SEMANTIC TECHNOLOGIES AND KNOWLEDGE GRAPHS

Professors:
Claudia D'Amato
Nicola Fanizzi

Students:
Francesco Didio
Donato Francioso

Index

Abstract:

Violence against women is a global problem encompassing various forms of physical, sexual, psychological and economic abuse, representing a serious human rights violation rooted in historical and structural gender inequalities. This work, developed in the context of the Horizon Europe Seeds project of the University of Bari Aldo Moro, aims to create a structured data collection using artificial intelligence to analyze the phenomenon of violence against women. The goal is to build knowledge graphs from the unstructured data of the rulings issued by the European Court of Human Rights (ECHR) [1].

The approaches adopted include the use of large language models (LLMs) for knowledge graph construction and extraction, compared with traditional entity and relation extraction techniques using NLP. The construction of the knowledge graphs was done by initially creating a Retrieval-Augmented Generation (RAG) system for each sentence analyzed in order to reduce the hallucination phenomenon. Finally, competency questions specific to violence against women were formulated to check the accuracy and completeness of each knowledge graph created.

Introduction:

Ontologies are a representation, formal and explicit, of a set of concepts within a domain and the relationships between them. The main goal of ontologies is to facilitate the sharing and reuse of knowledge among different systems and applications, enabling a common understanding of the domain among various stakeholders. However, their construction is a complex engineering task that requires significant time and resources [2].

Today, this task could be made easier thanks to several emerging technologies, such as large language models (LLMs). This study sought to analyze the usefulness of these models to see if their use could offer concrete support. LLMs, with their billions of training parameters, have revolutionized the field of natural language processing (NLP) [3]. They are capable of performing complex linguistic tasks and were therefore chosen for this case study.

Two specific LLMs were used in this study: GPT-4.o [4] and Mixtral 8x22b Instruct [5]. GPT-4.o was used to create the initial ontology, while Mixtral was used to enrich and specialize the ontology, create the Knowledge Graph (KG) for each document, and to create the competency questions (CQs).

Tested approaches:

The initial idea of this project was to use an inverse approach to the one finally adopted. This type of approach was examined throughout the study [6]. Initially, it was intended to create the Competency Questions (CQs) using large language models (LLMs). Then key concepts extracted from the responses of the CQs and the CQs themselves were used to construct the knowledge graph and ontology, respectively.
Therefore, the intended process was: *CQs → ontology → Responses to CQs → Knowledge graph.*

CQs were created using Mixtral by asking to generate them considering our domain.However, since they were not specific to the structure of the input documents, they were useless. This then resulted in a failed, or incorrect, generation of the KG.
In addition, the LLM could not generate a good ontology only from the key concepts of the CQs but needed a base ontology on which to build that specification.

Among the LLMs tested, Mixtral 8x7b-Instruct, Mixtral 8x22b-Instruct, CodeLlama-7b-Instruct, and StripedHyena-Nous-7B were considered. It was chosen to use Mixtra 8x22b because its maximum context-length was about 65 thousand tokens, twice as long as the others.

Prompt engineering:

For tasks involving the use of large language models (LLMs), employing an effective prompt is crucial. Prompt engineering involves creating an appropriate prompt to get from LLMs exactly what the user wants [7]. Two techniques were adopted in this work: *zero-shot prompting* and *few-shot prompting*.

- **Zero-shot prompting**: With this technique, no example of the desired result is given in the prompt, since it is easily expressed. It is suitable when the tasks are relatively simple.
- **Few-shot prompting**: When you cannot clearly express what you want, you can create a prompt by including an example of the expected result. If the tasks are complex, this technique is preferred to avoid the hallucination phenomenon.

Work pipeline:

The work pipeline was divided between work done with large language models (LLMs) and work done using more traditional NLP techniques.

The pipeline of LLMs consists of 5 main steps:

1) Document preparation(full-text and subpart)
2) RAG's creation
3) Base's ontology creations

4) KG creation and merging
5) Creation and answering of CQs

The NLP pipeline consists of 3 steps:

1) Preprocessing
2) Part of speech
3) Triple's creations

LLM's pipeline:

Document preparation:

The case study involved the analysis and creation of knowledge graphs (KG) from legal judgments, taking into consideration two types of input. The first type involved the use of the entire judgment (full-text), while the second was based on a specific sub-part (sub-part) chosen by domain experts. The objective was to evaluate the usefulness of both types of input.

For preparation, functions were used to extract text from PDFs in the first type, while for the second type, due to the complexity and different structure of each judgment, text extraction was done manually. Once extracted, the text was inserted into a text document, ready for the next step.

