LLM-driven Summarization and Distinguish Analysis of Multiple Entities in RDF Graphs

Hamza Iqbal[0009−0001−5152−8193] and Kostas Stefanidis[0000−0003−1317−8062]

Tampere University, Finland
hamzaiqbal1012@gmail.com, konstantinos.stefanidis@tuni.fi

Abstract. This research implements the application of Large Language Models (LLMs) in the summarization and distinguish analysis of multiple entities within Resource Description Framework (RDF) graphs. As the volume of structured data on the web is growing exponentially, the need for efficient and effective methods to interpret and summarize this data becomes increasingly important. This study focuses on utilizing LLMs to generate human-readable summaries from RDF graphs and particularly emphasizing on distinguishing between multiple entities. The study apply SPARQL queries to extract relevant data from DBpedia, subsequently a thorough process of frequency analysis and property unification to refine the dataset. Three LLMs including ChatGPT, DeepSeek, and Mistral have been evaluated for their ability to generate coherent and informative summaries. The evaluation process combines human-based assessments with automated metrics for the thorough analysis of generated texts. Key outcomes include the effectiveness of LLMs in generating summaries that are both informative and contextually relevant. The research also reflects the importance of data preprocessing techniques, such as frequency analysis and property unification in enhancing the quality of the generated summaries. Moreover, the study provides insights into the strengths and limitations of different LLMs in summarizing RDF data that offers a foundation for future research in this area. A framework for evaluating the performance of LLMs in summarization tasks has been designed in this research opens the way for future explorations in the application of advanced AI technologies in data interpretation and knowledge representation.

Keywords: Large Language Models · RDF · Summarization · Distinguish Analysis.

1 Introduction

Problem Overview. The exponential growth of data on the Internet day by day through different sources and of various kinds can be very complicated to gather, especially in a limited period of time. If any individual needs information related to a person, place, event, or anything then it would be very difficult to study the massive content available on the Internet. This is why offering data summaries is a useful alternative [17,15,16,4]. In today's world, when most people have a very short concentration span, LLMs play also a key role in providing summarized information. It is quick, human-readable, and easily available to users around the world. Having said this, there exist many limitations of generative AI models as well. The most common issue in generative AI models like ChatGPT is hallucination, it may produce information that is not being asked by the user or is not relevant. LLMs generate text descriptions based on vast amounts of data from different sources, which is why they lack external knowledge, leads to errors, especially when the input is ambiguous or the topic is outside the model's training scope. Collecting data from different sources can be

complicated and time-consuming; hence, the need for automated summarization of a siingle or multiple entities arises. While the comparison between entities can be an important ask, such as four states of Southern America (Alabama, Florida, Arkansas, Delaware) to see the commonalities and distinguishing features of multiple states, here comes the need of Linguistic Summarization and distinguishing analysis of multiple entities.

Background. The motivation behind this study is to fill the previous gaps in this research and the extension of the research done previously. The authors have done research on the linguistic summarization of multiple entities through coding strategy [19]. This study has automated the distinguishing analysis alongside the summarization of multiple entities using Large Language Models such as Chatgpt-3.5, Deepseek, and Mistral explicitly. A reasonable amount of research has been done in natural language generation, based on structured data such as explainability, education and data augmentation [14]. However, LLM's potential hasn't been utilized doing comparisons between multiple entities and distinguishing analysis which has been explored and covered in this research. Due to increasing interest in Natural Language Generation in recent times, the interest in using LLMs with NL Generation also increased, how can we utilize them to acquire knowledge that exists all over the Semantic Web (SW)?

Recent advancements in query-focused summarization using GraphRAG have further informed this work. For example, a two-stage GraphRAG approach has been proposed—first generating an entity knowledge graph and community summaries, then combining them into a global summary which shows improved comprehensiveness and diversity over standard RAG methods [3]. Other studies have extended this idea, such as FG-RAG, which enhances context awareness and fine-grained summarization from graph retrieval [5].These recent works emphasize the effectiveness of GraphRAG architectures for structured summarization tasks and reinforce the relevance of applying graph-based prompting in our study.

RDF and DBpedia as a Data Source. RDF, termed as Resource Description Framework, is a dataset model standard defined by the W3C–World Wide Web Consortium. It is a structured form of data consisting of subject-predicate-object triples, facilitating flexible knowledge representation. However, the volume and complexity of these structured datasets make them challenging for non-expert users to interpret. For example, "Albert Einstein was born in Germany." will be structured as `<Albert_Einstein>` as a subject, `<bornIn>` as a predicate, and `<Germany>` as an object. An RDF graph is a collection of RDF triples, where the RDF triples are edges, directed from the subject to the object [19]. In recent years, the interest in structured data has increased to utilize it in different forms. The dataset used in this study is from DBpedia expanding rapidly, and as of the 2016-04 release, it included detailed descriptions of 6.0 million entities. Among these, 5.2 million entities were systematically categorized within a coherent ontology. This classification includes 1.5 million individuals, 810,000 geographical locations, 135,000 music albums, 106,000 films, 20,000 video games, 275,000 organizations, 301,000 biological species, and 5,000 diseases.

