Knowledge graph Extraction from Textual data using LLM

Khasa Gillani
khasagillani22@gmail.com
Jožef Stefan Postgraduate School
Ljubljana, Slovenia

Erik Novak
erik.novak@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute and Qlector
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute and
Jožef Stefan Postgraduate School
Ljubljana, Slovenia

ABSTRACT

The advent of Large Language Models (LLMs), such as ChatGPT and GPT-4, has revolutionized natural language processing, opening avenues for advanced textual understanding. This study explores the application of LLMs in developing Knowledge graphs from textual data. Knowledge graphs offer a structured representation of information, facilitating enhanced comprehension and utilization of unstructured text. We intend to construct Knowledge graphs that capture relationships and entities within diverse textual datasets by harnessing LLMs' contextual understanding and language generation capabilities. The primary goal is to explore and understand how well LLMs can identify and extract relevant entities and relationships from textual data using prompt engineering while contributing to structured knowledge representation.

KEYWORDS

Knowledge graph, Large Language Models, prompt engineering, information extraction, textual data

1 INTRODUCTION

In an era where data is ubiquitous, efficient organization, retrieval, and interpretation of textual information are crucial. Knowledge graphs, representing facts and relationships in structured forms, play a pivotal role in various AI applications, from enhancing search engines to powering recommendation systems. However, the construction of these graphs is often hindered by the complexity and variability of human language. This paper explores the potential of Large Language Models, like GPT-4, to revolutionize this process. By leveraging their advanced natural language understanding capabilities, we aim to automate and refine the extraction of knowledge from textual datasets. The fundamental purpose of this research is to understand the extent to which LLMs can identify and extract relevant entities and relationships from textual data and then build a Knowledge graph using the extracted information.

The motivation behind this study stems from the growing need to effectively manage and utilize the vast amounts of textual data generated daily. Knowledge graphs offer a structured and intuitive way to represent information, but their construction is often

Information Society 2024, 7–11 October 2024, Ljubljana, Slovenia

https://doi.org/https://doi.org/10.70314/is.2024.sikdd.15

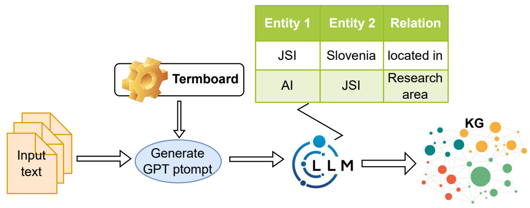

Figure 1: Overview of proposed approach where input text is processed through a Termboard to generate a structured prompt for LLM, creating an entity-relation table to build a Knowledge graph (KG).

labor-intensive and requires expert knowledge. However, constructing Knowledge graphs from unstructured text is intricate and depends on sophisticated natural language processing (NLP) methods, including named entity recognition (NER) and relation extraction. The advancement of LLMs like GPT-4 presents an opportunity to automate and improve this process as illustrated in Figure 1. Utilizing LLMs can lead to more efficient, scalable, and accurate Knowledge graph construction, thereby unlocking new possibilities in information management and AI applications.

2 BACKGROUND

An overview of recent research in Large Language Models and Knowledge graphs is provided in this section, which also emphasizes the potential for their integration.

2.1 Large Language Model (LLM)

Large Language Models are advanced AI systems pre-trained on extensive data, enabling them to comprehend and produce human language. Their recent surge in popularity is due to their proficiency in various language-processing tasks, including text completion, translation, summarization, and answering questions. These models, primarily based on transformer architecture, utilize self-attention mechanisms through encoder-decoder modules. Encoders transform input text into numerical embeddings that reflect the context and meaning, while decoders use these embeddings to generate coherent and pertinent textual output. The large language models feature a decoder-only architecture and, thus, make a prediction of the target output text using only the decoder module. The training paradigm for these models is to predict the next word in the sentence. Generally, large-scale decoder-only LLMs such as ChatGPT [7] and GPT-4 [2], focus on human-like language output, predicting subsequent words based on the preceding text for tasks like text generation.

