Semantic Web Project : Zero-shot prompting of LLM for question answer task with Knowledge Graphs

Karan Gupta
2021258
karan21258@iiitd.ac.in

Shivesh Gulati
2021286
shivesh21286@iiitd.ac.in

1. Motivation

Training or even fine-tuning a Large-Language Model (LLM) with millions of parameters from scratch takes a lot of resources. The main object of this project is to harness the zero-shot capabilities of large-language models by augmenting the prompt with relevant triples from the knowledge graph to provide context and generate better-suited responses to the input questions from the LLMs. Through this project, we aim to analyse the improvement in the responses generated from the LLMs after providing the context using various triple retrieval techniques. In this project, we also try to develop our triple retriever from the knowledge graph and analyse and compare its performance to pre-existing machine learning and NLP-based retrievers.

2. Literature Review

[1] Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering This study aims to develop a framework for improving the performance of large language models for question-answering tasks in a zero-shot setting. The paper achieves this objective by prepending top k-most relevant triples from a knowledge graph extracted from the knowledge graph using a triple extractor and then finding the most relevant triples after creating embeddings from natural language using sentence transformers like **(mpnet-base-v2)** and then analysing the performance jump achieved on the said task before and after adding the context.[1]

3. Dataset Description

For this project, we have chosen the Mintaka Dataset.It consists of general questions from a wide variety of domains. We used the test split of the dataset of 4000 questions. Out of 4000 questions we generated LLM responses for only 2000.

4. Methodology

Our current methodology follows three steps:-

4.1. Extracting the named entities from the prompts

For this project, we have chosen the Mintaka dataset **Mintaka Dataset**, which already has named entities for each of the questions extracted with their wiki database id alongside the answer entity. It consists of general questions from a wide variety of domains.

4.2. Retrieving triples from the knowledge graph using graph search algorithms

After receiving the relevant named entities from the question, we apply our triple retrieval process. This process consists

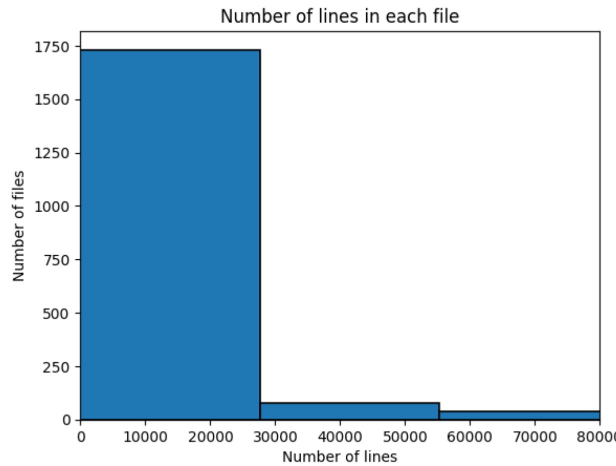

Figure 1. triples extracted per question

of executing a SPARQL query on the wiki data knowledge base, which retrieves all the triples in which the predicates belong to the **wdt: namespace**, which is the namespace for all the direct properties of a given entity. This filters all the identifiers and id triples, which may not be relevant for answering the question. Then, for each of the retrieved triples, a dfs call is made on the object retrieved, and at each call, the SPARQL query described is executed again to retrieve the relevant triples at further levels. This way, our retriever can extract the triples from the knowledge graph at a distance of k-hops from a given entity. For now, we have specified (k=2), i.e. for each of the entities extracted from the question text, we are retrieving triples at a distance of k=2 hops away. [1]

4.3. Verbalizing the triples

The subject, object and predicate of the triples extracted were concatenated and saved in a text file to convert them to natural language form for semantic retrieval. This may not be the best possible method, but it can be further improved to get many triples in a better-suited natural language form.

4.4. Getting the relevant triples

The triples extracted from the retriever in step 2 were very large and were much beyond the word limit of prompts for many existing LLMs; hence, we decided to reduce the number of triples by extracting the most relevant triples from the ones obtained. For this purpose, we created a pipeline using **Lang Chain** in combination with a pre-trained **Instructor Embedding Model** which is publicly available on Hugging Face. The lang-chain pipeline splits the text file containing the triples into sub-files. It was converted into a **FAISS (Facebook AI**

[1] https://drive.google.com/drive/folders/1UVgzQCk1FHU8gmkiGA_-tFqQ61bwpLQa?usp=drive_link

