformers library to encode input text into embeddings. By default, *all-MiniLM-L6-v2* is used but new or better models could be used when instantiating the class. The encode method takes a text value, passes it through the Sentence-Transformer model, and returns the resulting embeddings.

This design was intended for modularity, enabling easy dependency injection of specific encoders within the EmbeddingModel to experiment with different pre-trained models for generating embeddings, or based on their

capability for specific tasks. This design allows for hot-swapping embedding models without modifying the client code, making it a flexible and scalable solution. It also defines a blueprint for future embedding models to be added as more become available. It should be noted that this did not capture all available models, just a selection of some of the most capable available at the time of the evaluation.

4.2.2 The LLM Provider

Similarly to the embedding model provider, the LLM provider system is designed to offer an interface for various LLMs. It uses a factory method approach, where a client can request an instance of a specific model using the *llm_provider* function.

Each model is implemented as a property of the *LLM* class, which provides a common interface for generating text, answering queries, and performing other NLP tasks. An abstract base class *ChatModel* is defined, which serves as the interface for different language model implementations. Subclasses are expected to implement the *chat* method, transforming input messages into responses. The *LLM* wraps a language model and provides additional functionalities. A method is defined to manage the conversation context, including setting and retrieving the system message, ensuring the assistant maintains a coherent and contextually aware dialogue.

The following encoder models can be wrapped:

1. **OpenAILLM**: Inherits from *ChatModel*. Uses the OpenAI API to generate responses for input messages using a specified model variant. The *chat* method sends the input messages to the OpenAI API and retrieves the response from the API. Additionally, the *fn_chat* method

enhances the model's capabilities for specific tasks through the use of various tools. This method allows the model to leverage these tools to perform functions that are beyondit'snative capabilities, thereby extendingit'sutility. For example, it can be used to call external APIs, integrate with databases, or use specialised libraries for advanced data processing. It should be noted that in order to avoid scope creep this was not thoroughly evaluated as part of this research but would be a nice avenue to explore in the future.

2. **OllamaLLM**: Inherits from *ChatModel*. Uses the Ollama API to generate responses for input messages using a specified model variant. The *chat* method sends the input messages to the Ollama API and retrieves the response from the API. The choice to use Ollama allows for a wide variety of models which are available in the open source space including Mistral, Gemma, Llama, Phi, to name a few.

This design was intended for modularity, enabling easy dependency injection of specific LLMs within the *LLM* class to experiment with different models for generating responses, or based on their capability for specific tasks. This design allows for easily changing which LLM the user uses without modifying the client code, making it a flexible solution. This system does not capture all available models, just a selection of some of the most capable available at the time of the evaluation. Notably, the system currently omits models such as Google's Gemini or Anthropic's Claude for no reason in particular but given that they are comparable models to the likes of GPT-4o and Llama3 they are good candidates to be integrated in the future and evaluated against their peers.

4.2.3 The Traversal Algorithm

The semantic_traversal function enables a semantic traversal of a graph using a specified embedding model and context. This process allows for more meaningful navigation through the graph by considering the semantic similarity between vertices.

Parameters and Return Signature

Parameters

- `graph (Graph)`: The graph object to traverse.
- `embedding_model (EmbeddingModel)`: The model employed for encoding vertex properties and calculating semantic similarity scores.
- `start_vertex_id (str)`: The ID of the starting vertex for the traversal.
- `context (str)`: The context used to generate the initial embedding, guiding the traversal.
- `ignore_direction (bool, optional)`: If set to `True`, ignores the direction of edges in the graph. Defaults to `False`.

Return Signature

- `tuple`: A tuple containing:
  - `order`: The order in which vertices are visited.
  - `scores`: The semantic scores of the visited vertices.
  - `best_path`: The path to the vertex with the highest semantic score.

- `best_score`: The highest semantic score achieved.
- `paths`: All paths explored during the traversal.

Understanding the Parameters it's important to understand the reasoning behind some of these parameters, most notably `ignore_direction`, and `min_score_threshold`.

- **`ignore_direction`:** Treating a graph as undirected (i.e., ignoring edge directions) can simplify various tasks and reveal new insights in network analysis. This approach helps identify strongly connected components, detect cycles more easily, and find alternative paths or shortest paths between vertices. Viewing a graph undirected can help in assessing its robustness and connectivity, visualising its structure, and understanding topological sorting feasibility. By disregarding edge directions, an undirected perspective can provide a complementary view to directed graphs, offering new perspectives on network properties and relationships. In GT there is a concept of a "hub" or central vertex. In a network or graph, a hub is a vertex that has a high degree of connectivity, meaning it's connected to many other vertices. These hubs can act as gateways to various parts of the network, facilitating traversal or exchange between different components. In many real-world networks, such as social networks, transportation systems, or biological networks, hubs often emerge as a natural consequence of vertex interactions and relationships. In the ontology used in this research, for example, a movie can act as a hub between actors and directors, or a director can act as a hub between many movies and actors, and so on. Identifying these hubs can provide valuable insights into graph structure and dynamics.

- **`min_score_threshold`:** The minimum score threshold is designed to disregard vertices with very low cosine similarity to the context. This mechanism serves two purposes: it prevents the algorithm from pursuing irrelevant paths and stops traversal in certain directions when the similarity score reduces as the algorithm moves further from the source vertex. As in this current form of the algorithm there is no depreciation factor as the algorithm gets further away from the source vertex this is not possible as an exit

Algorithm Breakdown

1. **Encoding Context**: Using the provided embedding model, the context string is encoded into a vector representation:
2. **Initialisation**:
    - A min-heap (`min_heap`) is initialised to manage traversal priorities, starting with the initial vertex and a zero score.
    - Other data structures are initialised to track visited vertices, traversal order, scores, and paths.
3. **Direction Handling**: If `ignore_direction` is set to `True`, an additional structure (`incoming_edges`) is created to account for undirected traversal by mapping vertices to their incoming edges.
4. **Traversal Loop**:
    - The main loop processes vertices in order of their priority from the min-heap:
        - The current vertex is marked as visited and added to the traversal order.

- The path and score for the current vertex are stored.
- Neighbouring vertices are evaluated for traversal based on their semantic similarity to the context:
  * Each neighbour's properties are encoded, and a similarity score is calculated.
  * If the similarity score exceeds the threshold, the neighbour is considered for the next traversal step with a depreciated mean score.
- If `ignore_direction` is enabled, incoming edges are similarly processed to ensure undirected traversal.

5. **Result Compilation**:
   - After completing the traversal, the vertex with the highest score, along with its path and score, is identified and returned alongside the overall traversal order, scores, and paths.

4.2.4 The Analysis Provider

The `Analyser` class brings all the previously mentioned components together (Embedding model, traversal algorithm, and LLM) as a simple mechanism to carry out the final task of ranking, traversing, and analysing graphs. It's responsible for:

- Setting up and managing the configuration of traversal parameters.
- Configuring and managing prompt templates.
- Performing graph traversal and analysis.
- Keeping track of the traversal and analysis history.

- Interfacing with language models to provide context-based analysis.

When initialising the `Analyser` class, it initialises the language model, establishes parameters for embedding models, and minimum score thresholds, it also creates data structures to track traversal history, system prompts and default prompt templates are configured using predefined functions.

4.2.4.1 Constructor

Parameters:

- `embedding_model_name` (`EmbModelName`): The name of the embedding model to use.
- `llm_name` (str): The name of the language model to use.
- `traversal_min_score_threshold` (float, optional): Minimum score threshold for traversal (default is 0.01).
- `embedding_model_variant` (str, optional): Variant of the embedding model (default is `None`).

4.2.4.2 Methods

There are utility functions defined for configuring and retrieving information about how the analyser is configured. They include setting parameters such as minimum score thresholds, system prompts, default prompt templates, embedding models, and language models. These functions also include get functions for accessing the traversal history and the current configuration of the analyser. The configuration includes model names, traversal parameters, and prompt templates. While these functions are not critical to understanding the overall functionality, they can be useful for

customising or troubleshooting the analyser. Below is a list of the critical functions which are responsible for graph ranking, traversal and analysis.

- `traverse(graph, start_vertex_id, context, ignore_direction)`
  - Performs semantic rank and traversal on the given graph starting from a specific vertex and using the given context.
  - Uses a combination of traversal parameters and embedding models to find paths and scores within the graph.
  - Stores the results in the traversal history.
- `analyse(graph_data, start_vertex_id, context, ignore_direction)`
  - Calls the `traverse` function to get the traversal results.
  - Performs analysis on the graph and traversal data starting from a specific vertex with a given context.
  - Injects values into the prompt template (using the default prompt template, if not changed) and sends it to the configured LLM for analysis.
  - Stores the analysis results with traversal data in the traversal history.

4.2.5 The Representational State Transfer Application Programming Interface

The REST API is built to provide an interface for interacting with the analyser. The API consists of several endpoints that allow users to get and set configuration parameters, get the traversal history, and perform analysis

and traversal on graph data. The server is set up to run on `0.0.0.0` at port `8000` using `uvicorn`. The main components include:

- Configuration Parameters Management
- Traversal History Management
- Graph Analysis and Traversal

4.2.5.1 Configuration Model Request Parameters

- **min_score_threshold**: A float representing the minimum score threshold.
- **default_prompt**: A string for the default prompt used in analysis.
- **embedding_model**: An enumeration indicating the embedding model to use.
- **embedding_model_variant**: An optional string for specifying the variant of the embedding model.
- **llm_name**: An optional string for specifying the large language model name.

4.2.5.2 Traversal Request Parameters

- **graph_data**: An object representing the graph data.
- **start_vertex_id**: A string representing the starting vertex ID in the graph.
- **context**: A string providing context for the analysis.

- **ignore_direction**: A boolean indicating whether to ignore the direction of edges in the graph.

4.2.5.3 Endpoints

To avoid describing the entire API only the most

Get/Set Configuration Parameters The endpoints for getting or setting configuration parameters is located at `/api/v1/get_current_config/` via a GET method. `/api/v1/set_config_params/` via a POST method, respectfully. These endpoints facilitate viewing or configuring some of the core aspects of the system. The request / response body of these endpoints look like so:

Listing 4.1: Request Body Example

```
{
"min_score_threshold": 0.01,
"default_prompt": "Default prompt text",
"embedding_model": "model_name",
"embedding_model_variant": "variant",
"llm_name": "llm_name"
}
```

Get Traversal History The endpoint for retrieving the traversal history is located at `/api/v1/get_traversal_history/` via a GET method. This endpoint returns a list of traversal history entries, providing users with access to a record of previous analyses and traversals performed by the analyser.

Listing 4.2: Response Example

```
{
"traversal_history": [
    {"timestamp": "2023-07-29T12:00:00Z", "details": "Traversal
        details..."}
]
}
```

Analyse Graph Data The endpoint for analysing graph data is located at `/api/v1/analyse/`, and can be accessed via a POST method. The request body must contain *Traversal_Request_Params*. This endpoint enables users to analyse the provided graph data starting from a specified vertex with a given context, returning analysis results in response.

Listing 4.3: Request Body Example

```
{
"graph_data": {"vertices": [...], "edges": [...]},
"start_vertex_id": "vertex_id",
"context": "context information",
"ignore_direction": true
}
```

Traverse Graph Data The endpoint for traversing graph data is located at `/api/v1/traverse/`, and can be accessed via a POST method. The request body must contain *Traversal_Request_Params*. This endpoint allows users to traverse the provided graph data starting from a specified vertex with a given context, returning traversal results in response.

Listing 4.4: Request Body Example

```
{
"graph_data": {"vertices": [...], "edges": [...]},
"start_vertex_id": "vertex_id",
"context": "context information",
"ignore_direction": true
}
```

4.3 Approach

The analytical framework described in the system architecture section embraced a multi-faceted approach, integrating NLP, graph traversal methods, and LLMs. NLP techniques were applied for processing data within the graphs, and graph traversal methods uncovered intricate relationships and patterns within the data. LLMs were used for their advanced natural language understanding capabilities to analyse all of this data to generate insightful reports about the graph.

Chapter 5

Experiments

5.1 Overview

Throughout the research various experiments were carried out in order to discover the best fit models and techniques for the tasks of creating embeddings from complex data structures, analysing graph structures, traversing graph structures and analysing this information. In this section, these experiments will be explained. The following experiments are detailed:

- Embedding technologies and techniques for graph structures
- Evaluating LLMs capabilities to analyse graph data
- Semantic traversal versus traditional shortest path algorithms
- Evaluating of LLM analysis with semantic traversal data included

5.2 Embedding Technologies and Techniques for Graph Structures

5.2.1 Overview

This experiment sought to evaluate structured data embeddings, focusing specifically on graph representations. The primary objective of the experiment was to identify the most effective embedding technique for graph data, considering its unique challenges and complexities, in order to be able to execute meaningful queries against it. The experiment explored multiple methods for generating embeddings and systematically evaluated their performance across various similarity queries, including graph comparison, context similarity, term, and entity search.

To facilitate the experiment, various embedding techniques ranging from simple key-value embeddings to sophisticated language model-based embeddings, such as OpenAI's AdaV2, BERT, and MistralAI embed. A total of six methods were employed in the experiment to assess different techniques for generating embeddings from JSON-LD graph data.

5.2.2 What's Being Evaluated

Each method undergoes a meticulous evaluation, considering its efficacy in capturing the semantics of the JSON-LD graph: reference graph-a. The experiment employs various types of similarities to comprehensively assess the effectiveness of different embedding techniques for JSON-LD graph data. Each type of similarity serves a distinct purpose, offering insights into different aspects of the data representation and facilitating a thorough evaluation.

- **Graph Comparison:** Graph comparison assesses the overall similarity between two JSON-LD graphs, capturing the structural and semantic similarities in their representations. This metric is essential for evaluating the embedding techniques' ability to preserve the intricate relationships and hierarchies within the graph. It provides a holistic view of how well the embeddings capture the essence of the entire graph structure.

- **Context Matching:** Context matching evaluates how well the embeddings capture the context or overall meaning of a given text. In the context of JSON-LD graph data, understanding the context is crucial for interpreting the relationships and meanings embedded in the graph. This metric helps assess the embeddings' ability to represent the broader context of the structured data, contributing to semantic understanding.

- **Querying Terms and Entities:** Querying terms and entities involves searching for specific terms or entities within the embeddings and measuring their similarity. For applications that involve searching or querying specific elements within the graph, such as finding specific data or identifying entities in a given graph, this metric gauges the embeddings' precision and recall. It ensures that the embeddings can effectively capture and retrieve relevant information from the graph.

5.2.3 The Methods Under Evaluation

- Method 1 involves generating embeddings by encoding key-value pairs extracted from JSON-LD graph vertices using a sentence transformer model.

- Method 2 takes a different route by converting the entire JSON-LD graph into a string and subsequently generating embeddings using a sentence transformer model.
- Method 3 involves using large language models, specifically OpenAI's models: GPT-4-1106-preview and GPT-3.5-turbo-1106. Rich descriptions of the JSON-LD graph data were used to generate embeddings for further analysis. This method introduced a comparative aspect by exploring the performance differences between the GPT-3.5-turbo and GPT-4 models. The evaluation included an examination of their larger context windows and an apples-to-apples comparison to determine which model performs more optimally for the given task. Additionally, prompt engineering techniques are employed to understand how the model's responses vary based on different prompts. This comparative analysis provides insights into the capabilities of the two models and their suitability for generating embeddings from JSON-LD graph descriptions.
- Method 4 uses OpenAI's AdaV2 embedding endpoint to encode followed a similar route to Method 2, converting the JSON-LD graph into a string representation and subsequently generating embeddings, assessing performance across various similarity metrics.
- Method 5 applies BERT model to generate embeddings from a string representation of the JSON-LD graph.
- Method 6 leverages the MistralAI embedding service to encode a string representation of the JSON-LD graph.

In method 1 a class KeyValueEmbedder is initialised with a Sentence-Transformer model (model) and a set of English stopwords from NLTK for

text preprocessing. The preprocess text method performs basic text cleaning, converting text to lowercase, removing HTML tags, and filtering out non-alphabetic words and stopwords.

The embed method takes a dictionary of key-value pairs and preprocesses the text using preprocess text excluding ID's. It encodes each key-value pair using the SentenceTransformer model and appends the resulting embeddings to a list. The method returns a NumPy array containing all the embeddings.

The class provides two similarity calculation methods: similarity of graphs: Measures the cosine similarity between two graph embeddings. similarity of context: Measures the cosine similarity between graph embeddings and a given context.

For Methods 2 through 6 the class structure design described in Subsection 4.2.1 was used to carry out experiments.

The results of this experiment can be found in Section 6.2.

5.3 Evaluating LLMs Capabilities to Analyse Graph Data

5.3.1 Overview

This experiment sought to evaluate the performance of different LLMs in analysing graph data. The task involves providing a graph, and prompting the LLMs to analyse the graph based on various criteria such as semantics, discourse, pragmatics, correctness, relevance, and identifying outliers. A total of eight LLMs were selected to evaluate their individual capabilities in analysing graphs and identifying useful insights. The objective was to

determine the most capable model(s) available up to the end of May 2024 for performing detailed analysis over structured data. The selected LLMs for this evaluation were:

1. GPT-4-turbo
2. GPT-3.5-turbo
3. GPT-4o
4. Gemma
5. Mistral-7b
6. Phi3:medium
7. Phi3:mini
8. Llama3

5.3.2 Configurations

To extract meaningful information from the graph, it was important to use a well-structured prompt to evaluate certain aspects of the data. Providing a prompt template also sets a equal baseline for all the LLMs to analyse the same data in order to measure: given the same input, do we get a similar quality in output. Previous research shows that a robust prompt template is crucial for getting accurate results from LLMs, and that small changes to wording or component selection can have a significant impact on performance [38]. As such, each LLM was configured using the following system prompt:

System prompt

ROLE: You are an advanced language model tasked with analysing graphs. Your goal is to evaluate the graph based on various criteria such as syntax, semantics, discourse, and pragmatics. Additionally, you will assess correctness, relevance, and identify outliers. Your analysis should be thorough and objective, providing clear reasoning for your conclusions.

CRITERIA: Analyse the graph based on the following criteria:

- Semantics: Explain the meaning and interpretation of the vertices and edges.
- Discourse: Evaluate the flow and coherence of paths within the graph.
- Pragmatic: Assess the practical implications and relevance of significant paths.

TASK: You should evaluate:

- Correctness: Verify the accuracy of significant paths.
- Relevance: Assess how relevant the path is to the graph as a whole.
- Reasoning for disregarded paths: Explain why other paths were not chosen.
- Outliers: Identify and analyse any outliers in the graph.

Additionally, another prompt template is designed for prompting the LLM with the graph structure injected:

Prompt template

CONTEXT: You are provided with the following graph: total_graph

Please provide a detailed analysis of the graph.

5.3.3 Evaluation Process

The class structure described in Subsection 4.2.2 was used in this experiment, and each LLM was configured using their default hyperparameters to replicate a like for like evaluation process as best possible. Each LLM was prompted using the Prompt template outline in the previous subsection in which the graph (refer to Listing 8.1 "MOVIE-GRAPH-JSON") was injected using the `PromptTemplate.format` function [39]. This produced an analysis of the graph from each LLM.

5.3.3.1 Judging the Analysis

In order to evaluate the analytical capabilities of each LLM when analysing graph structures, a “panel of judges” would be required in order to make the evaluations of the outputs of each LLM. To accomplish this “Judge’s Panel” the same eight LLMs are set up with their default hyperparameters and the following system prompts to change their way of acting to the role of being a “judge”:

Judge template

Context: You are a judge responsible for analysing and scoring the output of a graph analysis conducted by various large language models. Your task is to evaluate the reasoning, accuracy, and factual statements presented in their analyses.

Graph: (Note: The graph is typically inserted here, but for this document, please refer to Listing 8.1 "MOVIE-GRAPH-JSON".)

Scoring Criteria:

- Reasoning: Score the logical coherence of the large language models analysis. Evaluate the clarity and thoroughness of the explanations provided.
- Accuracy: Assess the correctness of the insights derived from the graph. Verify the relevance of the statements made.
- Factual Statements: Confirm that all factual claims are supported by the data. Check the validity of any external references or examples used.

Use this example output below for your output: Example Output:

```
{
    name: <llm-name>,
    reasoning: <0 to 10>,
    accuracy: <0 to 10>,
    factual: <0 to 10>,
    total\_sum: <0 to 30>
}
```

Your Task:

Using the provided graph and data, evaluate the analysis' from various LLMs according to the criteria above. I will pass you the analysis' 1 by 1

The LLMs demonstrated their capacity to understand from complex data structures and identify meaningful patterns within them. The results of this experiment can be found in Section 6.3.

5.4 Semantic Traversal Over a Graph

5.4.1 Overview

This experiment sought to define and compare a semantic traversal algorithm, which is described in Section 4.2.3. Once a suitable algorithm is defined, the next objective was to compare it against already existing popular traversal algorithms, in an undirected manner over the movie-actor-director graph shown in Figure 5.1, to evaluate its ability to reach a destination from a given source and context.

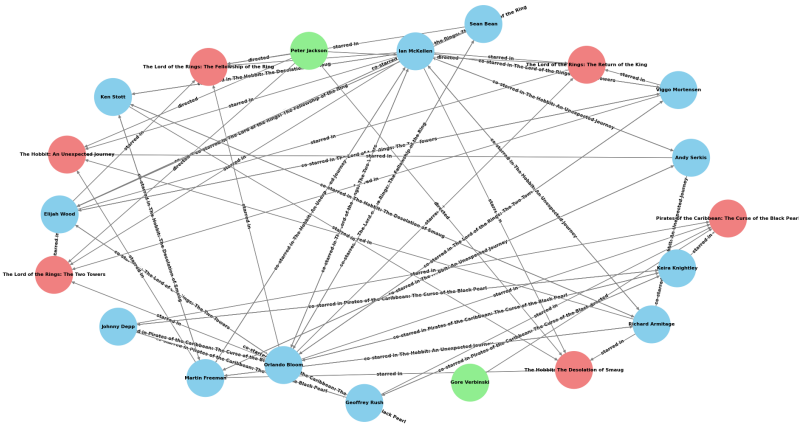


Figure 5.1: A graph of actors, directors, and movies.

5.4.2 Evaluation Process

In this comparison, the semantic traversal algorithm will be evaluated against the following:

- BFS
- DFS

- DSP
- A* using PR scores
- A* using semantic scores

A comparative analysis of the listed traversal algorithms aims to reveal insights into their performance characteristics and computational demands. By examining factors such as path length, time complexity, and space complexity, the differences in how these methods balance trade-offs between speed and memory usage become apparent. Each algorithm is designed to traverse and retrieve a path between the specified source and target vertex using a source and target ID that represent the desired start and destination vertices. However, the semantic traversal algorithm uses a source vertex ID in conjunction with contextual text to guide its traversal. The expected outcome of each traversal function is to yield a possible path from the source to the target vertex, with shorter paths being more desirable. The results of this experiment can be found in Subsection 6.4.

5.5 Evaluating of LLM Graph Analysis With Traversal Information Supplemented

5.5.1 Overview

This experiment sought to determine that if supplemented with additional data about the graph and contextual traversal data would a rise in “Judge’s Panel” score for the LLM be observed. As in the previous experiment described in Section 5.3 the same group of eight LLMs were used to evaluate

their capabilities in analysing graphs now with the traversal data supplemented.

5.5.2 Configuration

In this experiment, each LLM was configured using their default hyperparameters and system prompts as they were in Section 5.3 to ensure a standardised evaluation process. The system prompt provided to each model outlined their role, task, and criteria for their analysis, the same system prompt found in the previous experiment Subsection 5.3.2 was used. Each LLM was provided with same graph as described in Section 5.3 (refer to Listing 8.1 "MOVIE-GRAPH-JSON") containing vertices and edges connecting different movies, actors and directors. This graph served as the foundation for evaluating their analytical capabilities in the initial experiment and would serve as a means to record any observed changes between this and the previous experiments.

5.5.2.1 Changes Versus the Previous Configuration

To facilitate the analysis of the supplemental data, each LLM received the updated context using a modified prompt template from the previous experiment in Section 5.3:

Prompt template

CONTEXT: You are provided with the following graph: total_graph

User context: user_provided_context

Starting from the vertex with the ID start_vertex_id,

The order of visited vertices. order

A dictionary mapping vertex IDs to their corresponding scores. scores

The path leading to the vertex with the highest score. best_path

The score of the vertex with the highest score. best_score

A dictionary mapping vertex IDs to their corresponding paths. paths

Please analyse the graph and its chosen best path. Provide a detailed analysis of the graph, the best path, and the user context.

5.5.3 Evaluation Process

To carry out this updated experiment, the system described in Section 4.2.4 the analysis provider was configured using the best performing embedding system found in the results of Section 6.2, and the previously mentioned configurations (see Subsection 5.5.2). Each LLM was prompted with the graph and traversal data using the prompt template (reference 5.5.2.1) from which each LLM provided its analysis. In order to judge the outputs from this experiment the same "judges panel" described in Section 5.3.3.1 was used. From the results data analysis can be carried out to check whether a positive change in the Judges Panel score can be observed. The results of this experiment can be found in Section 6.5.

Chapter 6

Results

6.1 Overview

The following section will discuss the results of each experiment and offer some commentary about the results and how they relate to the initial research questions proposed in the introduction

- Results for embedding technologies and techniques for graph structures evaluation
- Results for LLMs capabilities to analyse graph data evaluation
- Results for semantic traversal versus traditional shortest path algorithms
- Results for semantic traversal and graph data analysis

6.2 Results for Embedding Technologies and Techniques for Graph Structures Evaluation

6.2.1 Overview

The following section will present and discuss the results of the experiment described in Section 5.2.

6.2.2 Table of Scores

No	Method	Graph Comparison	Context Similarity	Text Search	Entity Search	Mean
1	Key-Value Embedding	0.2799	0.1227	0.2425	0.26	0.226275
2	Stringified Graph Embedding	0.8485	0.4994	0.6171	0.2003	0.541325
3	LLM Inference + Sentence Embedding	0.8797	0.3777	0.5333	0.3907	0.54535
4	OpenAI AdaV2 Embedding	0.9409	0.852	0.8243	0.8489	0.866525
5	BERT	0.9069	0.6737	0.3981	0.3791	0.58945
6	MistralAI Embedding	0.925	0.8409	0.7185	0.7535	0.809475

Table 6.1: Methods

6.2.3 Analysis

When interpreting this data all terms (graph, context, text, and entity) are present or structurally and conceptually similar to graph-a and as such should show a value which represents a higher level of similarity, thus:

- Values which trend towards 1 show a higher level of similarity, and are deemed desirable.
- Values which trend towards 0 show a lower level of similarity, and are deemed less desirable.

6.2.3.1 Method Comparison

The averages provide insight into the overall performance of each method across all the different similarity measures, and as such can be used as a means to rank each method from 1 best to 6 worst.

- **Averages:**
  - Method 1 has the lowest mean score at 0.2263, indicating lacklustre overall performance across all similarity metrics. When inspecting the individual scores, poor performance is observed on all comparisons, as such this method ranks in last place.
  - Method 2 performs well in graph comparison (0.8485), and respectably in text search (0.6171), but has a notable drop in Entity Search (0.2003), resulting in a lower observed mean of 0.5413, thus signalling a lack of exceptional ability to understand across all comparisons made. This method ranks in fifth place.
  - Method 3 performs well in graph comparison (0.8797) but struggles in context similarity and entity search, (0.3777) and (0.3907), respectfully. The mean observed for method 3 was 0.5453 signalling a lack of exceptional ability to understand across all comparisons made. This method ranks in sixth place.
  - Method 4 achieved the highest average score at 0.8665, signalling a deep level of understanding of the structure and concepts detailed. When inspecting the individual scores strong and consistent performance across graph comparison (0.9409), context similarity (0.852), text search (0.8243), and entity search(0.8489) is observed, and as such the method is ranked in first position.

- Method 5 performed exceptionally in graph comparison (0.9069), and showed decent performance in context similarity (0.6737). This method struggled in text and entity search, achieving 0.3981 and 0.3791 respectfully. Its mean score was 0.58945 signalling a trend towards desirable performance.
- Method 6 performed exceptionally well in graph comparison (0.925), and strong performance across context similarity, text and entity search, achieving 0.8409, 0.7185, and 0.7535 respectfully. The high mean score of 0.8095 signals a deep level of understanding of the structure and concepts detailed, placing this method in second.

In light of this, the original table can be updated to reflect each method's rank and sort them by their respective ranks: ranked and sorted table:

No	Name	Graph Comparison	Context Similarity	Text Search	Entity Search	Mean	Rank
4	OpenAI AdaV2 Embedding	0.9409	0.852	0.8243	0.8489	0.866525	1
6	MistralAI Embedding	0.925	0.8409	0.7185	0.7535	0.809475	2
5	BERT	0.9069	0.6737	0.3981	0.3791	0.58945	3
3	LLM Inference + Sentence Embedding	0.8797	0.3777	0.5333	0.3907	0.54535	4
2	Stringified Graph Embedding	0.8485	0.4994	0.6171	0.2003	0.541325	5
1	Key-Value Embedding	0.2799	0.1227	0.2425	0.26	0.226275	6

Table 6.2: Methods ranked

6.2.4 Recommendations

Given the results of the evaluation, Method 4 (OpenAI's AdaV2 Embedding model) can be confidently selected as the primary choice going forward. This method observed consistently high performance across all measures, making it a robust and reliable choice.

6.2.5 Conclusion

This experiment highlighted the varying capabilities of different embedding models and techniques and their ability to understand graph structures and their underlying concepts. By systematically evaluating their performance, the experiment identified the strengths and limitations of each model, guiding the selection of the most suitable model for complex data structure understanding. In light of the results of the experiment it can be said with a great deal of confidence that: Yes, meaningful information can be understood and searched from embedding representations, answering the question:

> "RQ1: Can meaningful information be understood from embedding representations of complex graph data structures?".

It should be noted that OpenAI has released new embedding models: text-embedding-3-small and text-embedding-3-large after this evaluation was completed. These models are claimed to have better performance than their predecessor text-embedding-ada-002 [40]. With some initial experimentation on the performance of the text-embedding-3-small in this scenario, the ability to capture at least the same if not a greater amount of understanding was observed, but more robust research is required to come to a concrete conclusion if this is indeed the case.

6.3 Results for LLMs Capabilities to Analyse Graph Data Evaluation

6.3.1 Overview

The following section will present and discuss the results of the experiment discussed in Section 5.5. For simplicity, and later comparisons, this set of results will be referred to as **"Group A"** going forward.

6.3.2 Table

model	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	Mean	MAX	MIN
Llama3	23	27	25	27	26	29	24	27	26	29	23
Mistral	25	21	24	26	29	30	28	24	25.875	30	21
GPT-3.5-t	23	25	26	23	27	26	26	23	24.875	27	23
Phi3-med	25	24	27	24	24	24	26	24	24.75	27	24
Phi3-mini	25	21	24	26	27	24	24	24	24.375	27	21
GPT-4-t	23	25	29	20	27	26	22	22	24.25	29	20
Gemma	21	23	21	23	27	24	24	24	23.375	27	21
GPT-4o	24	24	26	23	21	22	20	19	22.375	26	19

Table 6.3: Group A - Overall scores

For a breakdown of the Group A Overall scores please refer to the appendix tables Table A.1 - Group A - Reasoning scores, Table A.2 - Group A - Accuracy scores, Table A.3 - Group A - Factual scores.

6.3.3 Analysis

In order to review the Group A data from the "Judges Panel" the output data was populated into Excel for review. Each Judge had the potential to score the model between 0-30, with a maximum total points to be achieved: 30 x 8 Judges = 240. Given this, the following can be derived: Summed scores trending towards 240 indicate strong performance, whereas summed scores trending towards 0 indicate weak performance.

The result of this process produced the following table:

model	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	Mean	MAX	MIN	SUM	Rank
Llama3	23	27	25	27	26	29	24	27	26	29	23	208	1
Mistral	25	21	24	26	29	30	28	24	25.875	30	21	207	2
GPT-3.5-t	23	25	26	23	27	26	26	23	24.875	27	23	199	3
Phi3-med	25	24	27	24	24	24	26	24	24.75	27	24	198	4
Phi3-mini	25	21	24	26	27	24	24	24	24.375	27	21	195	5
GPT-4-t	23	25	29	20	27	26	22	22	24.25	29	20	194	6
Gemma	21	23	21	23	27	24	24	24	23.375	27	21	187	7
GPT-4o	24	24	26	23	21	22	20	19	22.375	26	19	179	8

Table 6.4: Group A: Overall scores (ranked highest to lowest sum of scores)

6.3.3.1 Model Rankings

1. Llama3 has placed in first as it has scored the highest, scoring a total of 208 points out of a total possible point(s) of 240 (30×8) and has the highest average grade of 26 indicating that it would appear to be the "most capable".

2. Mistral placed second behind Llama3 separated by only 1 point(s), scoring 207 / 240 point(s). There is a small margin of 0.125 between Llama3 and Mistrals averages. These two model's overall scores are separated by a small margin indicating similar performance, with the next closest model being 8 point(s) behind Mistral.

3. GPT-3.5-Turbo placed in third, 8 point(s) behind Mistral. There is a 1 point margin between its average and Mistral in second place.

4. Phi3-med placed in forth, it's very close to GPT-3.5-t, with just a 0.125 margin in the average score and a 1 point difference in sum score.

5. Phi3-mini ranks fifth, just slightly below Phi3-med by 0.375 in average score and 3 in sum score.

6. GPT-4-turbo in sixth closely follows Phi3-mini, with a difference of 0.125 in average score and 1 point in sum score.

7. Gemma ranks seventh, trailing GPT-4-t by 0.875 in average score and 7 in sum score.

8. There is a cluster of models separated by less 5.0 point(s) signalling similar performance among these models:

9. In third: GPT-3.5-turbo scored 199 / 240 point(s).

10. Forth: Phi3-medium with 198 / 240 point(s).

11. Fifth: Phi3-mini with 195 / 240 point(s), and in sixth with 194 / 240 point(s) is GPT-4-turbo.

12. These models also had similar average scores, with a difference of less than 1.0 point(s). The next cluster of models include Gemma and GPT-4o, placing seventh and eight respectively.

13. GPT-4o ranks last, it scored the worst receiving 179 / 240 point(s) representing a 29 point(s) difference between it and the highest performer Llama3, in the initial analysis of the graph structures according to the “Judges Panel”.

6.3.3.2 Scoring Range

The range of upper and lower values observed for each of the LLMs scores were as follows:

- Llama3 had a range of 6 points between its highest and lowest score.
- Mistral had a range of 9 points between its highest and lowest score.
- GPT-3.5-turbo had a range of 4 points between its highest and lowest score.

6.3 Results for LLMs Capabilities to Analyse Graph Data Evaluation

- Phi3-medium had a range of 3 points between its highest and lowest score.
- Phi3-mini had a range of 6 points between its highest and lowest score.
- GPT-4-turbo had a range of 9 points between its highest and lowest score.
- Gemma had a range of 6 points between its highest and lowest score.
- GPT-4o had a range of 7 points between its highest and lowest score.

From this, the following comments can be made:

1. The models which represented the smallest range between high and low scores of 3 and 4 were: Phi3-medium and GPT-3.5-turbo respectively, signalling that the judges scored these two models to have the most desirable variation in range.
2. The models which represented a range of 6, 6, 6, and 7 were: Llama3, Phi3-mini, Gemma, GPT-4o respectively had a greater range between their highest and lowest score signalling that these models have greater variation in range according to the judges scores.
3. The models which had the largest range of 9 each were: Mistral and GPT-4-turbo. The judges scores for these models show the largest disparity between highest and lowest score, this could be interpreted as the models being perceived as less consistent or there simply could be outliers in the scores.

6.3.4 Recommendation

In light of these results, Llama3 is the top recommendation due to it producing the highest summed score of all the LLMs evaluated. Although, it should be noted that all the models evaluated scored well, with the lowest model achieving 74.58% of the total points available.

6.3.5 Conclusion

These findings indicate that while some models perform better than others, some potentially struggle with consistency and the choice of model might depend on the specific requirements, such as consistency versus overall performance. However, all models performed reasonably well and were capable of extracting meaningful insights from graph data and as such research question two:

> "RQ2: Can LLMS carry out detailed analysis of complex graph data structures?"

can be answered: yes.

6.4 Results for semantic traversal versus traditional shortest path algorithms

6.4.1 Overview

The following section will present and discuss the results of the experiment described in Section 5.3.

6.4 Results for semantic traversal versus traditional shortest path algorithms

Algorithm	Path	Path Length	Time Complexity	Space Complexity
BFS	[’474996da-1982-40a6-9175-329324bc5bad’, ’30a2b51d-27f4-4cbd-bf7b-c7b72479d244’, ’1946b274-fd20-4e44-b1c2-a099f61631cc’, ’646a1bc2-18de-4e40-84b3-1b34c16ce688’, ’35d69c0c-5f62-474e-a80b-e71ce67feeb3’, ’035bbdbb-95b6-428c-bbe1-3275eb618ae1’, ’32e683df-c89d-41bf-a4b7-c854033b22c0’, ’bbf679b0-ff13-44d1-95c2-abaac79cefae’, ’d1ae107c-fbf8-4d22-8039-3ca0fd5e25c7’, ’384521da-fd61-473f-b8c4-b3172a144b81’, ’64a8f336-3d2a-4bcc-aec9-6685982365cf’, ’42dcc9fc-1862-45d2-8d71-138d4dc23d2e’, ’43f96a08-135c-479e-948f-a0d9eb0a5931’, ’a205f32c-cf5f-42d6-9b8c-bdc5d3b01ead’, ’cb1637eb-4b6a-4c41-ae98-1c19bf058a47’, ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’]	16	O(V)	O(V)
DFS	[’474996da-1982-40a6-9175-329324bc5bad’, ’035bbdbb-95b6-428c-bbe1-3275eb618ae1’, ’cb1637eb-4b6a-4c41-ae98-1c19bf058a47’, ’43f96a08-135c-479e-948f-a0d9eb0a5931’, ’d1ae107c-fbf8-4d22-8039-3ca0fd5e25c7’, ’42dcc9fc-1862-45d2-8d71-138d4dc23d2e’, ’a205f32c-cf5f-42d6-9b8c-bdc5d3b01ead’, ’35d69c0c-5f62-474e-a80b-e71ce67feeb3’, ’384521da-fd61-473f-b8c4-b3172a144b81’, ’0186eb7a-ce10-4724-86c3-b929e1d11b06’, ’a1cd48b5-e1ff-490a-91b4-37cf95b1f3cb’, ’2c478fb9-afb3-4b00-9198-0ef141212057’, ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’]	13	O(V)	O(V)
DSP	[’474996da-1982-40a6-9175-329324bc5bad’, ’1946b274-fd20-4e44-b1c2-a099f61631cc’, ’384521da-fd61-473f-b8c4-b3172a144b81’, ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’]	4	O((V)logV)	O(V)
A* using PR scores	[’474996da-1982-40a6-9175-329324bc5bad’, ’1946b274-fd20-4e44-b1c2-a099f61631cc’, ’384521da-fd61-473f-b8c4-b3172a144b81’, ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’]	4	O((V)logV)	O(V)
A* using semantic scores	[’474996da-1982-40a6-9175-329324bc5bad’, ’1946b274-fd20-4e44-b1c2-a099f61631cc’, ’384521da-fd61-473f-b8c4-b3172a144b81’, ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’]	4	O((V)logV)	O(V)
Semantic traversal best path	[’474996da-1982-40a6-9175-329324bc5bad’, ’30a2b51d-27f4-4cbd-bf7b-c7b72479d244’, ’384521da-fd61-473f-b8c4-b3172a144b81’, ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’]	4	O((V)logV)	O(V)

Table 6.5: Traversal results

6.4.2 Table of Results

6.4.3 Analysis

In the table of traversal results (Table 6.5, the individual performance of each algorithm can be observed. From this, the following comments can be made:

- The BFS algorithm explored the graph level by level, visiting each vertex once until reaching the target vertex. It produces a path of length 16 and has a space and time complexity of O(V), where V is the number of vertices. BFS produced the longest path of all algorithms tested.

- The DFS algorithm explored the graph by travelling towards the peripheries of the graph before backtracking. It reached the target vertex ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’ on a path of length 13. DFS has a space and time complexity of O(V) due to the recursion stack in the worst case where the graph is a linear chain. DFS produced the second-longest path of all algorithms tested.

- The DSP algorithm found a path of length 4 to the target vertex ’238471cc-3fa8-4b74-8ea5-9b2b890dbc84’ in the graph. It has a time complexity of O((V)log V), and space complexity of O(V) as it stores the shortest path estimates and uses a priority queue (min-heap). It produced the shortest path length observed so far.

- The A* algorithm using PR scores also found a path of length 4 to the target vertex, matching the performance of DSP. It leveraged the importance of vertices, using as a heuristic (PR) to efficiently navigate the graph. This approach has a time complexity of O((V)log V), and a

space complexity of O(V) for maintaining the priority queue and other auxiliary data structures.

- The A* search algorithm using semantic scores also found a path of length 4 to the target vertex by leveraging semantic relevance as a heuristic guide. This approach matched the time complexity of DSP and A* + PR: O((V)log V), and a space complexity of O(V).

- The semantic traversal algorithm also found a path of length 4 by prioritising semantically relevant vertices according to the given context of 'an adventure about pirates' as its guide. It also has a time complexity of O((V)log V), combining traversal with heuristic evaluations based on semantic scoring, and a space complexity of O(V).

BFS and DFS, which do not use any heuristic or weight considerations, produced longer paths of 16 and 13 vertices, respectively, indicating they are less efficient in identifying the shortest or most contextually relevant paths in this scenario. DSP, A* using PR scores, and A* using semantic scores, all yielded optimal paths of length 4, demonstrating their effectiveness in finding the shortest paths in weighted graphs or when guided by heuristics. The semantic traversal algorithm designed as part of this research produced an optimal path of length 4, matching the capabilities of well-established algorithms like DSP and A*. This indicates that the semantic traversal algorithm is capable of identifying significant or relevant paths efficiently, leveraging contextual information to guide the traversal process.

6.4.4 Conclusion

The results confirm that the semantic traversal algorithm can indeed match the capabilities of well-established algorithms in identifying significant or

relevant paths. By incorporating contextual information into the traversal process, the semantic traversal algorithm effectively produced an optimal path similar to those found by traditional algorithms like DSP and A*. Thus, the research question:

> "RQ3: Can significant or relevant paths be identified between a starting vertex and contextually relevant vertex?"

is answered: yes, significant or relevant paths can be identified between a starting vertex and a contextually relevant vertex using the semantic traversal algorithm.

6.5 Results for LLM Graph Analysis With Traversal Information Supplemented

6.5.1 Overview

The following section will show and discuss the results of the experiment discussed in Section 5.5.

6.5.2 Tables

NAME	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG	MAX	MIN
GPT-4o	23	27	26	24	27	27	26	24	25.5	27	23
GPT-4-t	24	23	23	23	26	25	26	24	24.25	26	23
GPT-3.5-t	23	21	24	26	27	29	27	24	25.125	29	21
Gemma	21	16	24	26	26	27	26	24	23.75	27	16
Llama3	20	24	26	26	29	30	26	26	25.875	30	20
Mistral	26	17	24	26	27	24	25	24	24.125	27	17
Phi3-med	19	20	24	23	27.5	27	27	24	23.9375	27.5	19
Phi3-mini	22	22	24	26	26	27	24	24	24.375	27	22
AVG	22.25	21.25	24.375	25	26.9375	27	25.875	24.25	24.6171875	27	21.25
MAX	26	27	26	26	29	30	27	26	25.875	30	26
MIN	19	16	23	23	26	24	24	24	23.75	26	16

Table 6.6: Group B: Overall scores

For a breakdown of the Group B Overall scores please refer to the appendix tables Table A.4 - Group B - Reasoning scores, Table A.5 - Group B - Accuracy scores, Table A.6 - Group B - Factual scores.

6.5.3 Analysis

In order to review the Group B data from the Judges Panel in the same methodology as Subsection 6.3.3 the output data in was populated into Excel for review. As previously established, each Judge had the potential to score the model between 0-30, with a maximum total points to be achieved: 30 × 8 Judges = 240. Given this, the following can be derived: Summed scores trending towards 240 indicate strong performance, whereas summed scores trending towards 0 indicate weak performance.

The result of this process produced the following table:

NAME	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG	MAX	MIN	SUM	Rank
Llama3	20	24	26	26	29	30	26	26	25.875	30	20	207	1
GPT-4o	23	27	26	24	27	27	26	24	25.5	27	23	204	2
GPT-3.5-t	23	21	24	26	27	29	27	24	25.125	29	21	201	3
Phi3-mini	22	22	24	26	26	27	24	24	24.375	27	22	195	4
GPT-4-t	24	23	23	23	26	25	26	24	24.25	26	23	194	5
Mistral	26	17	24	26	27	24	25	24	24.125	27	17	193	6
Phi3-med	19	20	24	23	27.5	27	27	24	23.9375	27.5	19	191.5	7
Gemma	21	16	24	26	26	27	26	24	23.75	27	16	190	8

Table 6.7: Group B - Overall scores (ranked highest to lowest sum of scores)

6.5.3.1 Model Rankings

To cross-examine, the performance of Group A rank and Group B rank were then plotted in a lined scatter plot in Excel to visualise how the additional information supplemented to Group B impacted the overall output observed in Figure 6.1:

The resulting graph initially indicates that:

- Llama3 and GPT-3.5-turbo remained the same ranking(s).

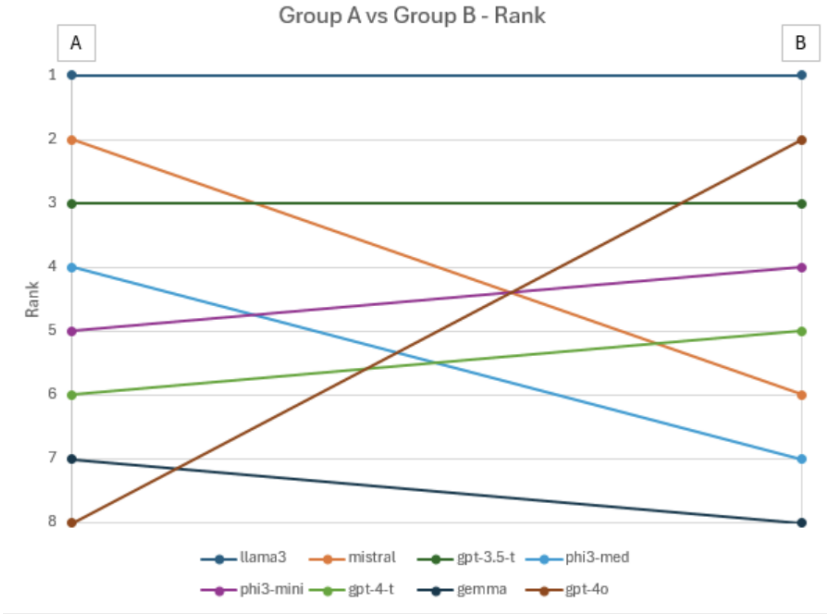


Figure 6.1: Group A versus Group B rank

- Phi3-mini, GPT-4-turbo and GPT-4o all appear to have gained in their ranking(s) with the additional information supplemented; with GPT-4o possessing the most significant increase in rank.
- Mistral, Phi3-medium and Gemma all appear to have declined in ranking with the additional information supplemented; with Mistral and Phi3-med dropping significantly, -4 places, and -2 places respectfully.

To gain insights, a review was completed to visualise the Group A and Group B for each model, to verify any trends or identify outliers within the data sets. It's expected that the scores should see a higher value in the outcome observed in Group B, when supplemented with additional data.

6.5.3.2 Scoring Change

The individual scoring of each model is plotted to gain an understanding of the nature of the change:

In Figure 6.2 GPT-4o is the only Judge to have given a lower score for Group B with a difference of -1, all other Judges showed a higher score for Group B. On initial review of Avg, Max, and Min, the data spread appears to indicate that a positive change in score has been observed for Group B.

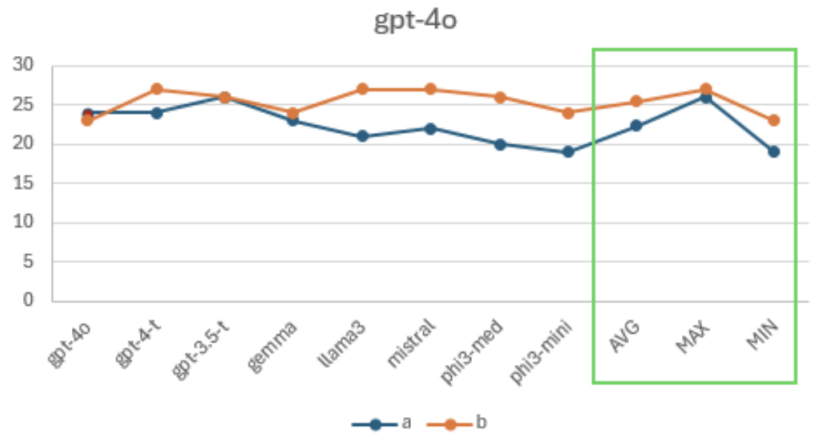


Figure 6.2: GPT-4o Group A versus Group B

In Figure 6.3 GPT-4o, GPT-4-t, Gemma and Phi3-mini Judges have all given a lower score for Group B; with a difference of -3, -3, -1, -1, respectfully. On initial review of Avg, Max and Min, the data spread appears to have increased with the mean staying consistent.

In Figure 6.4 GPT-4-turbo and GPT-3.5-t Judges have given a lower score for Group B; with a difference of -4, -2, respectfully. On initial review of Avg, Max and Min, the data spread appears to have increased with the mean staying consistent.

In Figure 6.5 GPT-4-t and Llama3 Judges have all given a lower score for Group B; with a difference of -7, -1, respectfully. On initial review of Avg, Max and Min, the data spread appears to have increased with the mean staying consistent.

In Figure 6.6 GPT-4o, GPT-4-t, Gemma and Phi3-mini Judges have all

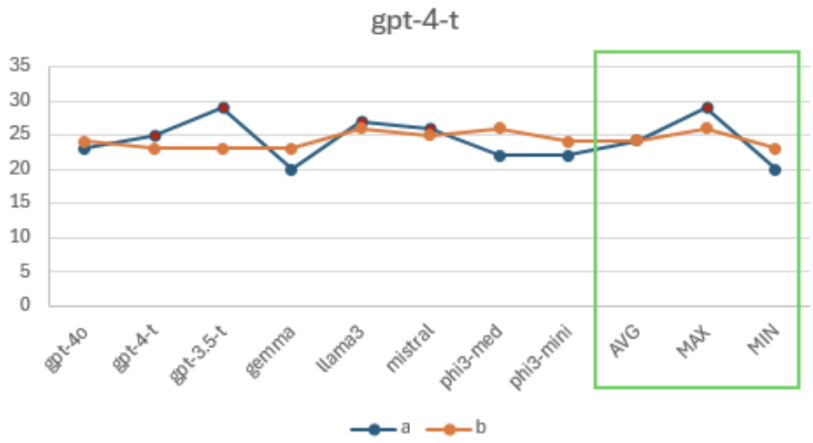


Figure 6.3: GPT-4-turbo Group A versus Group B

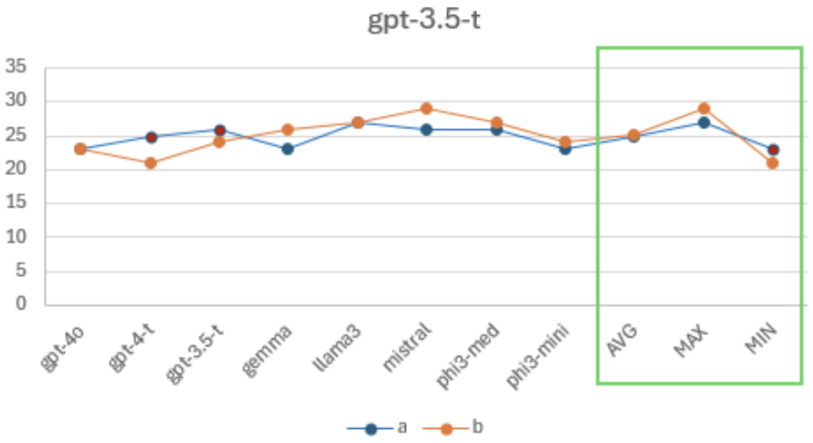


Figure 6.4: GPT-3.5-Turbo Group A versus Group B

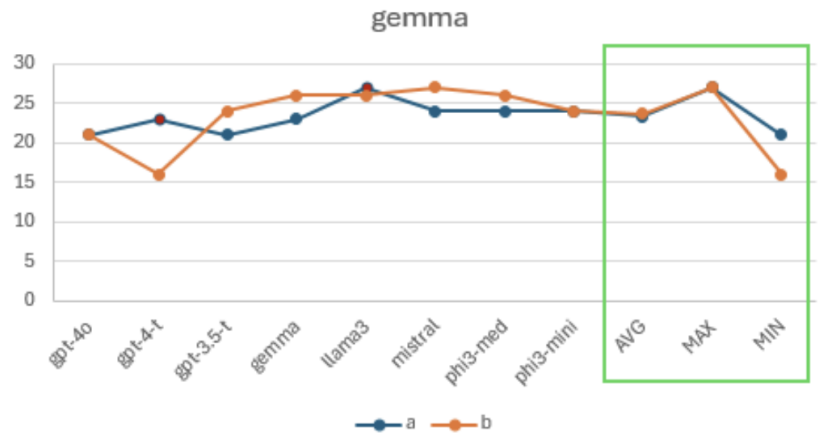


Figure 6.5: Gemma Group A versus Group B

given a lower score for Group B; with a difference of -3, -3, -1, -1, respectfully. On initial review of Avg, Max and Min, the data spread appears to have increased with the mean staying consistent.

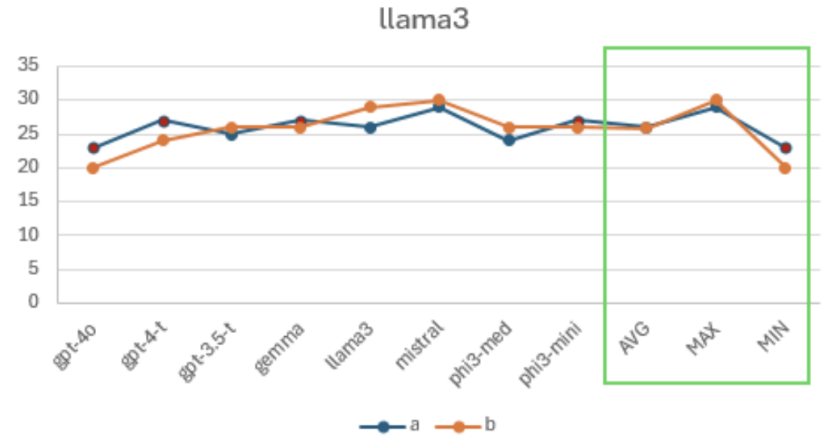


Figure 6.6: Llama3 Group A versus Group B

In Figure 6.7 GPT-4-t, Llama3, mistral, and Phi3-medi Judges have all given a lower score for Group B; with a difference of -4, -2, -6, -3, respectfully. On initial review of Avg, Max and Min, the data spread appears to have stayed consistent with an overall value drop across all.

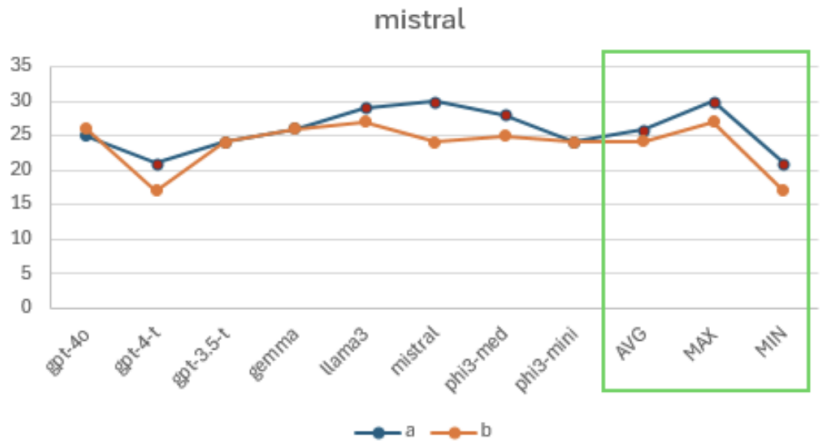


Figure 6.7: Mistral Group A versus Group B

In Figure 6.8 GPT-4o, GPT-4-t, GPT-3.5-t and Gemma Judges have all given a lower score for Group B; with a difference of -6, -4, -3, -1, respectfully. On initial review of Avg, Max and Min, the data spread appears greater and drops significantly, however the mean staying consistent.

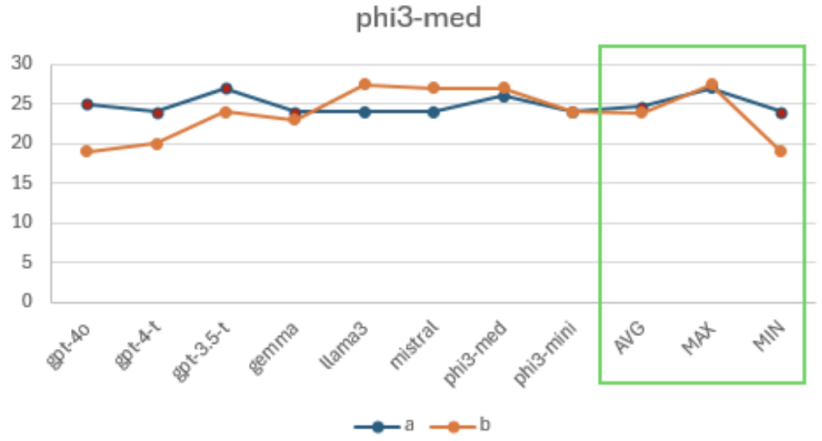


Figure 6.8: Phi3-medium Group A versus Group B

In Figure 6.9 GPT-4o and Llama3 Judges have all given a lower score for Group B; with a difference of -3, -1, respectfully. On initial review of Avg, Max and Min, the data spread appears similar and the mean staying

consistent.

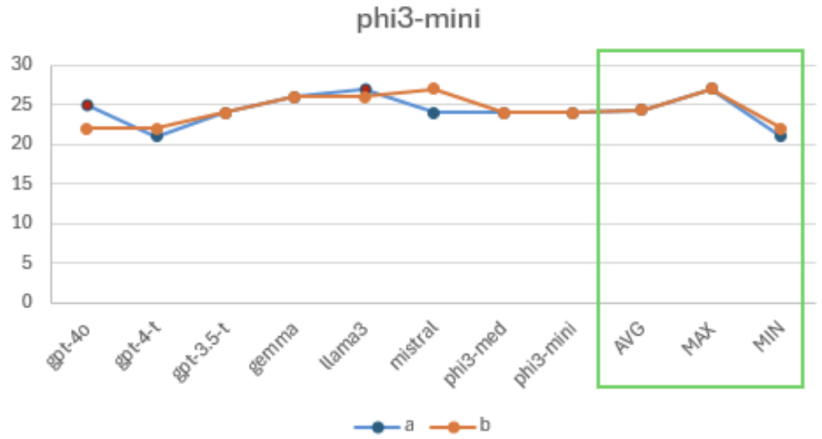


Figure 6.9: Phi3-mini Group A versus Group B

6.5.3.3 Calculating a Final Rank

To analyse the differences between Group A and Group B, a comparison system was designed. A comparison table was created to calculate the Final Rank, where 1 is the best and 8 is the worst. The method involved subtracting the Judge Panel value for each model in Group A from the corresponding value in Group B. The following rules were established:

- Where a positive integer is observed, it can be assumed there has been a positive effect due to the supplement of the additional information provided to Group B.
- Where a zero is observed, it can be assumed that there has been no change due to the supplement of additional information provided to Group B.
- Where a negative integer is observed, it can be assumed that there has been a decline due to the supplement of the additional information

provided to Group B.

The resulting table can be seen in the appendix in Figure A.1: Final results table

To assess the overall comparison, the following steps were completed:

- Initial review to identify the most positively effected, to have the highest “Sum” score.
- Where the score for “Sum” was equal between models, the next step for comparison would review “improved”. The highest “improved” score was determined as favourable.
- Where the score for “improved” was equal between models, the next step for comparison would review “Same”. The highest “Same” score was determined as favourable.
- Where the score for “Same” was equal between models, the next step for comparison would review “worse”. The lowest “worse” score was determined as favourable.
- Where the score for MODEL “worse” was equal between models, the next step for comparison calculated the negative integer score, with the calculated score closest to zero determined favourable.

The table depicted in Figure 6.10 resulted from the comparison and the new rank order was established.

The updated ranking was confirmed and populated into the lined scatter plot in Excel to again visualise how additional information supplemented to Group B benefited each model observed in Figure 6.11.

The resulting graph indicates that:

MODEL	Judges Panel of Scores - *Group B - Group A*								Review					
NAME	gpt-4o	gpt-4-t	gpt-3.5-t	gemma	llama3	mistral	phi3-med	phi3-mini	improved	same	worse	Equation	Sum	Final Rank
gpt-4o	-1	3	0	1	6	5	6	5	6	1	1	6+1-1	6	1
gpt-4-t	1	-2	-6	3	-1	-1	4	2	4	0	4	4+0-4	0	6
gpt-3.5-t	0	-4	-2	3	0	3	1	1	4	2	2	4+2-2	4	2
gemma	0	-7	3	3	-1	3	2	0	4	2	2	4+2-2	4	3
llama3	-3	-3	1	-1	3	1	2	-1	4	0	4	4+0-4	0	5
mistral	1	-4	0	0	-2	-6	-3	0	1	3	4	1+3-4	0	8
phi3-med	-6	-4	-3	-1	3.5	3	1	0	3	1	4	3+1-4	0	7
phi3-mini	-3	1	0	0	-1	3	0	0	2	4	2	2+4-2	4	4

Figure 6.10: Final results table ranked

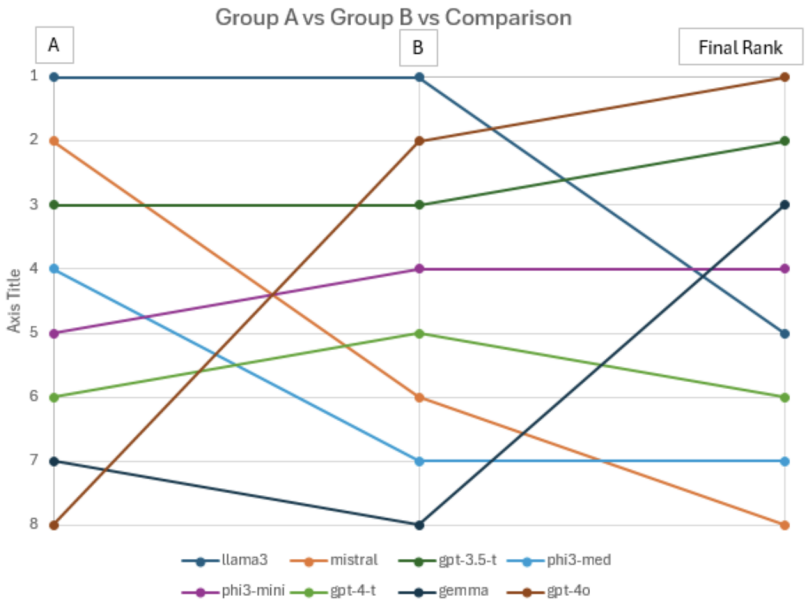


Figure 6.11: Group A versus Group B vs Comparison

1. Initially Llama3 appeared to be the preferred model ranking as #1 in both Group A and Group B overall, however in terms of improvement given additional data, its ranking resulted in a final position of #5.

2. GPT-4o gained significantly during the initial ranking, moving from #8 (Group A) to #2 (Group B), with the final comparison placing this model as #1.

3. GPT-3.5-turbo remained in the same #3 position in both Group A and Group B overall, however with the additional information, increasing its ranking resulted in a final position of #2.

4. Gemma declined in ranking given additional data, moving from #7 to #8, with the final comparison placing it as #3.

5. Initially, Phi3-mini and GPT-4-turbo appear to have gained ranking with the additional information, with Phi3-mini remaining #4 post comparison and GPT-4-turbo dropping back to #6.

6. Initially, Mistral and Phi3-med appear to declined in ranking with the additional information, moving from #2 and #4 respectfully to #6 and #7. Post comparison Phi3-med remained as #7 with Mistral plummeting to #8.

To further understand some changes within the data, MiniTab [41] was used to visualise Group A vs Group B for each model, and to verify the method of “Final Ranking” to match with the most optimal model for Group B. Box Plots and Histograms can be seen in the Appendix under the subsection: A.3.1 Minitab boxplots and histograms.

6.5.4 Conclusion

In light of this analysis, the following table assessing the improvement or deterioration of performance is derived Table 6.8: Final assessment.

NAME	Comment/Review	Final Rank
gpt-4o	Improvement Observed	1
gpt-3.5-t	Small improvement but less reliable.	2
gemma	Small improvement but less reliable.	3
phi3-mini	No specific improvement and less reliable.	4
llama3	No specific improvement with indication of poorer performance.	5
gpt-4-t	No specific improvement with indication of poorer performance	6
phi3-med	Decline in performance apparent.	7
mistral	No specific improvement with indication of decline in performance.	8

Table 6.8: Final Assessment

Based on the final assessment table: Table 6.8: Final assessment, the benefit of supplementing semantic graph traversal data with LLM analysis appears to be model-dependent. While GPT-4o significantly improved with additional data, others showed minimal benefit or even a decline in performance. Thus, the benefit of supplementing the data is not universal across all models. With regards the research question:

> "RQ4: Does the LLMs analysis of the graph improve as a result of supplementing semantic graph traversal data?"

It's reasonable to conclude that the benefit of supplementing semantic traversal data with LLM analysis is model-dependent. Therefore, "it depends" is a practical answer for now, but additional research is needed to achieve a more concrete understanding.

Chapter 7

Conclusion

7.1 Overview

Throughout the research, we discovered the ability of embedding models to understand complex data structures such as graphs. This fed into the semantic traversal algorithm, which shows promise when compared against some traditional methods to identify relevant paths between two vertices, though further research is still required. It also explored and assessed LLMs and their ability to comprehend graph structures, resulting in a positive finding. All the findings of these experiments were then combined to assess whether a deeper level of understanding could be achieved when supplementing the LLM with additional traversal data. The result of this concluded that supplementation of additional data was model-dependent, and suggests that more research is required in order to give a concrete yes or no answer.

7.2 Limitations

7.2.1 Sample Size of LLMs

A limitation of this research is the relatively small sample size of eight LLMs used in the analysis. Both Group A and Group B, as well as the judging task, were based on a limited number of LLMs. To obtain a more statistically significant representation of their capabilities, future studies should aim to evaluate a minimum of twenty-one LLMs. This expanded sample size would enhance the robustness of the conclusions drawn and mitigate any anomalies or biases stemming from an insufficiently representative sample.

7.2.2 Variety in Graph Traversal Data

The research primarily tested the graph traversal algorithm on a single dataset, which limits the understanding of its ability to generalise of the findings. There was minimal testing across varying graph sizes, contexts, and datasets. To determine the domains in which the algorithm is most useful and to assess its versatility, it's advised to conduct more extensive testing across different types and sizes of graphs. This broader range of data would help evaluate the algorithm's performance in diverse scenarios and provide a more comprehensive understanding of its potential.