Table 1: Simplified comparison between Large Language Models (LLMs) and Knowledge graphs (KGs)

Feature	LLM	KG
Knowledge type	Broad, general knowledge	Structured, domain-specific knowledge
Data handling	Flexible, can process varied inputs	Requires structured data
Accuracy	May lack precision in understanding	Highly accurate with structured data
Understanding	Can interpret and generate language	Designed for specific queries and relationships
Adaptability	Adapts to new information by retraining	Adaptable when updated with new data
Transparency	Often seen as "black boxes" with unclear reasoning	Clear decision-making pathways
Error rate	Can make mistakes due to broad generalizations	Can be prone to errors if data is incorrect or missing
Complexity	Handles complex language tasks	Manages complex relationships and attributes
Usage	Broad applications in text generation, translation, etc.	Used for specific tasks like recommendations, search optimization
Scalability	Scales with computational power	Scales with the amount of structured data available

2.2 Knowledge graph (KG)

Knowledge graphs are structured representations of information that depict the relationships between entities in a specific domain. They are used extensively in various applications, such as search engines, recommendation systems, and question-answering systems. These graphs use detailed connections between data to help with smart thinking, finding specific information easily, and running applications that use knowledge. Hence, allows us to better understand and use information across multiple fields.

Knowledge graphs provide a structured way of representing interconnected knowledge. They are precise and consistent, aiding in decisive and informed decision-making. KGs are particularly valuable for their interpretability and explainability due to the explicit representation of entities and relationships. They can capture domain-specific information accurately and evolve to incorporate new data. However, KGs may suffer from incompleteness and may not always reflect the most recent or unseen facts. They also typically cannot understand natural language in an unstructured format [3][6]. Moreover, KGs are preferred in scenarios where explainability and interpretability are crucial, as they provide structured knowledge representation.

2.3 Combining LLM and KG

The comparison between Large Language Models and Knowledge graphs (Table 1) can be supported by various references that highlight their respective strengths and weaknesses [4]. Large Language Models like ChatGPT [7] are celebrated for their generalizability and ability to process diverse text data, allowing them to perform various language-related tasks without extensive task-specific training. They can act as reservoirs of general knowledge, aiding in information synthesis and research. Their proficiency in language processing is useful in tasks like natural language understanding and sentiment analysis. However, they can suffer from hallucinations, where they generate plausible but factually incorrect information. Their "black-box" nature makes it difficult to understand the internal decision-making processes, and they can be indecisive, producing uncertain responses to ambiguous inputs. Additionally, while they have vast general knowledge, they may not be up-to-date with domain-specific or the latest information. Critics of LLMs argue that these models lack transparency and interoperability.

Recent research [3] [4]efforts are, however, improving LLM's interpretability through techniques like attention mechanisms and model introspection. KGs also present advantages over LLMs by providing knowledge about long-tail entities, thus improving recall for knowledge computing tasks. However, both LLMs and KGs can perpetuate biases present in their training data or construction methodologies. In conclusion, both LLMs and KGs have their unique strengths and challenges. While LLMs excel in general language processing and knowledge extraction from vast corpora, KGs provide a structured and interpretable way to organize explicit knowledge. These differences underscore the potential benefits of integrating LLMs and KGs to create more robust AI systems that leverage the strengths of both approaches.

3 PROOF OF CONCEPT: ANALYSIS AND KNOWLEDGE GRAPH GENERATION

This section demonstrates how to process and analyze textual data to build a Knowledge graph using LLM. It is important to mention that prompt engineering [5] is of great importance when it comes to the results generated from ChatGPT. Since it is a generative model, small variations in the input sequence can create large differences in the produced output as demonstrated below. We use two different textual files containing contextual data: (i) APRIORI proposal (containing project details, job description, potential candidate skills, hosting organizations, etc.) and (ii) ADRIA Motorhome instruction manual (containing textual as well as tabular data). Moreover, building KG out of the ADRIA instruction manual has potential applications for the manufacturing industry.