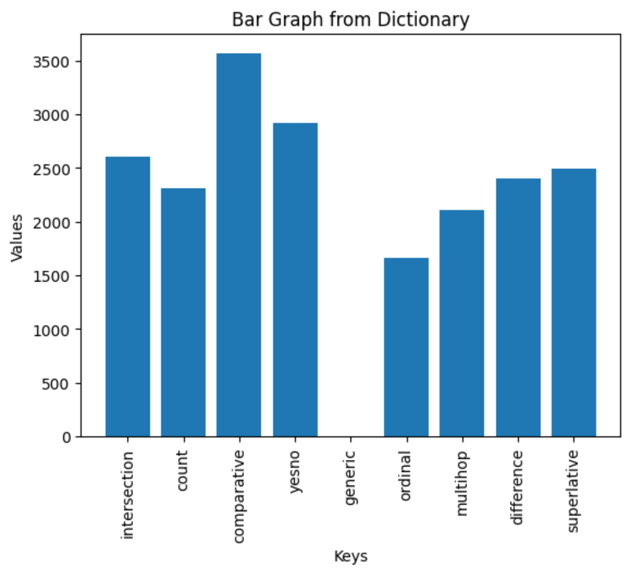

Figure 2. median of number of triples per category

Similarity Search) vector database, and a semantic search was conducted based on the question provided to the retriever (Instructor Embedding Model). Cosine-similarity scores were calculated based on the embedding vectors created by the Instructor Embedding Model to evaluate the relevance of the triples, and then the final triples were retrieved.

4.5. Extracted triples analysis

The extracted triples have a mean of 20081.5795, median of 8537.5, max 553210. This shows even in 2 hop we have a vast amount of context for each question. Figure 4 shows the total number of triples per question.Figure 2 shows the median of the number of triples for each category.

4.6. Feeding the context into the Large Language Model

After extracting the most relevant triples (10-sub documents) in our case, we then proceeded on to feed the triples into the Large Language Model.
We need mainly two types of responses from the LLMs:-

- Word/Phrase Response to the input question.
- Chain of Thought Response to the input question.

5. Large Language Models

For the purpose of our experimentations, we mainly relied upon 2 models, both of which were run via APIs. The details regarding each of the models are listed in the following sections

5.1. Gemini-1.5-Pro

5.1.1 Setup

Gemini-pro-1.5 was obtained via **vertex-ai API**, however due to extensive knowledge base of Gemini-pro-1.5, special prompts were engineered so that it focusses more on the context provided rather its own inherent knowledge base.

5.1.2 Generating Objective Phrase Response

The input prompt is of the form:-
"Give a phrase to answer to the questions given below using only the Context given for each question. Also note if the answer is numeric, for example: 5, then respond as 5 and not "Five".
Some examples:
Context-0:A is father of B
B is father of C
Question-0:Is A grandfather of C?
Context-1:A is father of C
B is father of C
Question-1: Is A uncle of C?
Answer-0:Yes
Answer-1:No

Context-{question_id}:{context}
Question-{question_id}:{question}
Always answer all the questions as "Answer-id" where "id" is provided in the question and context statement."

The text highlighted is repeated 50 times; that is, 50 questions are fed at a time to Gemini as it has a context size of 1 Million tokens so this is done to speed up the process of response generation, and the responses are stored in text-files for different question ranges.
The response generated follow the following format:-
Answer-{id}:Answer
The response text files are then saved and then converted into CSV format for further analysis.

5.1.3 Generating Chain of Thought Response

The input prompt is of the form:-

Input Prompt
Answer the questions given below using only the Context given for each question.
Some examples:
Context-0: A baker needs to decide how many cupcakes to bake for a party. They know there will be 10 children and 5 adults attending.
Question-0: How many cupcakes should the baker prepare in total?
Answer-0: The baker should prepare 15 cupcakes in total. (Reasoning: There are 10 children and 5 adults, so 10 + 5 = 15 people attending the party.)
Question-1: If the baker decides to bake 2 cupcakes per person, how many cupcakes will they need?
*Answer-1:The baker will need 20 cupcakes in total. (Reasoning: There are 15 people attending (from the previous question), and they plan to bake 2 cupcakes per person, so 15 * 2 = 30 cupcakes.)*

Context-{question_id}:{context}
Question-{question_id}:{question}
Give the chain of thought that helped you answer the given question from the provided context only.
Give the output in the format: Answer-id: Answer.

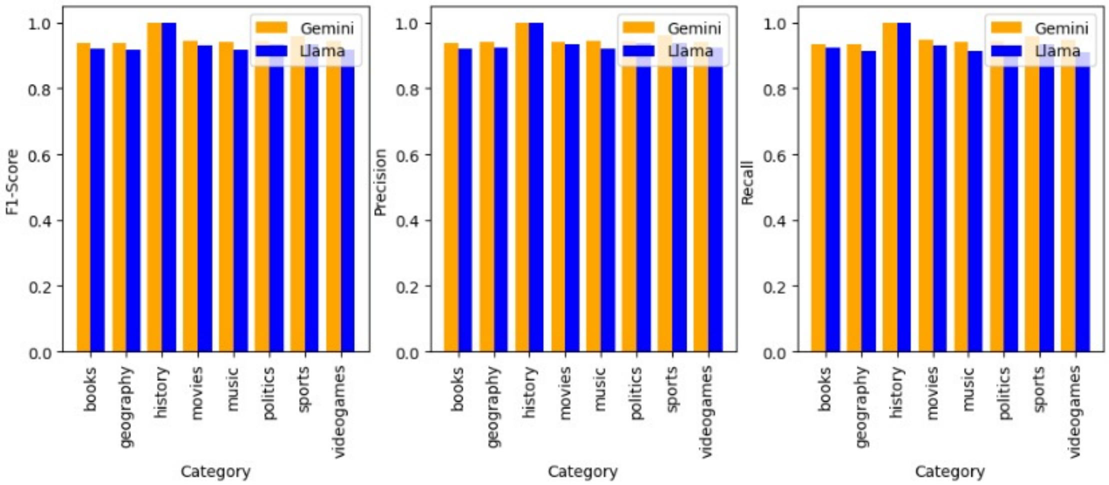

Figure 3. Bert score for each category

Reasoning : Reasoning only for all the questions."

The Context and Question line is repeated 10 times; that is, 10 questions are fed at a time to Gemini as it has a context size of 1 Million tokens so this is done to speed up the process of response generation, and the responses are stored in text-files for different question ranges.

Output Format

***Answer-id**: Answer to the question. Reasoning(Reasoning for the answer)*

5.2. Llama 3

5.2.1 setup

The model was run via **groq API** which provides Llama 3. This allowed us to get faster results without the use of our own GPU. The inputs were engineered such that to prevent context overflow as well as rate limiting of the API.

5.2.2 Generating Objective Phrase Response

The input prompt is of the form:-

system_prompt:
Give a phrase to answer to the questions given below using only the Context given for each question. Also note if the answer is numeric, for example: 5, then respond as 5 and not Five. Do not give any explanation.
Write outputs in JSON in schema: {schema}
examples:
Context-0:A is father of B
B is father of C
Question-0:Is A grandfather of C?
Context-1:A is father of C
B is father of C
Question-1: Is A uncle of C?
Answer-0:Yes
Answer-1:No

user_prompt:
Context-{question_id}:{context}
Question-{question_id}:{question}

The system prompt is attached with the user prompt for every question.Each question is passed individually as the context size of Llama 3 is only 8K tokens.The response is generated as a JSON object.

The response generated are of the form :

```
"question": {
    "type": "string",
    "description": "the question asked"
},
"answer": {
    "type": "string",
    "description": "single phrase"
}
```

5.2.3 Generating Chain Of Thought Response

The input prompt is of the form:-

system_prompt:
Give a phrase to answer to the questions given below using only the Context given for each question. Also note if the answer is numeric, for example: 5, then respond as 5 and not Five. Give explanation for your answers .
Write outputs in JSON in schema: {schema}
examples:
Context-0:A is father of B
B is father of C
Question-0:Is A grandfather of C?
Context-1:A is father of C
B is father of C
Question-1: Is A uncle of C?
Answer-0:Yes
Answer-1:No