SPARQL Queries – A Standard Way. SPARQL (Protocol and RDF Query Language) is the standard way to retrieve data from RDF graphs. Generally, users are required to formulate the queries in order to get information and it can be difficult for a person who is unfamiliar with graph databases. We designed an automated system in which we can provide multiple entities names to the system, it accesses the RDF Graph through SPARQL query, clean the data and encapsulates in the Excel file which is called Frequency Analysis of the respective

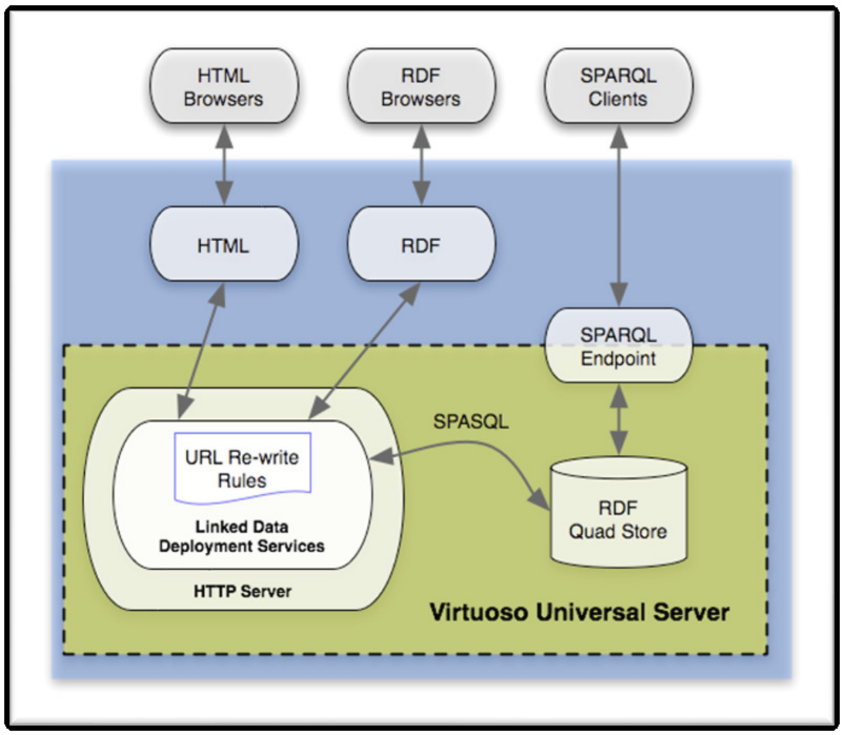

Fig. 1: Illustration of current DBpedia Data Provision Architecture from its Official Website.

entities. These queries enables me to retreive data and make it more user friendly and accessible. SPARQL is the language which machine can understand and different URIs are involved in the whole query.

Figure 1 demonstrates that how the data is being delivered through SPARQL queries, RDF browsers and HTML browsers. In our case, SPARQL queries have been made to the SPARQL endpoint of RDF graph. The SPARQL Endpoint in an RDF-based system acts as an interface where queries are executed against an RDF store, returning results in formats such as XML, JSON, or tabular structures. While SPARQL provides flexibility in accessing linked data, its complexity poses a challenge for non-expert users, necessitating automated summarization techniques to improve readability and usability.

The rest of the paper is structured as follows. Section 2 presents the related work and Section 3 our methodology, including the data preparation, the prompt design and text generation and the setup of the evaluation. Section 4 discusses the experimental evaluation and analyzes the results. Finally, Section 5 concludes the paper.

2 Related Work

The field of querying RDF data and knowledge graphs has seen significant advancements, driven by the need to make complex data more accessible to non-expert users. Summarizing RDF graphs in natural language plays a vital role in helping users understand large and complex datasets.

An automatic natural language summarization system for multiple entities in RDF graphs has been introduced. The system extracts property-object pairs related to entities and converts the structured knowledge into natural language. Practical experiments using human-based evaluations compared summaries generated by machines with those written by humans, both independently and with assistance. The results showed that machine-made summaries are particularly useful under tight time constraints and for difficult topics [19].

The efficacy of text-to-text pre-training for data-to-text tasks has been explored, demonstrating significant performance enhancements for transformer-based models. The study leverages the "Text-to-Text Transfer Transformer" (T5), pre-training models on diverse tasks like translation, summarization, and question answering. Structured data is converted into a linearized text format for input into T5 models, allowing for a straightforward, end-to-end generation process. Results indicate that T5 pre-training achieves state-of-the-art performance across multiple benchmarks, with superior generalization on out-of-domain datasets [6].

The main aim of a recent study is to verbalize SPARQL queries into natural language for better end-user understanding. The authors convert SPARQL queries into natural language questions using LLMs, designing prompts with human-readable labels from knowledge graphs to ensure context understanding. They fine-tune LLMs and evaluate the produced questions using BERTScore, demonstrating that LLMs can convert complex queries into simple, human-readable texts. However, the framework's dependence on knowledge graph labels and the limited evaluation scope are noted as challenges [14].

Earlier works on converting SPARQL queries to natural language relied on grammar rules and smaller language models. The SPARQL2NL approach standardizes queries and refines them through simplification and substitution principles. The LD2NL approach verbalizes OWL and RDF vocabularies into natural language through lexicalization, single triples realization, clustering, ordering, and grouping. These frameworks generate sentences or summaries from resources, rules, or queries, and have been evaluated through human ratings for adequacy, fluency, and completeness [12][13].

More recent approaches have leveraged encoder-decoder architectures for SPARQL-to-text conversion. The NABU approach uses an encoder inspired by Graph Attention Networks (GANs) and a Transformer decoder, showing promising results in multiple languages. However, it still faces challenges with complex queries and conversational contexts [11].

The complexity of formulating SPARQL queries has led to the development of assistive tools using data graph summaries. These summaries recommend structural query elements during interactive query formulation and are represented as RDF graphs. The use of LLMs has further revolutionized query generation, enabling the creation of SPARQL queries from natural language prompts combined with ontologies [2] [1].

Evaluating the quality of generated text is crucial in natural language processing tasks. Traditional metrics like BLEU and METEOR often fail to capture semantic equivalence, leading to the development of BERTScore. BERTScore utilizes pre-trained BERT embeddings to compute similarity between generated and reference texts, providing a more semantic evaluation. This metric has shown high correlation with human judgments in tasks such as machine translation and image captioning [18].

The introduction of BART, a denoising autoencoder, has expanded pre-training methods for natural language generation. BART's pretraining involves corrupting text with arbitrary noise and training the model to reconstruct the original text. This approach allows BART to generalize across different tasks, including text generation, summarization, and machine

translation. However, BART occasionally struggles with hallucinating unsupported information, indicating a limitation in maintaining factual accuracy [10].

The generation of natural language questions from SPARQL queries is significant for tutoring systems and conversational agents. This study focuses on conversational contexts and the challenges of generating coherent and contextually relevant questions. The authors utilized four knowledge-based QA corpora and introduced a new challenge set with unseen query types and domains. The study compared different fine-tuning approaches for BART and T5 models, using various input features and training data. The results showed that while simple questions were well-handled, complex queries and conversational dimensions remained challenging [9].

Semantic graph-based approaches have emerged as a powerful method for abstractive text summarization. These approaches leverage semantic graph representations, such as Abstract Meaning Representation (AMR) graphs, to capture the semantic essence of the text. By combining these graphs with advanced deep learning models, researchers generate more meaningful and coherent summaries. Experiments on datasets like Gigaword and CNN/DailyMail have shown that these methods outperform traditional summarization techniques, particularly in maintaining factual accuracy and generating coherent summaries [8]. Summarization techniques serve various purposes, including indexing for efficient querying, source selection, visualization, and schema discovery. The quality of a summary is evaluated based on its coverage, precision, recall, and computational complexity. Ensuring that the summary accurately represents the original data while remaining computationally efficient is a key challenge in the field [7].

3 Methodology

3.1 RDF Data Preparation

To generate structured and meaningful summaries from RDF-based knowledge graphs, it is crucial to extract, filter, and refine the data before summarization. In this research, data was being retrieved through SPARQL queries to the RDF knowledge graph from DBpedia. Then performed frequency analysis to identify relevant properties and unify synonymous properties to standardize property names. Extracted raw data was very non-informational, with duplicate entries and properties that wouldn't make any sense to retain. These preprocessing steps ensure that the extracted information is concise, relevant, and structured for effective summarization by Large Language Models (LLMs).

SPARQL-Based Retrieval DBpedia is a structured knowledge base that represents information in RDF triple format (subject-predicate-object). SPARQL is a query language that is used to query entity-specific property-value pairs from knowledge graphs. Each query was formulated to:

- Extract all available predicates (properties) and corresponding values for a given entity.
- Filter values to include only English-language literals or URIs while excluding non-linguistic literals (e.g., numerical codes).
- Retrieve multiple entities simultaneously by iterating over a list of entity URIs.

Frequency Analysis & Property Unification The RDF data extracted from DBpedia contains a large number of properties many of which are redundant, irrelevant, or non-informative for summarization. To proceed with the goal, a systematic filtering and frequency analysis approach was applied to refine the data.

Filtering Irrelevant Properties: Several properties in DBpedia store metadata, links, or other non-descriptive information that do not contribute to human-readable summaries. These properties were excluded using predefined filtering rules. Some examples of excluded properties are as follows:

- **Metadata properties** (e.g., `wikiPageID`, `wikiPageRevisionID`, `prov#wasDerivedFrom`) were discarded as they serve administrative purposes rather than conveying semantic information.
- **External links** (e.g., `wikiPageExternalLink`, `owl#sameAs`, `url`) were excluded to avoid diverting focus from factual content.
- **Non-descriptive properties** (e.g., `image`, `fontcolor`, `logosize`, `width`) were removed due to their inability to contribute meaningfully to textual summaries.
- Additionally, all properties with URL values (e.g., http://example.com/resource) were filtered to eliminate redundant hyperlinks.

Frequency-Based Property Ranking: Post-filtering, the remaining properties were analyzed to quantify their occurrence across entities. For example:

- **High-frequency properties**, such as `dbo:birthDate` and `dbo:populationTotal`, were prioritized for critical information. Properties like these that occurs in higher frequency are ranked in descending order, so that LLM can give much weightage to that kind of information with higher frequency.
- **Lower-frequency properties** (e.g., `dbo:nickname`) were retained only if they added unique contextual value. For the entity type 'Person', this information is important and can be critical in a summary when comparing different entities. It is only retained when the field is not empty; otherwise, there is no need to add it.

Handling Numeric and Textual Data:

- **Numeric Properties:** Attributes like `dbo:areaTotal` and `dbo:elevationMaxFt` were aggregated to retain salient quantitative facts.
- **Textual Properties:** Descriptive fields such as `dbo:description` were merged while avoiding redundancy.

Table 1 elaborates on the steps taken to complete the frequency analysis section. It shows example inputs of how occurrences were counted, the basis on which synonyms were handled, how garbage properties were filtered, and how properties were ranked. Examples of the output are then shown. This entire mechanism has been automated through a Python method.

3.2 Prompt Design and Text Generation

Models such as ChatGPT, DeepSeek, and Mistral have demonstrated their ability to process and create human-readable summaries from structured data. This study utilizes LLMs to generate summaries and perform distinguish analysis of multiple entities retrieved from RDF graphs. They need an input as a prompt, a structured prompt with RDF filtered data in our case, and they respond to that prompt accordingly, well-organized summaries in our case.

Table 1: Steps in RDF Property Preprocessing and Their Outcomes

Step	Example Input	Action Taken	Output / Result
1. Count Occurrences	dbo:birthDate (9,500), dbo:height (2,100)	Identify high vs. low frequency properties	dbo:birthDate selected as key info
2. Handle Synonyms	dbo:birthDate, dbp:dateOfBirth	Unified into one property: Birth Date	Reduces redundancy
3. Filter Garbage Properties	wikiPageID, rdf:type, owl:sameAs	Removed from dataset due to low human readability	Cleaner, concise summaries
4. Rank Properties	dbo:birthPlace = High, dbo:award = Medium, dbo:height = Low	Ranked by contextual importance	Priority for summarization set

LLM Prompt Structure To make sure the summaries are coherent and informative enough, a structured prompt was composed in a text file through an automated Python function, combined with preprocessed frequency analysis data. That file was provided to the models for our required summaries. The best possible customized prompt has been formulated after thorough testing in generating concise and human-readable summaries while keeping factual accuracy. The structured prompt includes the following things:

- Entity names to make sure that the generated summary remains entity-specific.
- Key properties, values, and their frequencies to help the model understand the importance of such properties so that it can prioritize those details.
- A well-defined instruction set guiding an LLM to remain focused on key characteristics, differences, and unique attributes of respective entities.

Fig.2 demonstrates the example of the prompt design that how this whole structure has been compiled through an automated agent(method) and then further provide it to an LLM for the exact context we need our summary for.

Summarization and Comparison Tasks

Summarization Process: Once the structured prompts were prepared, the LLMs (ChatGPT, DeepSeek, and Mistral) were used to generate summaries and distinguish analyses. In the summarization process, a structured prompt was provided as input to each large language model (LLM). The models processed the RDF data and generated human-readable summaries. The output from each model was then collected and stored for further evaluation.

Distinguish Analysis: For distinguishing analysis, the LLMs were instructed to highlight key differences between entities based on structured data. The generated text was expected to emphasize unique attributes while maintaining coherence. The results of this analysis were stored to facilitate comparisons between different models.