3.1 Using ChatGPT Prompts:

We compare ChatGPT-3.5 and GPT-4 extracted entities and relations using the same prompts. We use Termboard[1] which offers customized ChatGPT prompts to create terms, entities, and relations to visualize larger graphs from the provided text.

Prompt: Extract an ontology and create a table of relations with 3 columns in this order: source, target, and relation name. Also Create a table with 2 columns: put in the first column the name of the term and in the second column an elaborate definition of the term. Use this text as a basis: ¨APRIORI¨- (contains textual data about the job description, candidate skills, project description, hosting organization, etc).

Observing the Knowledge graphs generated by ChatGPT-3.5 (Figure 2) and GPT-4 (Figure 3); we notice, that it didn't extract all entities and relations and missing terms/concepts. For this reason, we ran the second prompt, where we redefined a more detailed prompt to ask GPT-4 to explicitly generate a comprehensive ontology including all entities and relations from the provided text, categorize entities into types like Persons, Organizations,

[1]https://termboard.com/

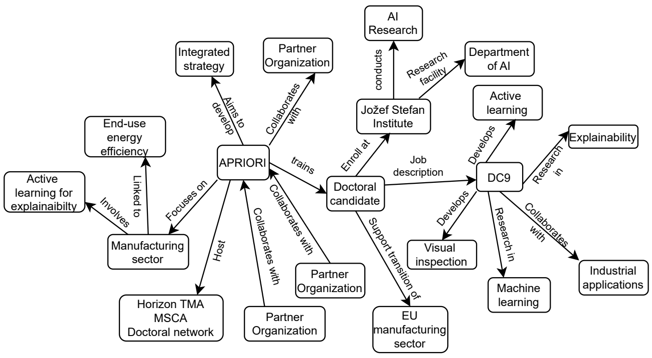

Figure 2: The KG generated using ChatGPT-3.5 contains 20 entities. It was able to extract entities and link them to relations, but it failed in abstracting concepts and specifying entities (i.e. partner organizations, location, etc.).

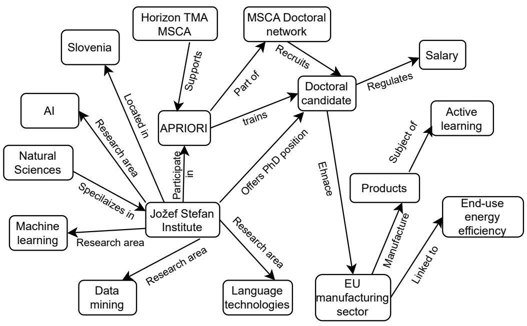

Figure 3: The KG generated by GPT-4 contains 16 entities. It was able to identify abstract concepts, and geographic entities that ChatGPT-3.5 doesn't. Extracted more elaborated entities with relations.

and concepts, and Geographic Locations, and then identify the relations between these entities. Providing additional information to GPT-4 resulted in an improved Knowledge graph (Figure 4). However, ChatGPT-3.5 didn't produce a quality graph (Figure 5) compared to Figure 2.

3.2 Python Implementation

We use a free, open-source library called spaCY [2] for advanced NLP in Python. We employ the named entity recognition technique to identify named entities from a given text using the spaCY model (en-core-web-sm). We used a chunk of textual data from the ADRIA Motorhome manual for experiment purposes. Table 2 compares entities, relations, and triplets extracted from the raw texts. The table shows that the number of triplets extracted by algorithms is similar–(Figure 6 and Figure 7). However, the number of entities that spaCY extracts are larger but not every pair of entities is connected by meaningful relation, leading to fewer triplets. Thus defeating the purpose of creating a Knowledge Base. When using spaCy for entity extraction, the entities are typically recognized based on the named entities present in the text. Named entities are often specific nouns, such as names of people, organizations, locations, dates, or product names. spaCy might not identify it as a specific entity by default. So to extract specific entities, it might need to customize spaCy's NER model