user_prompt:
Context-{question_id}:{context}
Question-{question_id}:{question}

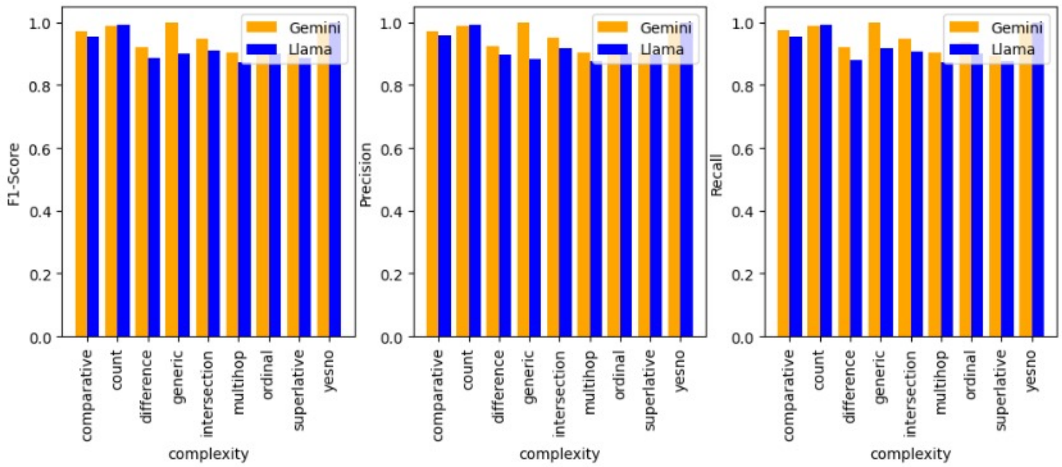

Figure 4. Bert score for each Complexity

The response generated are of the form :
”question”: {
    ”type”: ”string”,
    ”description”: ”the question asked”
},
”answer”: {
    ”type”: ”string”,
    ”description”: ”single phrase”
}
”reasoning”:{
    ”type”: ”string”,
    ”description”: ”reasoning for the answer”
}

6. Results

We ran our experiments for 2000 questions, which is exactly half the questions in the test-split of the Mintaka Dataset. The metrics obtained have been discussed in Table 1.

Metric	Gemini	Llama
Accuracy Score	0.533	0.3995
Bert Score : F1	0.9443	0.9251
Bert Score : Precision	0.9449	0.9292
Bert Score : Recall	0.9447	0.9223
Meteor Score	0.4339	0.3140

Table 1. Metrics of the LLMs

7. Chain of Thought

7.1. Gemini-Pro

Question: How many children did Donald Trump have?

- Answer: Donald Trump has 5 children.
- Reasoning:The text states ”Donald Trump number of children 5”
- Category:Count

Question: Which actress was nominated for but did not win a Best Actress Academy Award for the 2000 movie American Beauty?

- Answer: Annette Bening.
- Reasoning: We are told that the movie *American Beauty* was nominated for a Best Actress Academy Award. We know that Annette Bening was nominated for this award for her role in the film. Since we are also told that *American Beauty* won Best Picture, we can infer that Bening did not win Best Actress.
- Category:difference

Question: What state is the author of Misery from?

- Answer: Maine.
- Reasoning: The context indicates that Stephen King, the author of Misery, currently resides in Bangor, Maine.
- Category:multihop

7.2. Llama 3

{
”question”: ”Which actor starred in Jerry Maguire and was born in Syracuse, NY?”,
”answer”: ”Tom Cruise”,
”reasoning”: ”According to the context, Tom Cruise starred in Jerry Maguire and was born in Syracuse, New York.”
”category”: ”intersection”
}

{
”question”: ”How many children did Donald Trump have?”,
”answer”: ”5”,
”reasoning”: ”According to the context, Donald Trump had 5 children: Donald Trump Jr., Ivanka Trump, Eric Trump, Tiffany Trump, and Barron Trump.”
”category”:”count”
}

{
”question”: ”Is the main hero in Final Fantasy IX named Kuja?”,
”answer”: ”No”,

"reasoning": "The main hero in Final Fantasy IX is Zidane Tribal, not Kuja."
"category": "yesno"
}

8. Individual Contributions

Task	**Member Name**
Researching Ideas and Resources	Both Members
SPARQL Query Formation for triple extraction	Both Members
K-Hop on the Knowledge Graph and verbalisation	Karan Gupta
Semantic Similarity Search	Shivesh Gulati
Llama 3	Karan Gupta
Gemini 1.5-pro	Shivesh Gulati
Dataset Analysis	Karan Gupta
Inference	Shivesh Gulati

Table 2. Contributions Table

9. Conclusion :

From the above analysis, we noted the following points:

- The accuracy score obtained by Gemini-Pro was higher compared to that of Llama-3 however the reason behind this could have been the fact that Gemini-Pro has higher external knowledge.
- The limited context size of Llama-3 $8K$ made it difficult to fit the entire context into the prompt, thus it underperformed on certain questions, due to lack of relevant triples.
- Gemini Pro on the other hand due to its large context window of 1M tokens, is able to perform much better since it has access to the complete context as can be seen from Figure-3 and the metrics.
- The chain of thought responses obtained from both the language models, clearly indicate that if given the right context, both of the models are capable of providing the correct reasoning to a lot of complex problems which are of varying degree of complexity.

References

[1] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. 2023.