3.3 Evaluation Setup

For the evaluation process of the outcomes from different LLMs, both automated metrics and human-based assessments have been implemented. Apart from text evaluations, all

```
Create a concise, human-readable summary comparing the following entities.
Focus on their key characteristics, differences, and notable commonalities,
based on the provided data. The text should resemble a short encyclopedia entry,
highlighting their unique features and important similarities.
Exclude any speculative or unverifiable facts.
Provide the summary in a well-structured and clear format.

Entity: Alabama
population: 5039877 (Frequency: 3)
nickname: The Yellowhammer State (Frequency: 2)
areaTotal: 135765 km² (Frequency: 5)
...

Entity: Florida
population: 21538187 (Frequency: 3)
nickname: The Sunshine State (Frequency: 2)
areaTotal: 170312 km² (Frequency: 5)
...
```

Fig. 2: Prompt Structure ensuring LLM generate summaries grounded in factual data.

three models, ChatGPT, DeepSeek, and Mistral have been compared to test state of the art models. Recently, these generative models opens the debate in several aspects that how they can be utilized to solve our real-world problems, which is one of the motivations to work on this particular problem.

Human Evaluation Protocol When it comes to longer texts, automated metrics cannot provide accurate or practical results, they have some limitations. The need for human-based evaluations arises because we can rely on a person who understands the text and has the knowledge to assess longer texts like summaries, in different aspects. While automated metrics such as ROUGE scores, BLEU scores, and entity-based metrics have been implemented to assess summaries in terms of content coverage, linguistic quality, and factual accuracy but none of them have shown sufficiently optimal results to rely upon them entirely. BERT has been implemented for the shorter text evaluations as demonstrated in [18], where authors evaluated questions generated by large language models (LLMs). Particularly, the BERT approach requires two texts to evaluate: 1) the candidate text, which is being evaluated, and 2) the reference text, which is the reference text to compare with. This requirement makes it challenging for summaries, relatively longer texts generated by LLMs. In such scenarios, human intervention is required to make reference summaries with a lot of effort. In a recent study on Linguistic summarization, the authors preferred to adopt human-based experiments for the assessments [19]. In this research, a structured evaluation framework has been developed to comprehensively evaluate the quality of summaries generated through different LLMs, particularly in the context of longer texts.The evaluation process involved generating three separate documents, each focusing on a specific entity type: Person, Place,

and Event. For each entity type, three examples were selected for detailed analysis in the summaries. Consequently, three summaries were generated for each example, resulting in a total of nine summaries across the three documents. These documents contained summaries and comparisons of multiple entities within their respective entity types. For each document, three summaries generated by different models were presented and evaluators were asked to rate the summaries according to several criteria, including accuracy, completeness, redundancy, relevance, and distinguishability of the content. Original data was provided in the form of property-value pairs in a separate Excel sheet, serving as a reference for evaluations. 25 evaluators have been selected from different field backgrounds and collected diverse feedback on the performance of the models. To ensure unbiased evaluations, the models were anonymized. The assessors were then tasked with ranking the summaries based on overall quality. The results of these evaluations were subsequently analyzed and discussed in detail in Chapter 4 of the study. This structured human-based evaluation approach highlights the challenges of relying solely on automated metrics to assess longer texts and underscores the importance of human judgment in ensuring the quality and reliability of LLM summaries.

Automated Metric Computation To complement the human evaluations, some automated metrics were also used to test the quality of text LLMs are generating. Three metrics were aimed at providing an objective comparison of different models based on linguistic quality, readability, and factual accuracy of the generated text.

- **Grammar Score:** To check the grammatical correctness of each summary, LanguageTool has been used, which is an automated grammar checker system. This score was computed by inspecting the number of grammatical errors per word. A high Grammar Score indicates fewer errors and better fluency.
- **Readability Score:** To analyze the text in terms of readability, the Flesch Reading Ease metric was implemented to assess the complexity and clarity of the generated summaries. This metric assigns a score based on sentence structure and word complexity. The higher the score, the more readable the text is for a lay user.
- **Entity Retention Score:** To test the factuality of the text and preservation of original details, a fuzzy matching approach was implemented. Entities extracted from the RDF dataset were compared against the summaries generated using the `fuzz.partial_ratio` function from the `fuzzywuzzy` library. This score reflects how well the original details were retained by the model, offering insights into the factual consistency of the summaries.

4 Results and Evaluation

4.1 Summary Generation for RDF Entities

Various RDF entities, including places, persons, and events, have been covered in the summarization process. Using frequency analysis (with purified and cleaned data), the most common and informative properties have been identified for each entity present in the data. Then, a refined, structured prompt was created in a text file to provide different LLMs for generating summaries in natural language.

The following Figure 3 is one of examples of the generated summaries for multiple entities of **Entity Type–Place**:

Entities Type: Place

Comparison of Alabama, Florida, and Delaware

Alabama

Alabama, known as *The Yellowhammer State*, ranks 30th in total area and 24th in population in the U.S. It covers **135,765 km²**, with 3.2% of its area consisting of water. It has a relatively low elevation, averaging **150 meters**, with its highest point at **735.5 meters (Mount Cheaha)**. In 2010, Alabama's population was approximately **5 million**, with a density of **38.4 people/km²**. The state was the **22nd** to join the Union and is represented by **6 congressional representatives**. Its official motto is *"We Dare Defend Our Rights"*.