[2]https://spacy.io/models

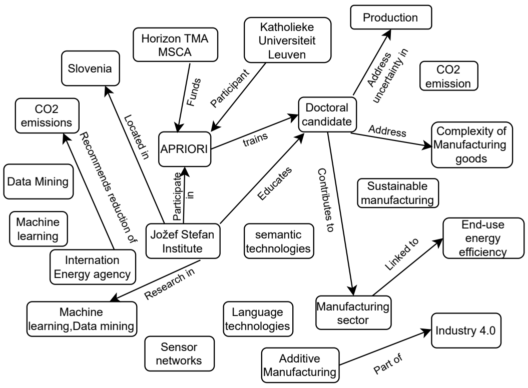

Figure 4: The KG generated by GPT-4 contains 22 entities. It Identified more key entities and relevant concepts and identified suitable relations to connect them (i.e. participant-Katholieke Universiteit Leuven). However, it didn't cover all relations and classes (i.e. skills). We also notice a few duplicated entities(i.e. data mining, CO2 emission, etc.) and some independent entities (i.e. sustainable manufacturing).

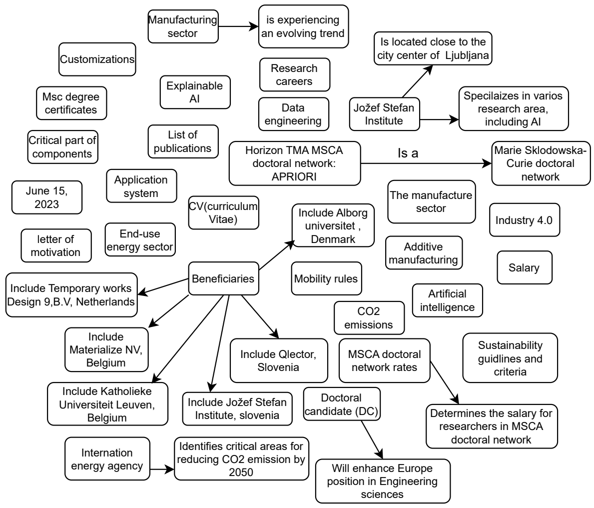

Figure 5: ChatGPT-3.5 was able to extract a larger number of entities but it was not successful at abstracting concepts and missing relations. Entities and relations found frequently represented complete sentences rather than concepts. This occurs because ChatGPT is a conversational model trained on a task to create responses to a given prompt and is not particularly trained to recognize entities and relations

or provide additional context for better recognition. Hence results can be improved by pre-processing data into a structured format.

4 EVALUATION

When there is no ground truth data available, creating an automated evaluation metric for a Knowledge graph becomes challenging. In such cases, the evaluation relies on qualitative principles to assess the results. Based on the practical framework defined in the study [1], the following principles were identified:

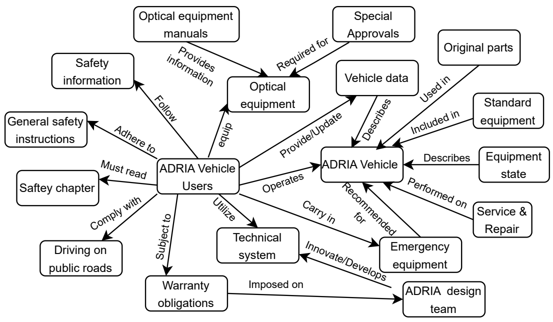

Figure 6: The KG generated by GPT-4 contains 18 entities using the ADRIA motorhome instruction manual. It extracted concepts relevant to ADRIA users and vehicle instructions, their functions, and how they are connected.

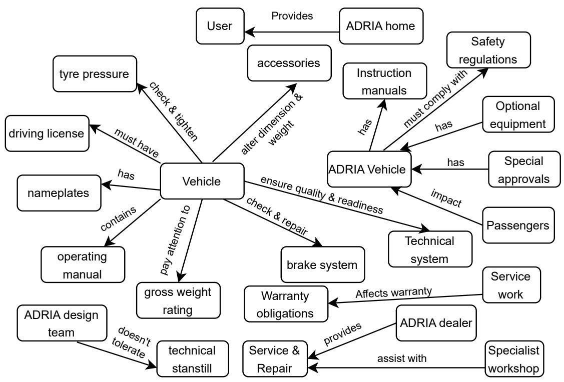

Figure 7: The KG generated by ChatGPT-3.5 contains 24 entities. Extracted more entities relevant to ADRIA vehicles but relations between entities are more generic and entities are duplicated.

- Triplets should be concise.
- Contextual information of entities should be captured.
- The Knowledge graph does not contain redundant triples.
- Entities should be densely connected.
- Relations among different types of entities should be included.
- Knowledge graphs should be organized in structured triples for easy processing by machine.
- For tasks specific to a particular domain, it's essential that the Knowledge graph is tailored and relevant to that specific field

According to these principles, in our use case, we manually inspected the Knowledge graphs generated above, and we can conclude that the ChatGPT-3.5 approach provides a more detailed Knowledge graph without abstract concepts compared to the GPT-4. However, to create these Knowledge graphs, a few steps of refining the answers from ChatGPT are needed. Sometimes the produced output is incorrect and needs to be corrected before proceeding. When we redefined the prompt, GPT-4 identified more specific entities, and concepts compared to ChatGPT-3.5. Even though ChatGPT extracted a larger number of entities, it failed to provide abstract concepts and entity-relation.

In the second part of the experiment, we employed the NER method to extract relations and entities from the given text (i.e. ADRIA). We analyzed that extracted entities are duplicated and relations have some noise and incomplete information. If you have specific patterns or structures in mind that you want to extract entities and relations based on, you may need to customize the relation extraction logic. Alternatively, more advanced natural language processing techniques or pre-trained models designed for relation extraction tasks might provide better results. Also, we analyzed half of the relations-entities extracted by spaCY and ChatGPT are overlapped.

Table 2: Knowledge extraction comparison. (ADRIA motorhome manual dataset)

Algorithm	Entities	Relations	Triplets
GPT-4	18	20	20
ChatGPT-3.5	24	18	18
spaCY	22	14	17

5 CONCLUSION

The proposed exploration of using LLMs for Knowledge graph extraction holds promise for advancing our understanding of how advanced language models can contribute to structured knowledge representation. This paper explores using LLMs to generate Knowledge graphs out of source documents. We utilized ChatGPT-3.5 and GPT-4 models to generate the Knowledge Graphs for two different textual data and compared the structure of the KGs. GPT-4 performed better as it successfully identified more abstract concepts and key entities compared to ChatGPT-3.5. Therefore, it provides insights into the practical application of LLMs in developing structured knowledge from unstructured textual data, with potential applications in knowledge-based AI applications, paving the way for more effective information processing and utilization. In future studies, we intend to use a more formal framework to evaluate the quality of created Knowledge graphs. Such a framework will allow us to efficiently analyze the quality of KG and provide a standardized method to forecast missing linkages between concepts and relationships within a given domain.

ACKNOWLEDGEMENTS

This research is supported by EU funding HE MSCA Project Apriori (GA: 101073551). The author acknowledges the usage of ChatGPT and Grammarly for content paraphrasing, grammar, and error checking.

REFERENCES

[1] Haihua Chen, Gaohui Cao, Jiangping Chen, and Junhua Ding. 2019. A practical framework for evaluating the quality of knowledge graph. In *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding: 4th China Conference, CCKS 2019, Hangzhou, China, August 24–27, 2019, Revised Selected Papers 4*. Springer, 111–122.

[2] R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2, 13.

[3] Jeff Z Pan et al. 2023. Large language models and knowledge graphs: opportunities and challenges. *arXiv preprint arXiv:2308.06374*.

[4] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: a roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

[5] Elvis Saravia. 2022. Prompt engineering guide. (2022).

[6] Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models. *arXiv preprint arXiv:2305.04676*.

[7] Ce Zhou et al. 2023. A comprehensive survey on pretrained foundation models: a history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.