RAG's creations:

As explained, Retrieval-Augmented Generation (RAGs) have been used to limit the output of large language models (LLMs) as much as possible to the context of each document. Although LLMs are trained on billions of data, they can sometimes find it difficult to respond to specific tasks. For this reason, RAGs, an enhancement of classical LLMs based on a well-defined corpus of text, are used [8]. In this way, LLMs have two types of data available to answer user questions: parametric data and nonparametric data. The parametric data represent the training data, while the nonparametric data are the new data, saved in an external vector container that the LLM will go to inspect to provide the answer.

For the creation of the RAGs, the TogetherEmbedding method [9] of the LangChain library was used. As the embedding model for semantic search within the vector dataset, bert-m2 was used, while, through the langchain_community library, FAISS [10] was used as the vector dataset. RAGs were created for all five documents considered, which were then used as the context for the LLM to respond to the various prompts provided.

Base's ontology creations:

At this stage, the ontology used as a T-box for the construction of the final knowledge graphs was created. Initially, the ontology was generated using GPT-4.o, since Mixtral 8x22b was unable to create a general schema for this type of topic. The use of GPT-4.o provided a basic ontology containing classes (such as Abuse, LegalCase) and ObjectProperty. Later, this ontology was enriched with additional elements using Mixtral, since in the first attempts to generate KG it kept creating, even if not explicitly requested in the prompt, additional elements such as DataProperty and ObjectProperty.

To enhance the ontology, a zero-shot prompting technique was applied to a document within our domain. The prompt asked for the ontology to be expanded based on the document's fundamental concepts. This process was repeated for several randomly selected documents. After multiple iterations, it was observed that Mixtral consistently generated the same elements regardless of the document. This led to the conclusion that the document structures were similar, allowing the generated elements to be commonly reused across different documents, thereby recognizing common patterns among them..

Once the ontology was completed, a manual check was made to remove superfluous elements within the schema, as some DataProperties were inconsistent or redundant for our purpose.

Kg's creations:

Then starting from the ontology created and the RAG of each document, a prompt engineering phase followed to create the Knowledge graphs for each document. The few-shot prompting technique was followed for the engineering phase of this prompt. Once the knowledge graphs for each document were created, they were merged while maintaining the entities of each and avoiding inconsistency.

CQs creations and answering:

As a final step, Competency Questions (CQs) were created using Mixtral. For this task, during engineering, the ontology was provided to the LLM and requested to create a list of CQs based on it. The CQs generated were perfectly in line with the ontology schema, so it was not necessary to refine them manually, but only a few were selected.

After the CQs were created, again using Mixtral, a zero-shot prompt was developed that, based on each document, provided the answers to each CQ. For each answer, a manual check was made for consistency with both the reference knowledge graph and the original sentence in order to avoid any hallucinations. Finally, once the responses were verified, they were formatted in CSV format.

NLP’s pipeline

Preprocessing:

This step, as explained above, was carried out to compare the new LLM-based approaches. The entire process was applied to only one of the five documents used previously. Initially, after reading and extracting text from the document, a preprocessing step was performed using the NLTK library [11]. The preprocessing steps included punctuation removal, tokenization, stopword removal and lemmatization.

Part of speech:

Once the text was tokenized, the Spacy library was used to identify the Part of speech (POS) of each token. POS is an NLP technique that helps categorize the elements that make up a sentence. Therefore of extreme utility in complex tasks such as Sentiment Analysis or Machine translation. In this study it was used to create the triples from the document. In fact, each token was divided into "subj," "verb" and "obj" according to the role within it.

Triple’s creations:

As the last step in the pipeline, triples were created by associating each subject with each verb and object.

Results and discussion:

This paper aims to study how large language models (LLMs) can contribute to knowledge engineering tasks by generating ontologies from scratch, creating instances by extracting information from different data sources, and comparing the results with older NLP standards.

Starting from the first step, it was observed that the LLM model encountered difficulties in generating a complete ontology for a specific domain such as the one under consideration, relying only on its initial knowledge. As a result, it was necessary to expand the ontology using documents as a guide for specialization. This suggests that, for ontology creation from scratch, the LLM needs additional support, ideally provided by a domain expert in cases such as this.

It was observed that the ontology worked effectively in most cases. However, there were documents where the LLM, despite the ontology having been created following their structure, failed to generate complete instances. This highlights the need for human oversight throughout the process, capable of checking the correctness of the data and recognizing any limitations of the model.

As for the final part, Table 1 shows the total number of responses to the CQs for the documents, considering both proposed approaches, with a focus on responses consistent with the text, for example, by removing the empty entities that LLM used to generate just to create links. The CQs in the table are ordered according to the importance given by the LLM in the context of our case study, from top to bottom. One check made for the correctness of these responses is the manual check within the response document.