7.3 Future Works

One of the key questions from this research is whether the supplemental data was genuinely helpful or simply added noise for the LLMs to process. To address this, further evaluation is needed in several areas:

1. Testing additional LLMs, such as Anthropic's Claude or Google Gemini.

2. Experimenting with different wordings of the prompt template.

3. Experimenting with few-shot scenarios. We gathered and examined results tested on a one-shot scenario, but the ability to elaborate on insights from graphs in a questions and answers style could yield better results / insights.

4. Assessing the use of tools or function calls for deeper analysis, such as sentiment analysis, classification, and tagging.

Another important question is how well the semantic traversal algorithm generalises across different contexts and datasets. With only a minor amount of experimentation outside the dataset presented in this research, preliminary tests showed positive signs, but undoubtedly more robust evaluation of this aspect is essential to determine the algorithm's versatility and reliability. Additionally, in this research, the algorithm operates with complete knowledge of the entire search space before executing its task. However, evaluating the algorithm's performance in a dynamic search space, where the environment or data changes over time, presents an interesting challenge. Controlling the limits of this dynamic search space could involve strategies such as:

1. Max Hop Counter: Implementing a maximum hop counter to limit the number of hops the algorithm can take.

2. Depreciation Factor with Minimum Score Threshold: Introducing a depreciation factor along with a minimum score threshold to penalise paths that exceed a certain lengths or complexity.

7.4 Ethical Issues

The “Six Degrees of Separation” concept is an idea that any two people on planet earth can be connected on average by six hops [42]. This concept shows connections which might seem unlikely, the traversal algorithm outlined could uncover connections to contexts that may not have otherwise been considered. If not carefully regulated and ethically guided, this could lead to misuse, enabling the potential to profile individuals through indirect associations causing privacy concerns, misinterpretations, and unwarranted accusation.

References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson Education, Inc., 2010. vii, 7, 8

[2] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge, UK: Cambridge University Press, 2010. 1

[3] M. Needham and A. E. Hodler, *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, Inc., 2019. 1, 6, 7, 13

[4] F. A. de Fernando, D. C. G. Pedronette, G. J. de Sousa, L. P. Valem, and I. R. Guilherme, "Rade+: A semantic rank-based graph embedding algorithm," *International Journal of Information Management Data Insights*, vol. 2, no. 1, p. 100078, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667096822000210 1, 16

[5] R. Gould, *Graph Theory*, reprint ed. Courier Corporation, 2013. 6

[6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. 8

[7] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems*

Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968. [Online]. Available: https://doi.org/10.1109/tssc.1968.300136 8

[8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107–117, 1998, proceedings of the Seventh International World Wide Web Conference. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016975529800110X 8, 9, 15

[9] A. Hogan, A. Harth, and S. Decker, "Reconrank: A scalable ranking method for semantic web data with context," in *Proceedings of the Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006)*. Semantic Web Knowledge Base Systems (SSWS 2006), 2006, in conjunction with the International Semantic Web Conference (ISWC 2006). [Online]. Available: http://hdl.handle.net/10379/492 9

[10] G. Klyne and J. J. Carroll, "Resource description framework (rdf): Concepts and abstract syntax," Feb 2004, w3C Recommendation 10 February 2004. [Online]. Available: http://www.w3.org/TR/rdf-concepts/ 9

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781 9, 11

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," pp. 1532–1543, 01 2014. 9, 11

[13] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975. [Online]. Available: http://doi.acm.org/10.1145/361219.361220 9

[14] A. Singhal and I. Google, "Modern information retrieval: A brief overview," *IEEE Data Engineering Bulletin*, vol. 24, 01 2001. 9

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805 9, 12

[16] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008. 9

[17] Meta, "Introducing meta llama 3: The most capable openly available llm to date," 2023, visited: 9th Aug 2024. [Online]. Available: https://ai.meta.com/blog/meta-llama-3/ 10

[18] K. Hu, "Chatgpt sets record for fastest-growing user base - analyst note," 2023, visited: 11th Aug 2024. [Online]. Available: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-02/ 10

[19] P. Gmyrek, J. Berg, and D. Bescond, "Generative ai and jobs: A global analysis of potential effects on job quantity and quality," *ILO Working Paper*, vol. 96, 2023. 10

[20] G. Lazaroiu and E. Rogalska, "How generative artificial intelligence technologies shape partial job displacement and labor productivity

growth," *Oeconomia Copernicana*, vol. 14, no. 3, pp. 703–706, 2023. 10

[21] R. M. Reese, *Natural Language Processing with Java Cookbook*. Packt Publishing, 2019. 11

[22] A. Chliaoutakis, Y. Varelas, E. Voutsakis, E. Petrakis, and E. Milios, "Information retrieval by semantic similarity. int j semantic web inf syst," *Int. J. Semantic Web Inf. Syst.*, vol. 2, pp. 55–73, 07 2006. 11

[23] L. Tunstall, L. von Werra, and T. Wolf, *Natural Language Processing with Transformers, Revised Edition*. O'Reilly Media, Inc., 2022. 11, 12

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165 12

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762 12, 13

[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998. 14

[27] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2009. 14

[28] R. Diestel, *Graph Theory*, 4th ed. Springer, 2013. 14

[29] T. Hegeman and A. Iosup, "Survey of graph analysis applications," *CoRR*, vol. abs/1807.00382, 2018. [Online]. Available: http://arxiv.org/abs/1807.00382 14

[30] M. Kolomeets, A. Chechulin, and I. V. Kotenko, "Social networks analysis by graph algorithms on the example of the vkontakte social network." *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 10, no. 2, pp. 55–75, 2019. 15

[31] D. Verdejo, "A review of single-source graph search and scoring algorithms." Mar 2023, cT5144 - Research Skills in Artificial Intelligence. 15

[32] T. Grainger, K. AlJadda, M. Korayem, and A. Smith, "The semantic knowledge graph: A compact, auto-generated model for real-time traversal and ranking of any relationship within a domain," in *2016 ieee international conference on data science and advanced analytics (dsaa)*. IEEE, 2016, pp. 420–429. 16

[33] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016. [Online]. Available: https://arxiv.org/abs/1607.00653 16

[34] X. Wang, J. Lin, C. Ren, and J. Chen, "Knowledge graph-based semantic ranking for efficient semantic query," in *2022 IEEE 10th International Conference on Computer Science and Network Technology (ICCSNT)*, 2022, pp. 75–79. 17

[35] D. Shu, T. Chen, M. Jin, C. Zhang, M. Du, and Y. Zhang, "Knowledge

graph large language model (kg-llm) for link prediction," 2024. [Online]. Available: https://arxiv.org/abs/2403.07311 17

[36] C. P. Kellogg G and L. D, "Json-ld 1.1 – a json-based serialization for linked data," *Technical Report W3C.*, 2020. [Online]. Available: https://www.w3.org/TR/2020/REC-json-ld11-20200716/ 18

[37] H. O., dataset used throughout experiments. [Online]. Available: https://www.kaggle.com/datasets/omarhanyy/imdb-top-1000 19

[38] S. Sun, S. Zhuang, S. Wang, and G. Zuccon, "An investigation of prompt variations for zero-shot llm-based rankers," 2024. [Online]. Available: https://arxiv.org/abs/2406.14117 40

[39] LangChain, "Prompttemplate," 2024, visited: 4th Jul 2024. [Online]. Available: https://api.python.langchain.com/en/latest/prompts/langchain_core.prompts.prompt.PromptTemplate.html 42

[40] OpenAI, "New embedding models and api updates," January 2024, visited: 6th Jul 2024. [Online]. Available: https://openai.com/index/new-and-improved-embedding-model/?tlaAppCB= 52

[41] Minitab, LLC, "Minitab," 2024, accessed: 2024-08-11. [Online]. Available: https://www.minitab.com/en-us/ 71

[42] L. Zhang and W. Tu, "Six degrees of separation in online society," 2009. 76

Appendix A

Appendix

A.1 Listings

Listing A.1: MOVIE-GRAPH-JSON

```
{
  "@context": {
    "entity": "https://schema.org/Thing"
  },
  "@graph": [
    {
      "@id": "30a2b51d-27f4-4cbd-bf7b-c7b72479d244",
      "@type": "Movie",
      "label": "The Lord of the Rings: The Return of the King",
      "Released_Year": "2003",
      "Certificate": "U",
      "Runtime": "201 min",
      "Genre": "Action, Adventure, Drama",
```

```
    "IMDB_Rating": 8.9,
    "Overview": "Gandalf and Aragorn lead the World of Men
       against Sauron's army to draw his gaze from Frodo and
       Sam as they approach Mount Doom with the One Ring.",
    "Meta_score": 94,
    "Director": "Peter Jackson",
    "Cast": [
      "Elijah Wood",
      "Viggo Mortensen",
      "Ian McKellen",
      "Orlando Bloom"
    ]
  },
  {
    "@id": "32e683df-c89d-41bf-a4b7-c854033b22c0",
    "@type": "Actor",
    "label": "Elijah Wood",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers"
    ],
    "co-starred": [
      {
        "@id": "384521da-fd61-473f-b8c4-b3172a144b81"
      }
    ]
```

```
  },
  {
    "@id": "bbf679b0-ff13-44d1-95c2-abaac79cefae",
    "@type": "Actor",
    "label": "Viggo Mortensen",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Two Towers",
      "Green Book",
      "Captain Fantastic",
      "Eastern Promises"
    ],
    "co-starred": [
      {
        "@id": "384521da-fd61-473f-b8c4-b3172a144b81"
      }
    ]
  },
  {
    "@id": "d1ae107c-fbf8-4d22-8039-3ca0fd5e25c7",
    "@type": "Actor",
    "label": "Ian McKellen",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
      "X-Men: Days of Future Past",
```

```
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey",
      "Stardust"
    ],
    "co-starred": [
      {
        "@id": "cb1637eb-4b6a-4c41-ae98-1c19bf058a47"
      }
    ]
  },
  {
    "@id": "384521da-fd61-473f-b8c4-b3172a144b81",
    "@type": "Actor",
    "label": "Orlando Bloom",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
      "Pirates of the Caribbean: The Curse of the Black Pearl
         "
    ],
    "co-starred": [
      {
        "@id": "0186eb7a-ce10-4724-86c3-b929e1d11b06"
      }
    ]
  },
```

```
{
  "@id": "474996da-1982-40a6-9175-329324bc5bad",
  "@type": "Director",
  "label": "Peter Jackson",
  "directed": [
    "The Lord of the Rings: The Return of the King",
    "The Lord of the Rings: The Fellowship of the Ring",
    "The Lord of the Rings: The Two Towers",
    "The Hobbit: The Desolation of Smaug",
    "The Hobbit: An Unexpected Journey"
  ]
},
{
  "@id": "1946b274-fd20-4e44-b1c2-a099f61631cc",
  "@type": "Movie",
  "label": "The Lord of the Rings: The Fellowship of the
     Ring",
  "Released_Year": "2001",
  "Certificate": "U",
  "Runtime": "178 min",
  "Genre": "Action, Adventure, Drama",
  "IMDB_Rating": 8.8,
  "Overview": "A meek Hobbit from the Shire and eight
     companions set out on a journey to destroy the
     powerful One Ring and save Middle-earth from the Dark
     Lord Sauron.",
  "Meta_score": 92,
```

```
    "Director": "Peter Jackson",
    "Cast": [
      "Elijah Wood",
      "Ian McKellen",
      "Orlando Bloom",
      "Sean Bean"
    ]
  },
  {
    "@id": "64a8f336-3d2a-4bcc-aec9-6685982365cf",
    "@type": "Actor",
    "label": "Sean Bean",
    "starredIn": [
      "The Lord of the Rings: The Fellowship of the Ring"
    ],
    "co-starred": [
      {
        "@id": "384521da-fd61-473f-b8c4-b3172a144b81"
      }
    ]
  },
  {
    "@id": "646a1bc2-18de-4e40-84b3-1b34c16ce688",
    "@type": "Movie",
    "label": "The Lord of the Rings: The Two Towers",
    "Released_Year": "2002",
    "Certificate": "UA",
```

```
    "Runtime": "179 min",
    "Genre": "Action, Adventure, Drama",
    "IMDB_Rating": 8.7,
    "Overview": "While Frodo and Sam edge closer to Mordor
       with the help of the shifty Gollum, the divided
       fellowship makes a stand against Sauron's new ally,
       Saruman, and his hordes of Isengard.",
    "Meta_score": 87,
    "Director": "Peter Jackson",
    "Cast": [
      "Elijah Wood",
      "Ian McKellen",
      "Viggo Mortensen",
      "Orlando Bloom"
    ]
  },
  {
    "@id": "35d69c0c-5f62-474e-a80b-e71ce67feeb3",
    "@type": "Movie",
    "label": "The Hobbit: The Desolation of Smaug",
    "Released_Year": "2013",
    "Certificate": "UA",
    "Runtime": "161 min",
    "Genre": "Adventure, Fantasy",
    "IMDB_Rating": 7.8,
    "Overview": "The dwarves, along with Bilbo Baggins and
       Gandalf the Grey, continue their quest to reclaim
```

```
      Erebor, their homeland, from Smaug. Bilbo Baggins is
      in possession of a mysterious and magical ring.",
    "Meta_score": 66,
    "Director": "Peter Jackson",
    "Cast": [
      "Ian McKellen",
      "Martin Freeman",
      "Richard Armitage",
      "Ken Stott"
    ]
  },
  {
    "@id": "42dcc9fc-1862-45d2-8d71-138d4dc23d2e",
    "@type": "Actor",
    "label": "Martin Freeman",
    "starredIn": [
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey",
      "Hot Fuzz"
    ],
    "co-starred": [
      {
        "@id": "cb1637eb-4b6a-4c41-ae98-1c19bf058a47"
      }
    ]
  },
  {
```

```
    "@id": "43f96a08-135c-479e-948f-a0d9eb0a5931",
    "@type": "Actor",
    "label": "Richard Armitage",
    "starredIn": [
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey"
    ],
    "co-starred": [
      {
        "@id": "cb1637eb-4b6a-4c41-ae98-1c19bf058a47"
      }
    ]
  },
  {
    "@id": "a205f32c-cf5f-42d6-9b8c-bdc5d3b01ead",
    "@type": "Actor",
    "label": "Ken Stott",
    "starredIn": [
      "The Hobbit: The Desolation of Smaug"
    ],
    "co-starred": [
      {
        "@id": "43f96a08-135c-479e-948f-a0d9eb0a5931"
      }
    ]
  },
  {
```

```
    "@id": "035bbdbb-95b6-428c-bbe1-3275eb618ae1",
    "@type": "Movie",
    "label": "The Hobbit: An Unexpected Journey",
    "Released_Year": "2012",
    "Certificate": "UA",
    "Runtime": "169 min",
    "Genre": "Adventure, Fantasy",
    "IMDB_Rating": 7.8,
    "Overview": "A reluctant Hobbit, Bilbo Baggins, sets out
       to the Lonely Mountain with a spirited group of
       dwarves to reclaim their mountain home, and the gold
       within it from the dragon Smaug.",
    "Meta_score": 58,
    "Director": "Peter Jackson",
    "Cast": [
      "Martin Freeman",
      "Ian McKellen",
      "Richard Armitage",
      "Andy Serkis"
    ]
  },
  {
    "@id": "cb1637eb-4b6a-4c41-ae98-1c19bf058a47",
    "@type": "Actor",
    "label": "Andy Serkis",
    "starredIn": [
      "The Hobbit: An Unexpected Journey",
```

```
      "Dawn of the Planet of the Apes"
    ],
    "co-starred": [
      {
        "@id": "43f96a08-135c-479e-948f-a0d9eb0a5931"
      }
    ]
  },
  {
    "@id": "238471cc-3fa8-4b74-8ea5-9b2b890dbc84",
    "@type": "Movie",
    "label": "Pirates of the Caribbean: The Curse of the
       Black Pearl",
    "Released_Year": "2003",
    "Certificate": "UA",
    "Runtime": "143 min",
    "Genre": "Action, Adventure, Fantasy",
    "IMDB_Rating": 8.0,
    "Overview": "Blacksmith Will Turner teams up with
       eccentric pirate \"Captain\" Jack Sparrow to save his
       love, the governor's daughter, from Jack's former
       pirate allies, who are now undead.",
    "Meta_score": 63,
    "Director": "Gore Verbinski",
    "Cast": [
      "Johnny Depp",
      "Geoffrey Rush",
```

```
      "Orlando Bloom",
      "Keira Knightley"
    ]
  },
  {
    "@id": "0186eb7a-ce10-4724-86c3-b929e1d11b06",
    "@type": "Actor",
    "label": "Keira Knightley",
    "starredIn": [
      "The Imitation Game",
      "Pirates of the Caribbean: The Curse of the Black Pearl
         ",
      "Atonement",
      "Pride & Prejudice"
    ],
    "co-starred": [
      {
        "@id": "384521da-fd61-473f-b8c4-b3172a144b81"
      }
    ]
  },
  {
    "@id": "a1cd48b5-e1ff-490a-91b4-37cf95b1f3cb",
    "@type": "Actor",
    "label": "Geoffrey Rush",
    "starredIn": [
      "The King’s Speech",
```

```
      "Pirates of the Caribbean: The Curse of the Black Pearl
         ",
      "La migliore offerta",
      "Shine"
    ],
    "co-starred": [
      {
        "@id": "0186eb7a-ce10-4724-86c3-b929e1d11b06"
      }
    ]
  },
  {
    "@id": "2c478fb9-afb3-4b00-9198-0ef141212057",
    "@type": "Actor",
    "label": "Johnny Depp",
    "starredIn": [
      "Pirates of the Caribbean: The Curse of the Black Pearl
         ",
      "Edward Scissorhands",
      "Ed Wood",
      "What’s Eating Gilbert Grape",
      "Donnie Brasco",
      "Finding Neverland",
      "Blow",
      "Fear and Loathing in Las Vegas",
      "Dead Man"
    ],
```

```
      "co-starred": [
        {
          "@id": "0186eb7a-ce10-4724-86c3-b929e1d11b06"
        }
      ]
    },
    {
      "@id": "ebaf675d-f388-401a-8c06-50f9ff041b32",
      "@type": "Director",
      "label": "Gore Verbinski",
      "directed": [
        "Pirates of the Caribbean: The Curse of the Black Pearl
           "
      ]
    }
  ]
}
```

Listing A.2: graph-a

```
{
  "@context": {
    "Movie": "https://schema.org/Movie",
    "Actor": "https://schema.org/Actor",
    "Director": "https://schema.org/Director"
  },
  "@graph": [
    {
```

```
    "@id": "30a2b51d-27f4-4cbd-bf7b-c7b72479d244",
    "@type": "Movie",
    "label": "The Lord of the Rings: The Return of the King",
    "Released_Year": "2003",
    "Certificate": "U",
    "Runtime": "201 min",
    "Genre": "Action, Adventure, Drama",
    "IMDB_Rating": 8.9,
    "Overview": "Gandalf and Aragorn lead the World of Men
       against Sauron’s army to draw his gaze from Frodo and
       Sam as they approach Mount Doom with the One Ring.",
    "Meta_score": 94,
    "Director": "Peter Jackson",
    "Cast": [
      "Elijah Wood",
      "Viggo Mortensen",
      "Ian McKellen",
      "Orlando Bloom"
    ]
  },
  {
    "@id": "32e683df-c89d-41bf-a4b7-c854033b22c0",
    "@type": "Actor",
    "label": "Elijah Wood",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
```

```
      "The Lord of the Rings: The Two Towers"
    ]
  },
  {
    "@id": "bbf679b0-ff13-44d1-95c2-abaac79cefae",
    "@type": "Actor",
    "label": "Viggo Mortensen",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Two Towers",
      "Green Book",
      "Captain Fantastic",
      "Eastern Promises"
    ]
  },
  {
    "@id": "d1ae107c-fbf8-4d22-8039-3ca0fd5e25c7",
    "@type": "Actor",
    "label": "Ian McKellen",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
      "X-Men: Days of Future Past",
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey",
      "Stardust"
```

```
    ]
  },
  {
    "@id": "384521da-fd61-473f-b8c4-b3172a144b81",
    "@type": "Actor",
    "label": "Orlando Bloom",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
      "Pirates of the Caribbean: The Curse of the Black Pearl
         "
    ]
  },
  {
    "@id": "474996da-1982-40a6-9175-329324bc5bad",
    "@type": "Director",
    "label": "Peter Jackson",
    "directed": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey"
    ]
  }
]
```

```
}
```

Listing A.3: graph-b

```
{
  "@context": {
    "Movie": "https://schema.org/Movie",
    "Actor": "https://schema.org/Actor",
    "Director": "https://schema.org/Director"
  },
  "@graph": [
    {
      "@id": "35d69c0c-5f62-474e-a80b-e71ce67feeb3",
      "@type": "Movie",
      "label": "The Hobbit: The Desolation of Smaug",
      "Released_Year": "2013",
      "Certificate": "UA",
      "Runtime": "161 min",
      "Genre": "Adventure, Fantasy",
      "IMDB_Rating": 7.8,
      "Overview": "The dwarves, along with Bilbo Baggins and
         Gandalf the Grey, continue their quest to reclaim
         Erebor, their homeland, from Smaug. Bilbo Baggins is
         in possession of a mysterious and magical ring.",
      "Meta_score": 66,
      "Director": "Peter Jackson",
      "Cast": [
        "Ian McKellen",
```

```
      "Martin Freeman",
      "Richard Armitage",
      "Ken Stott"
    ]
  },
  {
    "@id": "d1ae107c-fbf8-4d22-8039-3ca0fd5e25c7",
    "@type": "Actor",
    "label": "Ian McKellen",
    "starredIn": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
      "X-Men: Days of Future Past",
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey",
      "Stardust"
    ]
  },
  {
    "@id": "42dcc9fc-1862-45d2-8d71-138d4dc23d2e",
    "@type": "Actor",
    "label": "Martin Freeman",
    "starredIn": [
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey",
      "Hot Fuzz"
```

```
    ]
  },
  {
    "@id": "43f96a08-135c-479e-948f-a0d9eb0a5931",
    "@type": "Actor",
    "label": "Richard Armitage",
    "starredIn": [
      "The Hobbit: The Desolation of Smaug",
      "The Hobbit: An Unexpected Journey"
    ]
  },
  {
    "@id": "a205f32c-cf5f-42d6-9b8c-bdc5d3b01ead",
    "@type": "Actor",
    "label": "Ken Stott",
    "starredIn": [
      "The Hobbit: The Desolation of Smaug"
    ]
  },
  {
    "@id": "474996da-1982-40a6-9175-329324bc5bad",
    "@type": "Director",
    "label": "Peter Jackson",
    "directed": [
      "The Lord of the Rings: The Return of the King",
      "The Lord of the Rings: The Fellowship of the Ring",
      "The Lord of the Rings: The Two Towers",
```

```
        "The Hobbit: The Desolation of Smaug",
        "The Hobbit: An Unexpected Journey"
      ]
    }
  ]
}
```

A.2 Tables

model	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG
GPT-4o	8	8	9	8	7	7	7	6	7.5
GPT-4-t	7	8	9	6	9	8	7	7	7.625
GPT-3.5-t	8	9	8	7	9	8	8	7	8
Gemma	6	7	7	8	7	7	7	7	7
Llama3	6	8	8	8	8	9	8	8	7.875
Mistral	7	8	8	9	9	10	8	7	8.25
Phi3-med	8	7	8	7	8	7	7	8	7.5
Phi3-mini	7	7	7	9	9	7	8	7	7.625

Table A.1: Group A - Reasoning scores

model	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG
GPT-4o	8	8	8	7	6	7	6	6	7
GPT-4-t	8	8	10	7	9	9	7	7	8.125
GPT-3.5-t	7	8	9	8	9	9	8	8	8.25
Gemma	7	8	6	7	9	8	8	8	7.625
Llama3	8	9	8	8	8	10	7	9	8.375
Mistral	10	7	7	8	10	10	10	8	8.75
Phi3-med	7	8	9	8	7	8	9	9	8.125
Phi3-mini	8	8	8	8	8	9	7	8	8

Table A.2: Group A - Accuracy scores

A.3 Figures

A.3.1 Minitab boxplots and histograms

model	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG
GPT-4o	8	8	9	8	8	8	7	7	7.875
GPT-4-t	8	9	10	7	9	9	8	8	8.5
GPT-3.5-t	8	8	9	8	9	9	9	8	8.5
Gemma	8	8	8	8	9	9	9	9	8.5
Llama3	9	10	10	10	8	10	9	10	9.5
Mistral	8	6	9	9	10	10	10	9	8.875
Phi3-med	10	9	10	9	9	9	10	7	9.125
Phi3-mini	10	6	9	9	10	8	9	9	8.75

Table A.3: Group A - Factual scores

name	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG	MAX	MIN
GPT-4o	8	9	9	7	9	8	8	9	8.375	9	7
GPT-4-t	8	8	8	7	8	7	7	7	7.5	8	7
GPT-3.5-t	7	7	8	8	8	9	8	7	7.75	9	7
Gemma	6	6	8	8	9	8	8	8	7.625	9	6
Llama3	7	8	9	8	9	10	9	9	8.625	10	7
Mistral	8	6	7	8	9	8	8	7	7.625	9	6
Phi3-med	7	7	7	8	9	8	9	7	7.75	9	7
Phi3-mini	7	7	7	9	9	8	7	7	7.625	9	7
AVG	7.25	7.25	7.875	7.875	8.75	8.25	8	7.625	7.859375	8.75	7.25
MAX	8	9	9	9	9	10	9	9	8.625	10	8
MIN	6	6	7	7	8	7	7	7	7.5	8	6

Table A.4: Group B - Reasoning scores

Name	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG	MAX	MIN
GPT-4o	7	9	8	8	8	9	9	8	8.25	9	7
GPT-4-t	8	8	7	8	9	8	9	9	8.25	9	7
GPT-3.5-t	8	7	8	9	8	10	9	8	8.375	10	7
Gemma	7	5	8	9	8	9	9	9	8	9	5
Llama3	6	8	9	9	10	10	8	8	8.5	10	6
Mistral	9	6	8	9	8	7	9	8	8	9	6
Phi3-med	6	6	8	7	8.5	9	8	9	7.6875	9	6
Phi3-mini	7	7	8	8	8	9	9	8	8	9	7
AVG	7.25	7	8	8.375	8.4375	8.875	8.75	8.375	8.1328125	8.875	7
MAX	9	9	9	9	10	10	9	9	8.5	10	9
MIN	6	5	7	7	8	7	8	8	7.6875	8	5

Table A.5: Group B - Accuracy scores

NAME	GPT-4o	GPT-4-t	GPT-3.5-t	Gemma	Llama3	Mistral	Phi3-med	Phi3-mini	AVG	MAX	MIN
GPT-4o	9	9	9	9	10	10	9	7	9	10	7
GPT-4-t	8	7	8	8	9	10	10	8	8.5	10	7
GPT-3.5-t	8	7	8	9	10	10	10	9	8.875	10	7
Gemma	8	5	8	9	9	10	9	7	8.125	10	5
Llama3	8	8	9	9	10	10	9	9	9	10	8
Mistral	9	5	9	9	10	9	8	9	8.5	10	5
Phi3-med	7	7	9	8	10	10	10	8	8.625	10	7
Phi3-mini	8	8	9	9	9	9	8	9	8.625	9	8
AVG	8.125	7	8.625	8.75	9.625	9.75	9.125	8.25	8.65625	9.75	7
MAX	9	9	9	9	10	10	10	9	9	10	9
MIN	7	5	8	8	9	9	8	7	8.125	9	5

Table A.6: Group B - Factual scores

MODEL	Judges Panel of Scores - *Group B - Group A*								Review				
NAME	gpt-4o	gpt-4-t	gpt-3.5-t	gemma	llama3	mistral	phi3-med	phi3-mini	improved	same	worse	Equation	Sum
gpt-4o	-1	3	0	1	6	5	6	5	6	1	1	6+1-1	6
gpt-4-t	1	-2	-6	3	-1	-1	4	2	4	0	4	4+0-4	0
gpt-3.5-t	0	-4	-2	3	0	3	1	1	4	2	2	4+2-2	4
gemma	0	-7	3	3	-1	3	2	0	4	2	2	4+2-2	4
llama3	-3	-3	1	-1	3	1	2	-1	4	0	4	4+0-4	0
mistral	1	-4	0	0	-2	-6	-3	0	1	3	4	1+3-4	0
phi3-med	-6	-4	-3	-1	3.5	3	1	0	3	1	4	3+1-4	0
phi3-mini	-3	1	0	0	-1	3	0	0	2	4	2	2+4-2	4

Figure A.1: Final results table

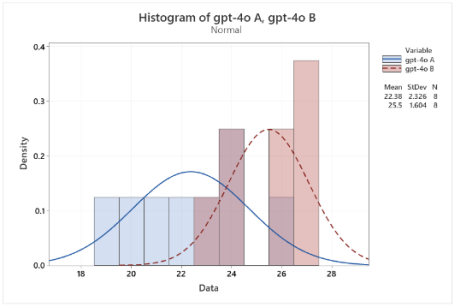
gpt-4o		
	Upward mean shift for Group B, spread of data smaller with higher values, with one whisker overlapping Group A.	Upward shift for Group B, spread of Bell curve minimized and output values are higher. Improvement observed.

Figure A.2: Boxplot and historgram - GPT-4o

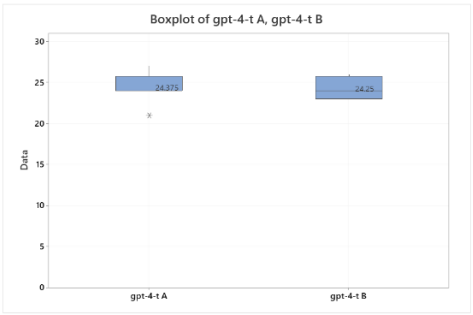
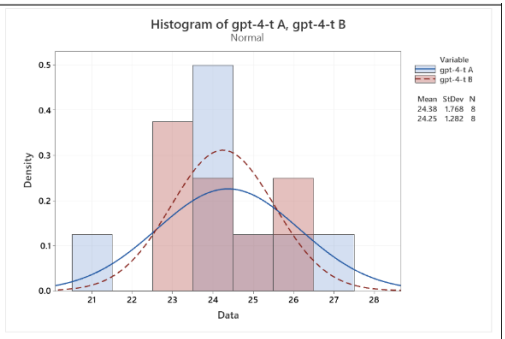
gpt-4-t		
	Downward mean shift for Group B (-0.125) Group B is somewhat centred however most of the data plotted is in the lower portion of the box. Group A Outlier distributes the data spread wider, beyond Group B results.	Both groups sit somewhat central under Bell Curves. A small shift downward for Group B as it is marginally left skewed, however curve spread is minimized. No specific improvement, decline in performance.

Figure A.3: Boxplot and histogram - GPT-4-turbo

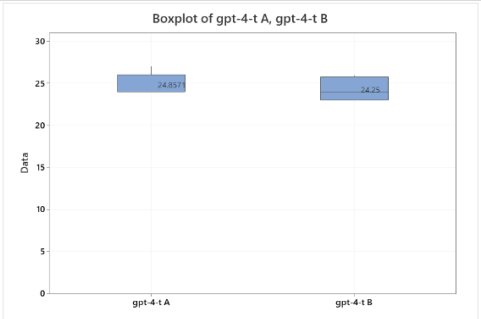
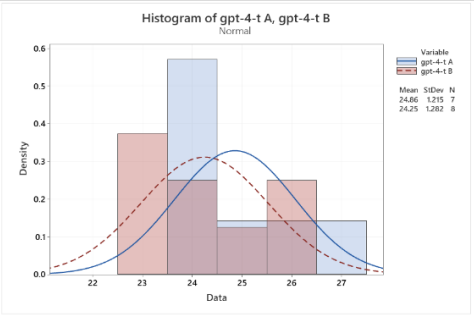
gpt-4-t		
	Still a downward mean shift for Group B post removal of outliers and spread of data still consistent.	A greater shift downward for Group B is observed, left skewed but with a similar data spread. Removal of outlier did not improve results. No specific improvement with indication of poorer performance.

Figure A.4: Boxplot and histogram - GPT-4-turbo (outliers removed)

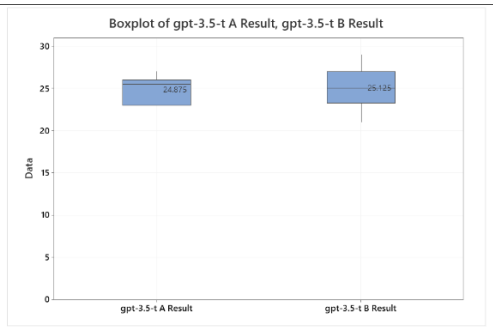
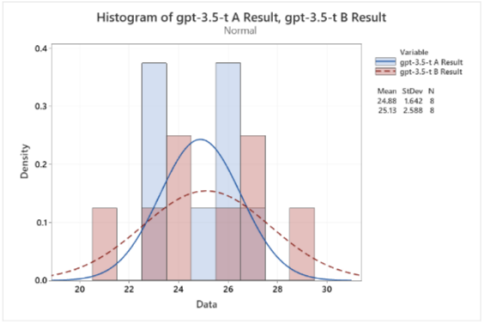
gpt-3.5-t		
	Upward mean shift for Group B (0.25), spread of data larger with higher and lower values observed.	Both groups sit somewhat central under Bell Curves. A small shift upward for Group B as it is marginally right skewed and greater curve spread. Small improvement but less reliable.

Figure A.5: Boxplot and histogram - GPT-3.5-turbo

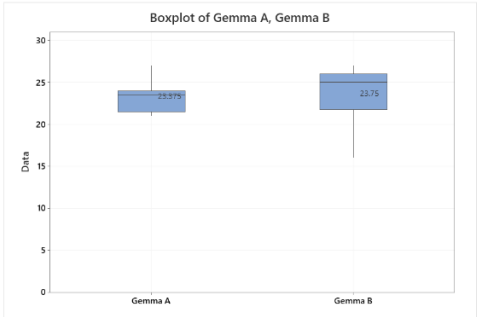
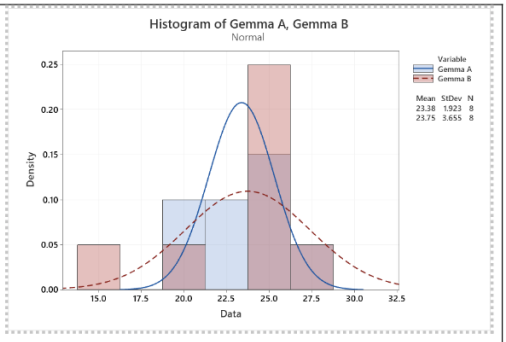
gemma		
	Upward mean shift for Group B (0.375), spread of data larger with lower values observed.	Both groups sit somewhat central under Bell Curves. A small shift upward for Group B as it is marginally right skewed and greater curve spread. Small improvement but less reliable.

Figure A.6: Boxplot and histogram - Gemma

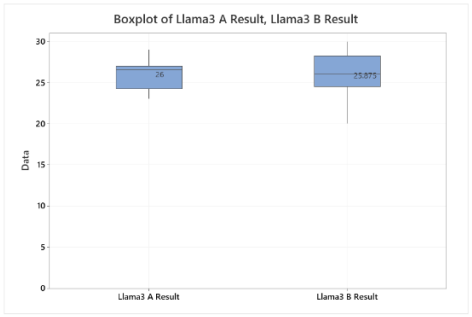
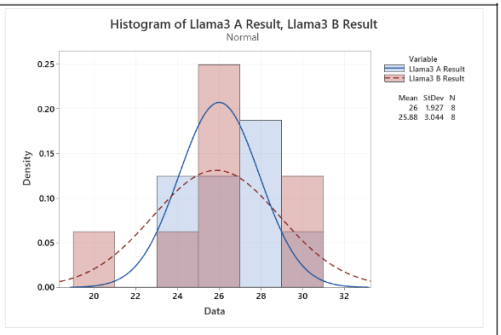
llama3		
	Downward mean shift for Group B (-0.125), spread of data smaller overall with tighter spread.	Both groups sit somewhat central under Bell Curves. A small shift downward for Group B as it is marginally left skewed and greater curve spread. No specific improvement with indication of poorer performance.

Figure A.7: Boxplot and histogram - Llama3

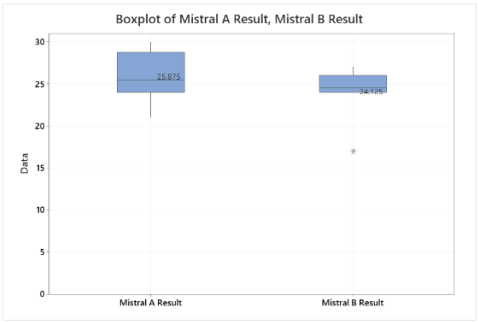
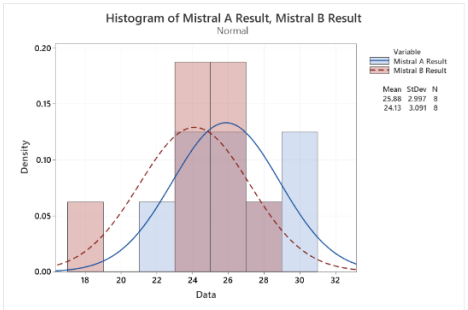
mistral		
	Downward mean shift for Group B (-1.75), spread of data larger with an Outlier present. Group A data somewhat centred within box. Group B Outlier distributes the data spread wider, beyond Group A results.	A shift downward for Group B as it is left skewed but with overall similar curve spread. No specific improvement with indication of decline in performance.

Figure A.8: Boxplot and histogram - Mistral

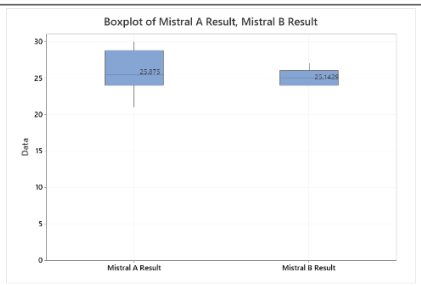
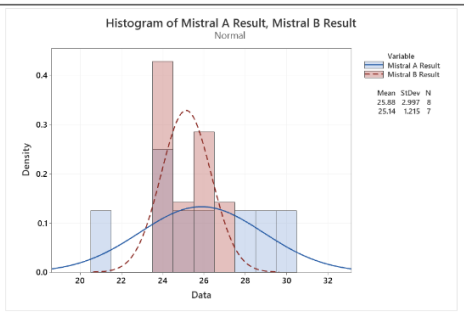
Mistral		
	Downward mean shift for Group B (-0.73), and improved data spread with outlier removal. Group A data somewhat centred within box. Group B centred.	A greater shift downward for Group B is observed, left skewed but improvement made post outlier removal. No specific improvement with indication of declined performance.

Figure A.9: Boxplot and histogram - Mistral (outliers removed)

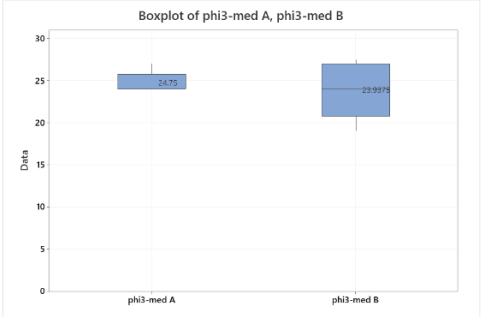
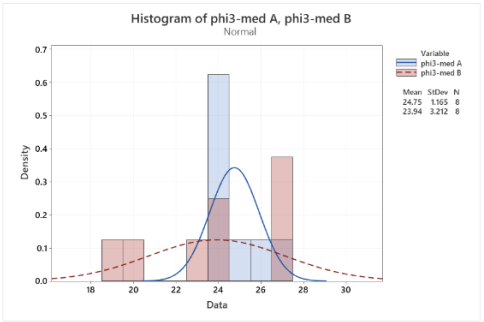
phi3-med		
	Downward mean shift for Group B (-0.8125), spread of data larger with higher and lower values observed.	A shift downward for Group B - left skewed, Spread of data increased significantly. Decline in performance apparent.

Figure A.10: Boxplot and histogram - Phi3-medium

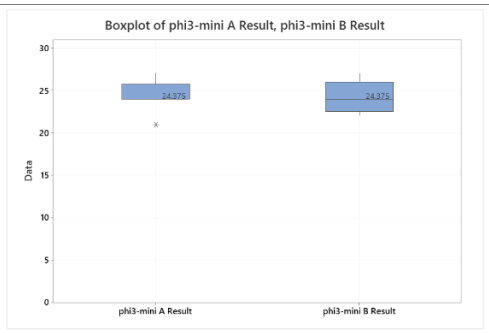
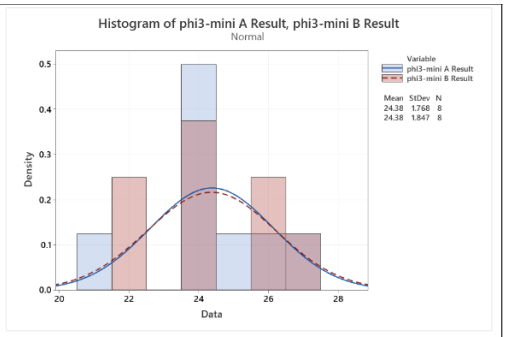
phi3-mini		
	Group B is somewhat centred within the box. Group A Outlier distributes the data spread wider, but comparable to Group B.	Visually Group A and Group B are almost identical. No specific Change.

Figure A.11: Boxplot and histogram - Phi3-mini

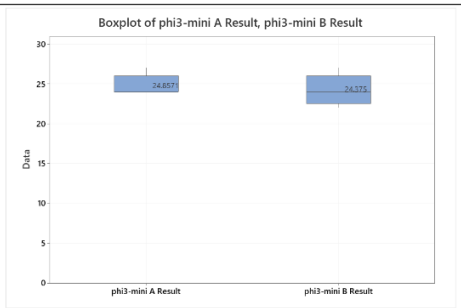
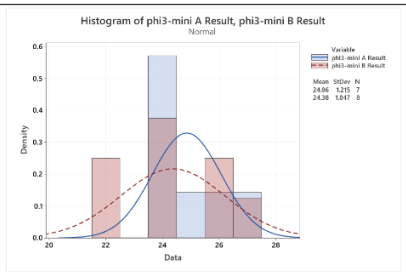
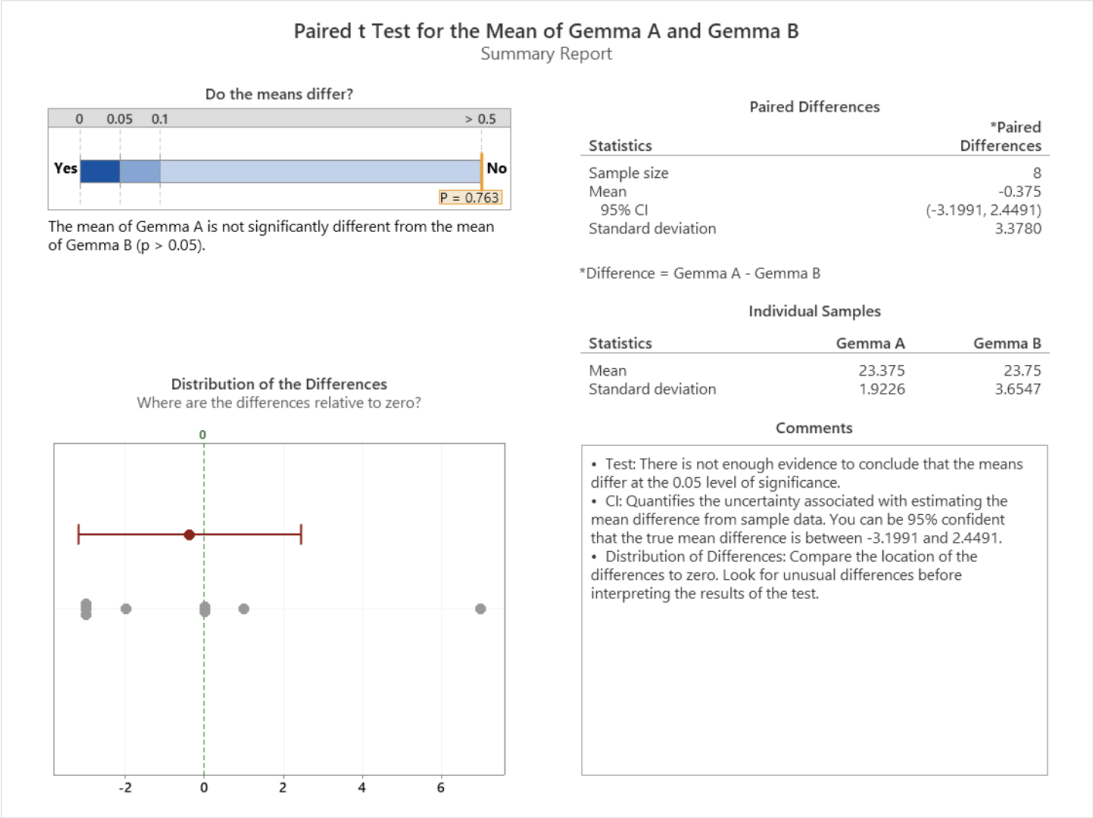
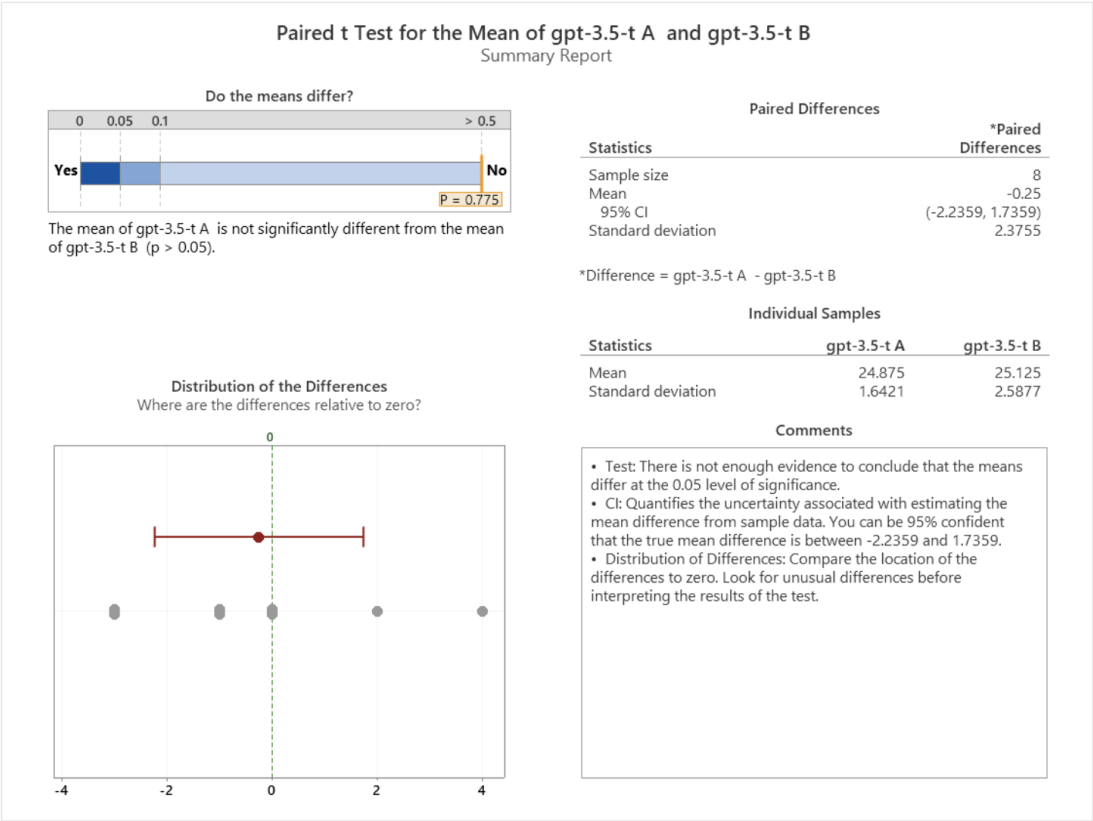
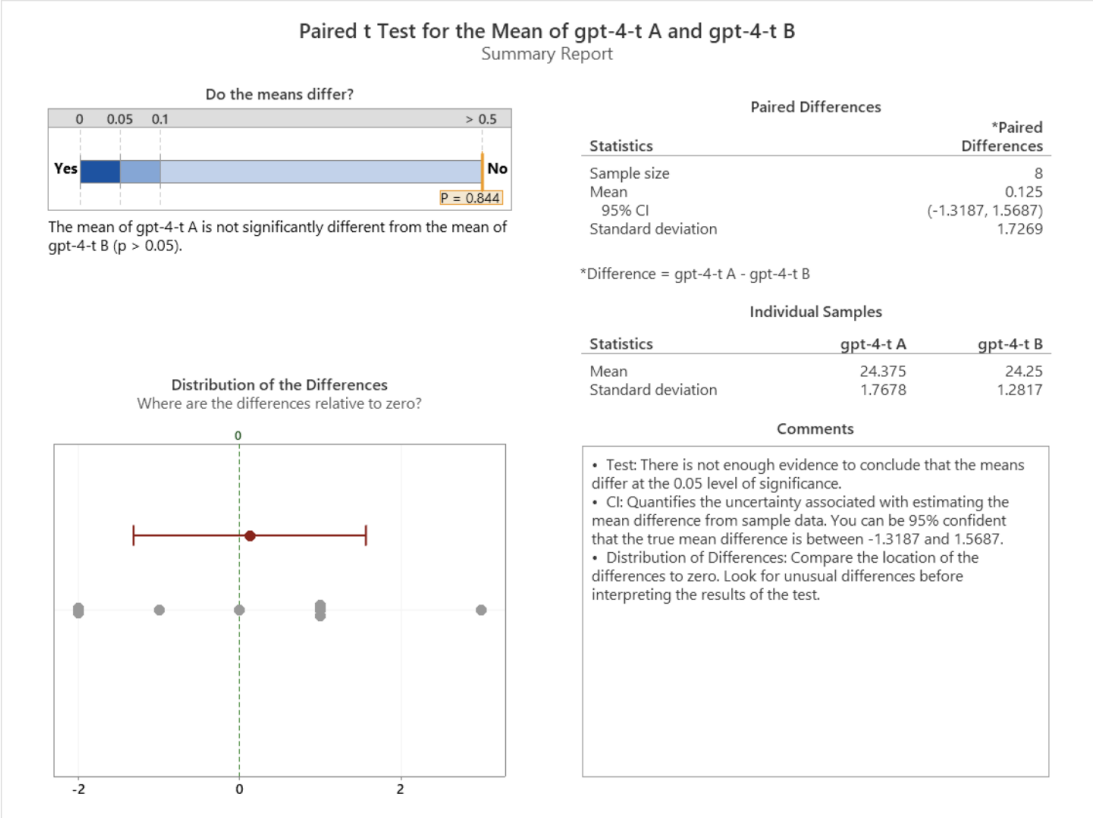
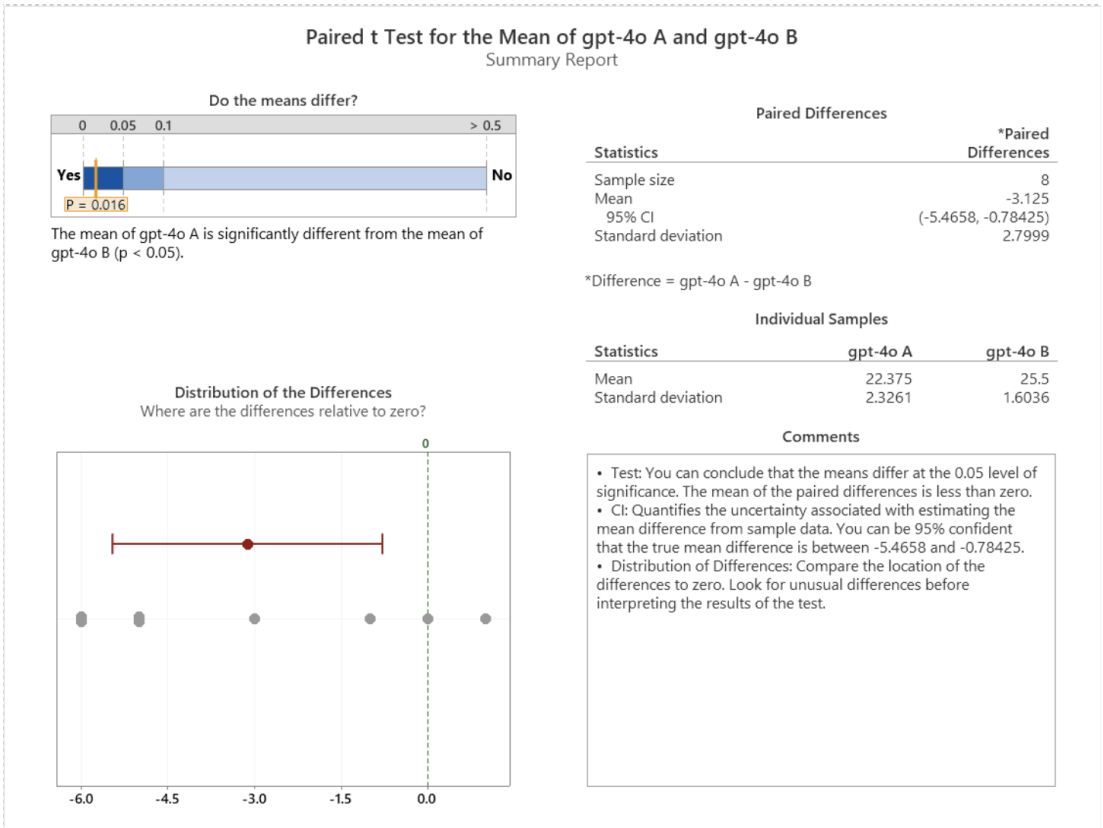
phi3-mini		
	Group B is somewhat centred within the box. Group A data sitting in upper section of the box.	Both groups sit somewhat central under Bell Curves. A small shift downward for Group B as it is marginally left skewed and greater data spread. No specific improvement and less reliable.

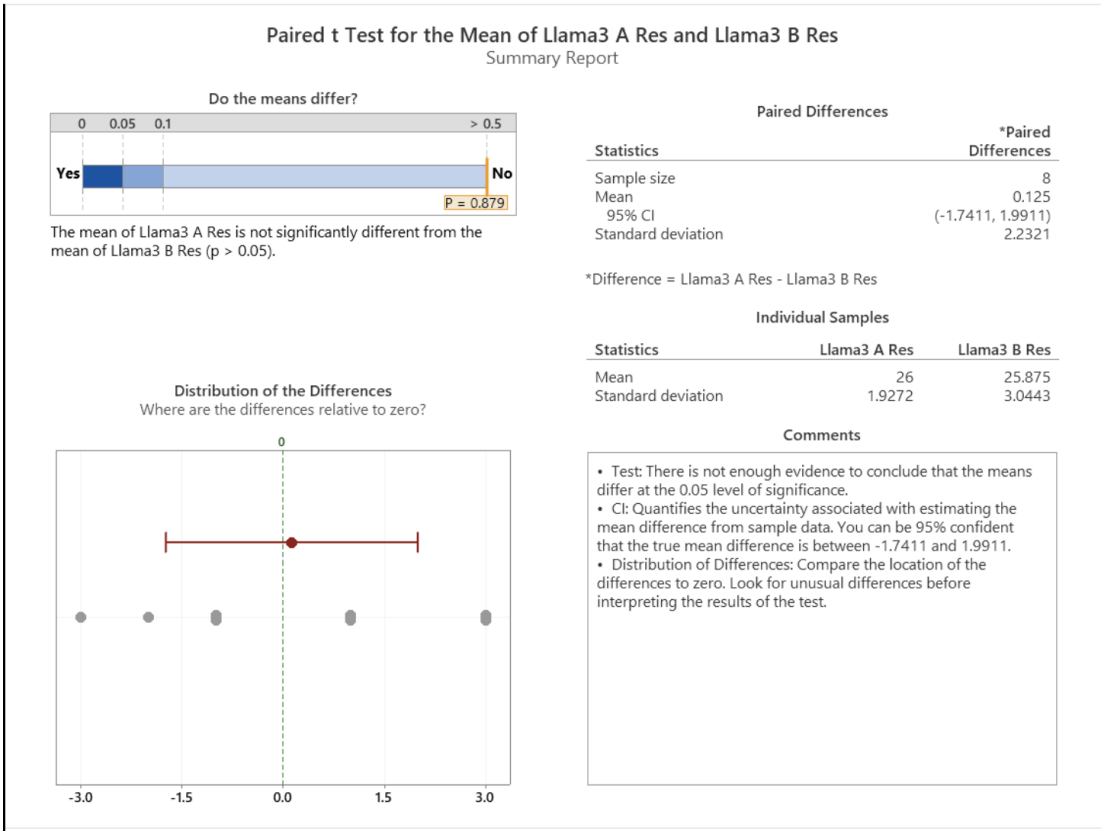
Figure A.12: Boxplot and histogram - Phi3-mini (outliers removed)


Paired t Test for the Mean of Gemma A and Gemma B
Summary Report
Do the means differ?
0
0.05
0.1
> 0.5
Yes
No
P = 0.763
The mean of Gemma A is not significantly different from the mean of Gemma B (p > 0.05).
Paired Differences
Statistics
*Paired Differences
Sample size 8
Mean -0.375
95% CI (-3.1991, 2.4491)
Standard deviation 3.3780
*Difference = Gemma A - Gemma B
Individual Samples
Statistics
Gemma A
Gemma B
Mean 23.375 23.75
Standard deviation 1.9226 3.6547
Distribution of the Differences
Where are the differences relative to zero?
0
-2
0
2
4
6
Comments
• Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
• CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -3.1991 and 2.4491.
• Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.


Paired t Test for the Mean of gpt-3.5-t A and gpt-3.5-t B
Summary Report
Do the means differ?
0 0.05 0.1 > 0.5
Yes
No
P = 0.775
The mean of gpt-3.5-t A is not significantly different from the mean of gpt-3.5-t B (p > 0.05).
Paired Differences
Statistics | *Paired Differences
Sample size | 8
Mean | -0.25
95% CI | (-2.2359, 1.7359)
Standard deviation | 2.3755
*Difference = gpt-3.5-t A - gpt-3.5-t B
Individual Samples
Statistics | gpt-3.5-t A | gpt-3.5-t B
Mean | 24.875 | 25.125
Standard deviation | 1.6421 | 2.5877
Distribution of the Differences
Where are the differences relative to zero?
0
-4 -2 0 2 4
Comments
• Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
• CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -2.2359 and 1.7359.
• Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.


Paired t Test for the Mean of gpt-4-t A and gpt-4-t B
Summary Report
Do the means differ?
0
0.05
0.1
> 0.5
Yes
No
P = 0.844
The mean of gpt-4-t A is not significantly different from the mean of gpt-4-t B (p > 0.05).
Paired Differences
Statistics
*Paired Differences
Sample size 8
Mean 0.125
95% CI (-1.3187, 1.5687)
Standard deviation 1.7269
*Difference = gpt-4-t A - gpt-4-t B
Individual Samples
Statistics
gpt-4-t A
gpt-4-t B
Mean 24.375 24.25
Standard deviation 1.7678 1.2817
Distribution of the Differences
Where are the differences relative to zero?
0
-2
0
2
Comments
• Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
• CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -1.3187 and 1.5687.
• Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.

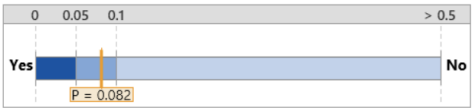

Paired t Test for the Mean of gpt-4o A and gpt-4o B
Summary Report
Do the means differ?
0 0.05 0.1 > 0.5
Yes
No
P = 0.016
The mean of gpt-4o A is significantly different from the mean of gpt-4o B (p < 0.05).
Paired Differences
Statistics
*Paired Differences
Sample size 8
Mean -3.125
95% CI (-5.4658, -0.78425)
Standard deviation 2.7999
*Difference = gpt-4o A - gpt-4o B
Individual Samples
Statistics
gpt-4o A
gpt-4o B
Mean 22.375 25.5
Standard deviation 2.3261 1.6036
Distribution of the Differences
Where are the differences relative to zero?
0
-6.0 -4.5 -3.0 -1.5 0.0
Comments
• Test: You can conclude that the means differ at the 0.05 level of significance. The mean of the paired differences is less than zero.
• CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -5.4658 and -0.78425.
• Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.


Paired t Test for the Mean of Llama3 A Res and Llama3 B Res
Summary Report
Do the means differ?
0 0.05 0.1 > 0.5
Yes
No
P = 0.879
The mean of Llama3 A Res is not significantly different from the mean of Llama3 B Res (p > 0.05).
Paired Differences
Statistics | *Paired Differences
Sample size | 8
Mean | 0.125
95% CI | (-1.7411, 1.9911)
Standard deviation | 2.2321
*Difference = Llama3 A Res - Llama3 B Res
Individual Samples
Statistics | Llama3 A Res | Llama3 B Res
Mean | 26 | 25.875
Standard deviation | 1.9272 | 3.0443
Distribution of the Differences
Where are the differences relative to zero?
0
-3.0 -1.5 0.0 1.5 3.0
Comments
• Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
• CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -1.7411 and 1.9911.
• Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.

Paired t Test for the Mean of Mistral A Re and Mistral B Re

Summary Report

Do the means differ?



The mean of Mistral A Re is not significantly different from the mean of Mistral B Re (p > 0.05).

Paired Differences

Statistics	*Paired Differences
Sample size	8
Mean	1.75
95% CI	(-0.28560, 3.7856)
Standard deviation	2.4349

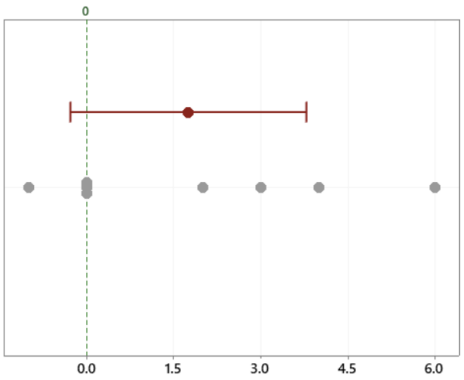
*Difference = Mistral A Re - Mistral B Re

Individual Samples

Statistics	Mistral A Re	Mistral B Re
Mean	25.875	24.125
Standard deviation	2.9970	3.0909

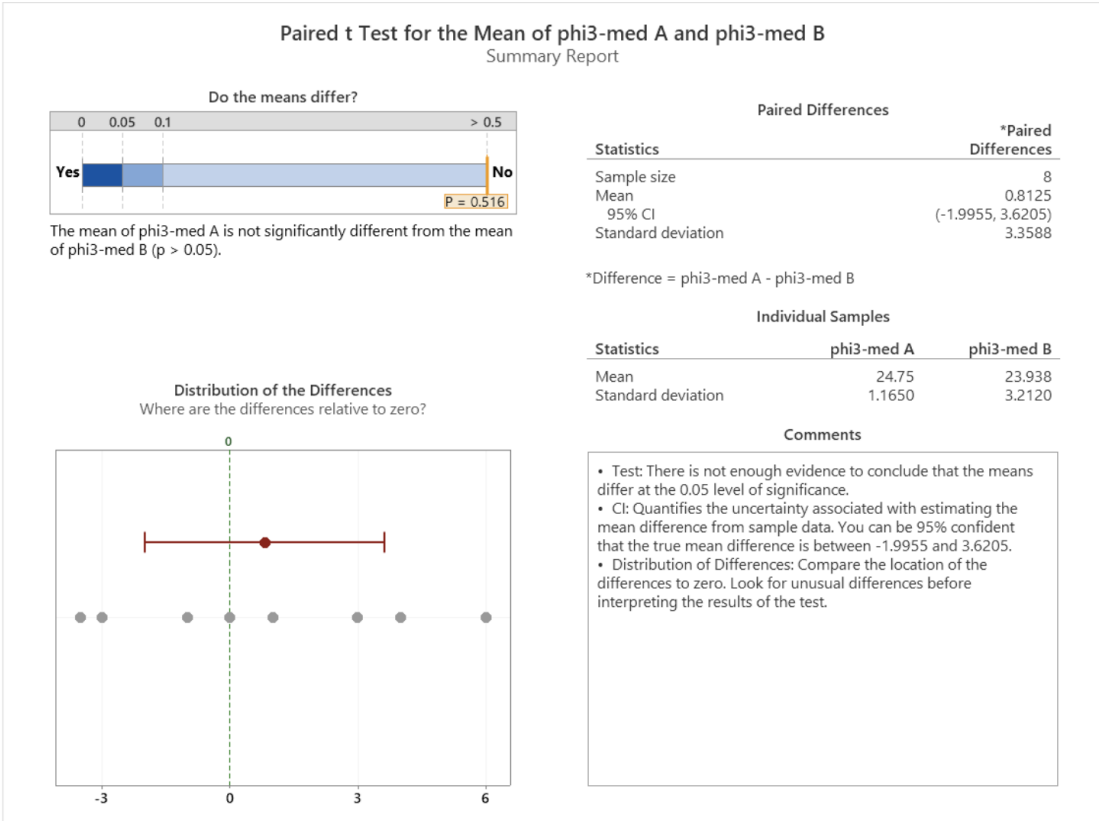
Distribution of the Differences

Where are the differences relative to zero?



Comments

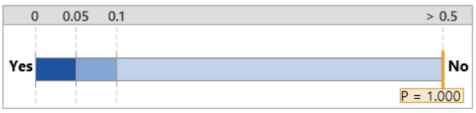
- Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
- CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -0.28560 and 3.7856.
- Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.


Paired t Test for the Mean of phi3-med A and phi3-med B
Summary Report
Do the means differ?
0
0.05
0.1
> 0.5
Yes
No
P = 0.516
The mean of phi3-med A is not significantly different from the mean of phi3-med B (p > 0.05).
Paired Differences
Statistics
*Paired Differences
Sample size 8
Mean 0.8125
95% CI (-1.9955, 3.6205)
Standard deviation 3.3588
*Difference = phi3-med A - phi3-med B
Individual Samples
Statistics
phi3-med A
phi3-med B
Mean 24.75 23.938
Standard deviation 1.1650 3.2120
Distribution of the Differences
Where are the differences relative to zero?
0
-3
0
3
6
Comments
• Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
• CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -1.9955 and 3.6205.
• Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.

Paired t Test for the Mean of phi3-mini A and phi3-mini B

Summary Report

Do the means differ?



The mean of phi3-mini A is not significantly different from the mean of phi3-mini B ($p > 0.05$).

Paired Differences

Statistics	*Paired Differences
Sample size	8
Mean	0
95% CI	(-1.4131, 1.4131)
Standard deviation	1.6903

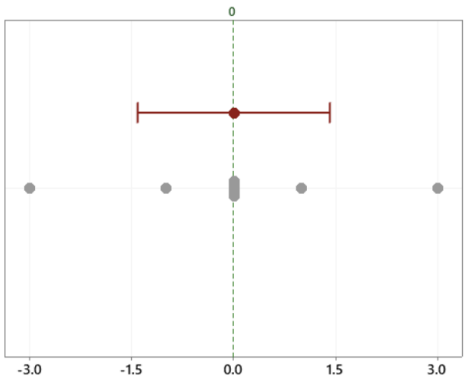
*Difference = phi3-mini A - phi3-mini B

Individual Samples

Statistics	phi3-mini A	phi3-mini B
Mean	24.375	24.375
Standard deviation	1.7678	1.8468

Distribution of the Differences

Where are the differences relative to zero?



Comments

- Test: There is not enough evidence to conclude that the means differ at the 0.05 level of significance.
- CI: Quantifies the uncertainty associated with estimating the mean difference from sample data. You can be 95% confident that the true mean difference is between -1.4131 and 1.4131.
- Distribution of Differences: Compare the location of the differences to zero. Look for unusual differences before interpreting the results of the test.