Florida

Florida, the *Sunshine State*, is significantly larger than Alabama, ranking **22nd in area (170,312 km²)** and **3rd in population**, with over **21.5 million people** in 2010. It has a high population density of **155 people/km²**. Florida's geography is distinct, with **18.5% of its area covered by water**, making it a state with a strong coastal presence. The state's elevation is much lower than Alabama's, with an average of **30 meters** and a highest point at **105 meters**. Florida joined the U.S. as the **27th state** and has **16 congressional representatives**. Its motto is *"In God We Trust"*.

Delaware

Delaware, the *First State*, is the smallest of the three, ranking **49th in area (5,133 km²)** and **45th in population** with just over **1 million residents in 2010**. Despite its small size, Delaware has a high population density of **190 people/km²**, surpassing both Alabama and Florida. **21.7% of its area is water**, making it more water-dominant than Alabama but less than Florida. It has the lowest elevation, averaging **20 meters**, with its highest point reaching **136.5 meters**. Delaware was the **first state to join the Union** and has the motto *"Liberty and Independence"*.

Fig. 3: Demonstration of generated text for multiple entities.

4.2 Summarization Strategy Comparison

Frequency-Based Selection By using the most frequently occurring properties across the entities to construct a summary that highlights the importance of that property.

LLM-Based Summarization Best prompts have been evolved during the text generation. Prompting different AI models efficiently to generate required summaries based on the data retreived from RDF graph.

Hybrid Approach Finally, the approach of combining the frequency based selections and making best prompts has been utilized to enhance the quality overall.
Three approaches were utilised in order to test the content generated by LLMs, first through just focusing on the most common properties exist for an entity, then just prompting LLM with an instruction set to reduce the hallucinations and then finally combining both to achieve the most realistic and human-like fluent summaries that is called Hybrid Approach as mentioned in Table 2

4.3 Distinguish Analysis of Entities

Distinguishing analysis becomes essential when the user needs to compare multiple entities of the same entity type to see their differences or uniqueness. It focuses on identifying key

Table 2: Overview of Summarization Strategies

Strategy	Focus	Strength	Weakness
Frequency-Based	Most common properties	High data accuracy	Less fluent, rigid structure
LLM-Based	Natural language generation	Human-like fluent summaries	Potential factual inconsistencies
Hybrid Approach	Combines frequency analysis with refined prompting	Balanced accuracy and readability	Slightly complex prompt design

Key Differences and Similarities

- **Size & Population:** Florida is the largest and most populous, Alabama is mid-sized, and Delaware is the smallest.
- **Water Coverage:** Florida has the highest percentage of water area (18.5%), followed by Delaware (21.7%), and Alabama (3.2%).
- **Elevation:** Alabama has the highest elevation (Mount Cheaha, 735.5m), while Florida and Delaware are relatively flat.
- **Statehood:** Delaware was the first to join the U.S., while Alabama (22nd) and Florida (27th) followed later.
- **Representation:** Florida has the most congressional representatives (16), followed by Alabama (6), and Delaware (1).

Each state has distinct characteristics, with Alabama known for its landmass and historical significance, Florida for its large population and coastal geography, and Delaware for its compact size and high density.

Fig. 4: Demonstration of Distinguish Analysis for multiple entities.

characteristics that set one entity apart from another. We analyzed entity descriptions to determine distinctive features based on RDF data by applying LLMs. For example, when we compare Alabama, Florida and Delaware the key differences includes historical events, notable figures and economic focus. Example of Distinguish Analysis can be seen in Figure 4.

4.4 Evaluation Outcomes

Human-Based Evaluation A total of 25 evaluators have been determined to complete the assessment for the generated summaries from diverse and distinct backgrounds to avoid biased results. Also, the model names were being anonymised to prevent the inclination of someone towards a particular model. The evaluators rated the summaries on various factors and ranked them based on their overall quality in the end. Table.3 summarises the distribution of rankings for each model based on evaluator preferences:

Table 3: Human Ranking of Models by Placement Positions

Model	1st Place	2nd Place	3rd Place
Mistral (M)	12	5	8
Deepseek (D)	9	10	6
ChatGPT 3.5 (C)	6	10	9

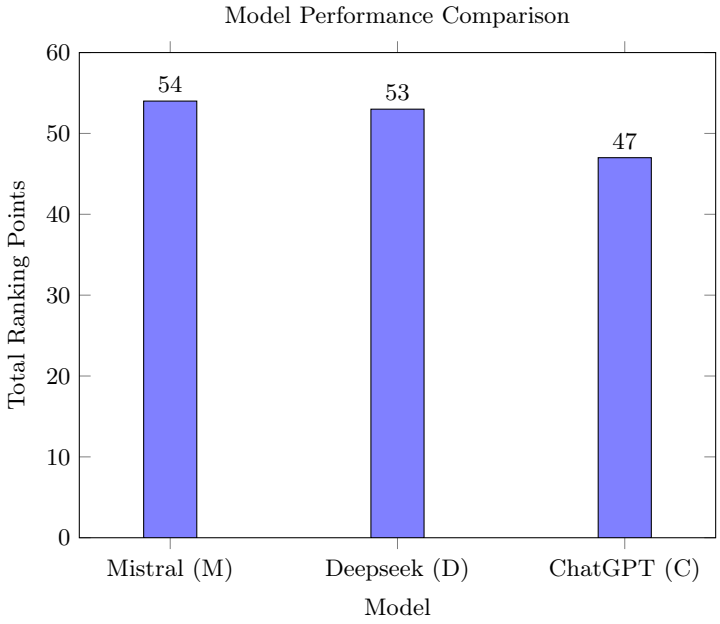

Fig. 5: Comparative evaluation of LLM performance using human assessment metrics.

The formula in equation 1 has been used to compute the overall ranking points for each model given by the evaluators to see the general pattern in people's preferences:

$$\text{Score} = 3 \times (\text{1st place count}) + 2 \times (\text{2nd place count}) + 1 \times (\text{3rd place count}) \quad (1)$$

- **Mistral** was the most successful model in terms of human metrics, securing 1st place in 12 evaluations. It also had the fewest 3rd place rankings, which indicates its consistent performance.
- **Deepseek** demonstrated balanced performance with a significant number of 2nd place rankings, suggesting it is a strong contender but not the top choice among the models.
- **ChatGPT** received the highest number of 3rd place rankings, making it the least preferred model by evaluators in terms of hallucinations, regardless of the prompt engineering. This clearly indicates that ChatGPT is frequently overlooked as the best model for summarization tasks.

These evaluation results in Figure 5, combined with visualizations of the rankings, provide a comprehensive overview of the human-evaluated model performance as perceived by the evaluators.

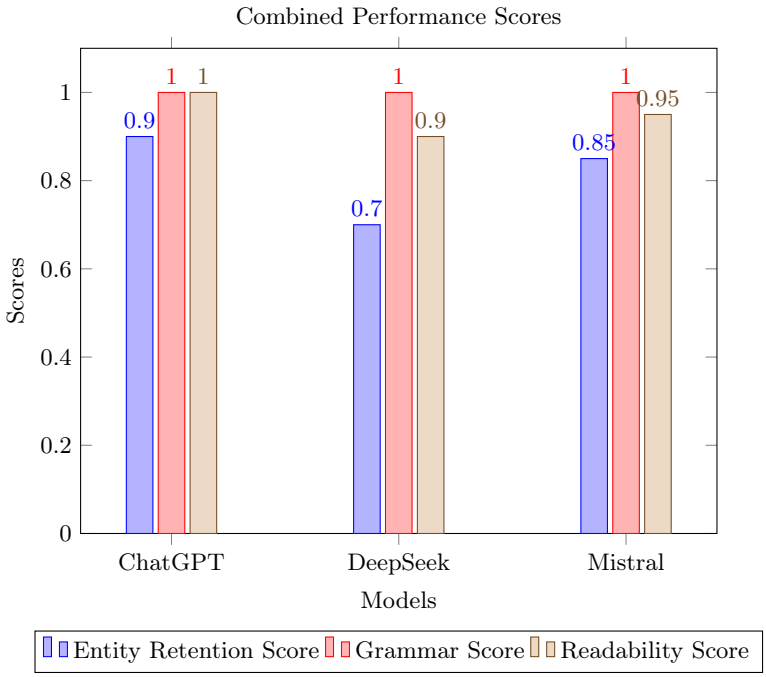

Fig. 6: Comparison of Entity Retention, Grammar, and Readability Scores Across Models

Automated Evaluation Text evaluation is a very crucial part in the projects based on NLP, therefore, to complement the human-based evaluations, three automated metrics have also been utilized. The automated metrics employed in this study provided a quantitative assessment of the summaries generated by different models. The results shown in Figures 9,10 and 11 portray the insights about the grammatical correctness, readability and factual accuracy of the summaries.

The results obtained by automated metrics in Figure 6 provide us with a comprehensive evaluation of the summaries generated by the LLMs. All the models produced text with high grammar scores that indicate the potential of the Large Language Generative Models in producing high-quality content, which is coherent and grammatically correct. Although variations have been observed in the readability and Entity Retention Scores, which suggests the strengths and weaknesses of each model. ChatGPT outplayed Mistral and Deepseek in terms of Readability and Entity Retention highlights its credibility in generating summaries that are both accessible and factually accurate. It shows that RDF data is being retained correctly in the summaries, which were provided within the prompt to summarize, and a user cannot miss facts and figures, showing its highlighting power for a piece of important information. Mistral performed consistently across all the metrics, showing its reliability in generating high-quality summaries. While Deepseek has room for improvement in some aspects, such as entity retention recommends the further refinement could enhance its factual accuracy.

5 Conclusion

In this paper, the potential of Large Language Models (LLMs) in generating summaries and distinguishing multiple entities of knowledge graphs is being tested thoroughly. This study concludes that LLMs, when combined with the refined and cleaned data with thorough data preprocessing techniques such as frequency analysis and property unification, can generate reasonable and informative summaries from complex RDF datasets. Three of the most credible LLMs, including ChatGPT, DeepSeek, and Mistral, were utilized to get the job done and further evaluated for their outputs. The evaluation process encompasses both human-based assessments and automated metrics to test the text quality in various aspects. In our human evaluations, Deepseek performed as the best model, followed by Mistral, and ChatGPT was considered as a last choice in terms of knowledge graph summarization. Moreover, automated metrics indicate that ChatGPT outperformed the other models in specific domains like Entity Retention and Readability Score. However, the Grammer Score remained almost equal for all the models at up to 99%.

References

1. Antoniou, C., Bassiliades, N.: Utilizing llms and ontologies to query educational knowledge graphs
2. Campinas, S., Perry, T.E., Ceccarelli, D., Delbru, R., Tummarello, G.: Introducing rdf graph summary with application to assisted sparql formulation. In: 2012 23rd international workshop on database and expert systems applications. pp. 261–266. IEEE (2012)
3. Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R.O., Larson, J.: From local to global: A graph rag approach to query-focused summarization (2025), https://arxiv.org/abs/2404.16130
4. Gkorgkas, O., Stefanidis, K., Nørvåg, K.: A framework for grouping and summarizing keyword search results. In: Advances in Databases and Information Systems - 17th East European Conference, ADBIS. Lecture Notes in Computer Science, vol. 8133, pp. 246–259. Springer (2013)
5. Hong, Y., Li, C., Zhang, J., Shao, Y.: Fg-rag: Enhancing query-focused summarization with context-aware fine-grained graph rag (2025), https://arxiv.org/abs/2504.07103
6. Kale, M., Rastogi, A.: Text-to-text pre-training for data-to-text tasks. arXiv preprint arXiv:2005.10433 (2020)
7. Kondylakis, H., Kotzinos, D., Manolescu, I.: Rdf graph summarization: principles, techniques and applications (tutorial). In: EDBT/ICDT 2019-22nd International Conference on Extending Database Technology-Joint Conference (2019)
8. Kouris, P., Alexandridis, G., Stafylopatis, A.: Text summarization based on semantic graphs: an abstract meaning representation graph-to-text deep learning approach. Journal of Big Data **11**(1), 95 (2024)
9. Lecorvé, G., Veyret, M., Brabant, Q., Barahona, L.M.R.: Sparql-to-text question generation for knowledge-based conversational applications. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 131–147 (2022)
10. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
11. Moussallem, D., Gnaneshwar, D., Castro Ferreira, T., Ngonga Ngomo, A.C.: Nabu–multilingual graph-based neural rdf verbalizer. In: International Semantic Web Conference. pp. 420–437. Springer (2020)
12. Ngomo, A.N., Moussallem, D., Bühmann, L.: A holistic natural language generation framework for the semantic web. CoRR **abs/1911.01248** (2019), http://arxiv.org/abs/1911.01248

13. Ngonga Ngomo, A.C., Bühmann, L., Unger, C., Lehmann, J., Gerber, D.: Sorry, i don't speak sparql: translating sparql queries into natural language. In: Proceedings of the 22nd international conference on World Wide Web. pp. 977–988 (2013)
14. Perevalov, A., Both, A.: Towards llm-driven natural language generation based on sparql queries and rdf knowledge graphs (2024)
15. Troullinou, G., Kondylakis, H., Stefanidis, K., Plexousakis, D.: Exploring RDFS kbs using summaries. In: The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11136, pp. 268–284. Springer (2018)
16. Troullinou, G., Kondylakis, H., Stefanidis, K., Plexousakis, D.: Rdfdigest+: A summary-driven system for kbs exploration. In: Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018). CEUR Workshop Proceedings, vol. 2180. CEUR-WS.org (2018)
17. Vassiliou, G., Troullinou, G., Papadakis, N., Stefanidis, K., Pitoura, E., Kondylakis, H.: Coverage-based summaries for RDF kbs. In: The Semantic Web: ESWC 2021 Satellite Events. Lecture Notes in Computer Science, vol. 12739, pp. 98–102. Springer (2021)
18. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)
19. Zimina, E., Järvelin, K., Peltonen, J., Ranta, A., Stefanidis, K., Nummenmaa, J.: Linguistic summarisation of multiple entities in rdf graphs. Applied Computing and Intelligence **4**(1), 1–18 (2024)