Results	**Full-text**	**Sub-part**
Which legal case is associated with Violation?	3	4
What is the legal outcome of Case?	4	4
What is the reason stated for the judgment?	4	2
What abuse is related to Judgment?	5	4
What is the severity level of Abuse?	2	2
What is the duration and frequency of Abuse?	3	2
Which legal articles are violated?	4	4
What is the context of Abuse?	5	4
Which court judged the Case?	2	1
What are the damages related to Judgment?	1	3
What are the consequences of Abuse?	5	4
How much in legal damages and costs were awarded?	0	4
What is the legal status?	5	3
Total	40/65	37/65

Table.1 Score CQs answering

This type of approach was chosen because, since we did not have the cooperation of domain experts and therefore did not have the opportunity to qualitatively evaluate the responses, it was still possible to conduct a quantitative analysis.

It can be seen from the data in the table that using the whole document the number of responses received is higher, even considering the importance of CQs.

Thus, the utilization of the entire document is better in terms of quantity, as could be expected considering that it contains a greater amount of information. However, it is critical that the responses generated are viewed and evaluated by a domain expert. This is because there is no guarantee that the responses drawn from the entire document are correct or relevant to the standards and interest of the experts. The LLM, by processing a vast amount of information, could direct the focus to details or aspects not needed by experts. On the other hand, in the case of the sub-part chosen by the experts, the information sought is more likely to be present and relevant to their specific interest. This does not necessarily imply that they are wrong, but it does indicate that the LLM may pick up on various aspects of the document depending on the approach used. As mentioned earlier, there is no guarantee that the answers generated by the LLM using the entire document are those sought for the particular purpose or interest of the experts. Therefore, the experts' interpretation and evaluation of the responses are crucial in determining their relevance and correctness for the specific application.

Another observation to consider concerns the time and limitations in using the full document for the Knowledge Graph creation process. In fact, the LLM has not only technical limitations regarding the large number of input tokens required for the full document, but it is also significantly slower in processing it.

It can be concluded that if the goal is to retrieve as much information as possible in quantitative terms, the approach using the entire document is more effective. However, if the focus is on a specific part of the document, such as that selected by the experts, then the second approach is preferable. The latter is faster, has fewer technical limitations, and can provide a number of responses of satisfactory quality for the specific purposes of the analysis.

Regarding the approach with the more classical NLP techniques, it can be said that the number of triples generated by a single file far exceeds those generated by the LLM approach. However, using a standard approach with text preprocessing and POS of tokens, a lot of semantic information is lost in sentences such as: "Article 3" which is tokenized as two different words, "Article" and "3" and not as one word. Therefore, from a qualitative point of view, this kind of approach is not comparable to that with LLMs.

Future works:

Based on this work, future developments could be thought of, trying to improve on what has been done. Below is a list:

- Have domain experts evaluate the results obtained to have a qualitative review on the outcome.
- Increase the number and process documents to enlarge the Knowledge graph, including using a combination with results obtained from NLP techniques.
- Use other LLMs than those used (Mixtral 8x22b and GPT 3.5), using weights through a provider such as Hugging Face.
- Improve prompting or use other techniques such as Graph Prompting [12].

- Have the CQs generated by domain experts so as to help the LLM with the generation of what is needed.

References:

[1]: Court of Human Rights (CEDU). "HUDOC - European Court of Human Rights." Available on: https://hudoc.echr.coe.int/

[2]: Funk, Maurice, Simon Hosemann, Jean Christoph Jung, Carsten Lutz. "Towards Ontology Construction with Language Models."

[3]: Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, Jianfeng Gao. "Large language models: A survey."

[4]: OpenAI. "GPT-3.5 Turbo Documentation." Available on: https://platform.openai.com/docs/models/gpt-3-5-turbo

[5]: Mistral. "Mistral Documentation." Available on : https://docs.mistral.ai/getting-started/models/

[6]: Kommineni, Vamsi Krishna, Birgitta König-Ries, Sheeba Samuel. "From human experts to machines: An LLM supported approach to ontology and knowledge graph construction."

[7] Diario di un Analista. "Guida al Prompt Engineering." available on: https://www.diariodiunanalista.it/posts/guida-prompt-engineering/

[8]: Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks."

[9]: LangChain. "LangChain Documentation: TogetherEmbedding." Available on: https://python.langchain.com/v0.2/docs/integrations/text_embedding/together/

[10]: LangChain. "LangChain Documentation: FAISS." Available on: https://python.langchain.com/v0.1/docs/integrations/vectorstores/faiss/

[11]: NLTK. "Natural Language Toolkit Documentation." Available on: https://www.nltk.org/

[12]: Liu, Zemin, Xingtong Yu, Yuan Fang, Xinming Zhang. "GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks."