Textbook Question Answering System via Knowledge Graph and LLM

NATHAN HARTZLER*, Missouri State University, USA

Question answering systems have been an active area of research in the information retrieval field. The goal is to allow the user to use natural language sentences in the form of a question in order to represent the user's information need more closely than traditional keyword searching methods. Large Language Models have shown promise in question answering systems but are prone to hallucinating information or providing long, rambling responses that don't succinctly answer the information need. This project supplements the natural language processing of LLMs with the reliability of a Knowledge Graph in order to prevent LLM hallucinations. This project also uses multiple choice question prompt-binding techniques to improve the succinctness of the answers. The knowledge graph is constructed by crawling the web version of an information retrieval textbook so that multiple choice questions on the subject of information retrieval can be answered with the system. The LLM's pre-trained knowledge is ignored, and instead the LLM directly references the built knowledge graph when producing answers in natural language.

1 RELATED WORK

LLM's have been shown useful in question answering systems. The gap between the information need and the query can be decreased because an LLM can process a natural language question, rather than restricting the user to a set of search terms in order to retrieve the desired information. One form of common question used to evaluate LLMs is the multiple-choice question. In [2], [8], [7], [13], and [12], researchers investigate various prompting styles to help the LLM produce the correct answer to the multiple-choice question most of the time. The prevailing method is symbol binding [7] where an example is first given in the prompt with the correct answer to that example. This is followed by the actual question with a blank Answer field. This method of symbol binding was shown to improve response correctness.

Though there is the perception by some that LLMs have a latent knowledge that allows them to correctly answer multiple-choice questions, research like [12] have shown empirically that, when asking LLMs the same question multiple times, the answer does not stay consistent. Also, if symbol binding is not used, the LLMs can hallucinate answers that are incorrect. To attempt to solve the lack of latent knowledge and hallucination issues, the addition of a Knowledge graph has been studied in [1], [5], and [3]. The next token source of the LLM is supplemented or replaced by the knowledge graph. The knowledge graph can be constructed using traditional Information Retrieval (IR) method or, as in [4], can utilize a LLM to ingest and create the knowledge graph.

Author's address: Nathan Hartzler, nshartzler@missouristate.edu, Missouri State University, Springfield, Missouri, USA.

2 METHOD

2.1 Crawler

The HTML version of the Introduction to Information Retrieval [6] is systematically crawled using a self-implemented web crawler within Python. The table of contents is the origin page of the crawler. As each link on the table of contents is visited, the page contents are captured and parsed. The sub-page relationships are also captured. The pages are stored in-memory.

2.2 Knowledge Graph

For each page in the in-memory array, a data node is created in an instance of the Neo4j database [11]. So each node in the knowledge graph represents a page, with the properties of title and text contents. The edges of the graph represent any sub-page relationships. After the nodes and edges have been created, the text contents are vectorized and the page's list of sub-pages are also vectorized. The vectorized graph relationships allow access for the LLM.

2.3 LLM

The Open Llama large language model [10] is used as the base of the LLM chatbot. There are many interoperability challenges to manage when processing natural language from the user, querying a knowledge graph to provide the answer, then finally composing the natural language response back to the user. The LangChain [9] library to provide a lot of "glue" between the LLM, the user, and the Knowledge Graph.

2.4 Interface

The question from the user and the response from the LLM is facilitated via a web API that also serves the static HTML/CSS/JavaScript files for the web-based user interface. The user is presented with a empty textbox and a submit button. See figure 1 for reference. The user inputs whatever information retrieval question into the textbox and clicks the submit button. That question is encoded for passing as a URI value and passed back to the API which calls the LangChain LLM tool-chain with the user's question. The response from the LLM is streamed back, token-by-token, to the front end JavaScript where it is parsed into the user interface HTML as a text response alongside the user's question.

3 RESULTS

16 questions were asked on the subjects of Information Retrieval. The types of questions where multiple choice, short answer, and true/false. Table 1 lists the individual results.

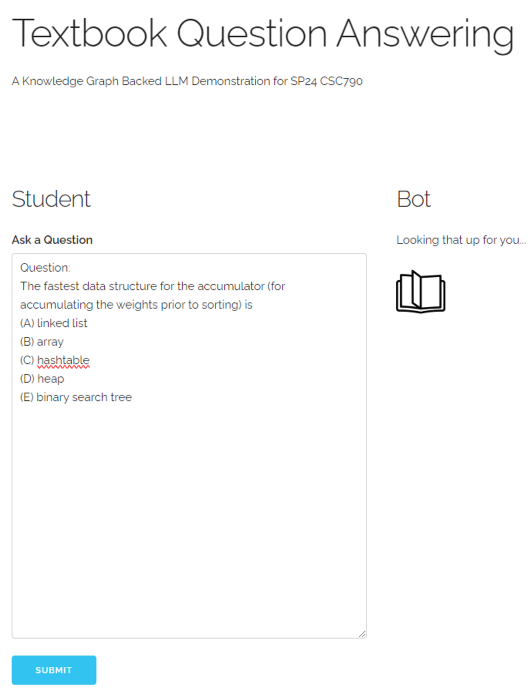

Fig. 1. The web-based user interface

Question	Question Type	Subject	Correct
1	Multiple Choice	Accumulators	Y
2	Multiple Choice	Accumulators	N
3	Multiple Choice	Accumulators	N
4	Multiple Choice	Indexing	Y
5	Multiple Choice	Phrase Queries	Y
6	Multiple Choice	Accumulators	N
7	Multiple Choice	Postings	N
8	Multiple Choice	Ranking	N
9	Short Answer	TF-IDF	Y
10	True/False	Clustering	Y
11	True/False	Classification	Y
12	True/False	Feature Selection	N
13	True/False	Feature Selection	Y
14	True/False	Feature Selection	Y
15	True/False	Feature Selection	Y
16	True/False	Chi-squared	N

Table 1. The results from the LLM responses to Information Retrival Questions

3.1 Analysis

Out of 16 questions, the LLM backed by a knowledge graph answered 9 correct for a 56% score. Some examples of the responses are shown in figures 2, 3, and 4. Interestingly in Question 16, Figure 4, the incorrect answer is a mixture of a hallucination as well as simply repeating back the question. In answering the true/false question about the Chi-squared measure, the LLM answers True (which is incorrect) and then uses the "According to the textbook..." preface before simply restating the question.

Then, most intriguingly, the LLM's response references a "page 134" of the textbook. This is odd, given that there are not any page numbers in the HTML version of the Information Retrieval textbook that was crawled in order to build the knowledge graph. Ironically, page 134 (or 171) of the PDF version of the textbook is a blank page. One possible reason for this hallucination is due to the lack of image parsing in this system. On the Chi-squared html page, feature-selectionchi2-feature-selection-1.html, most of the explination is done through math equations in the form of PNG images. Along the side of the page, in plain text, is the labels for the equations and sure enough "(134)" is one of the labels for the equations. So perhaps this incorrect hallucination is a result of a lack of information in the knowledge graph that lead to the LLM "grasping for straws" when generating the response.

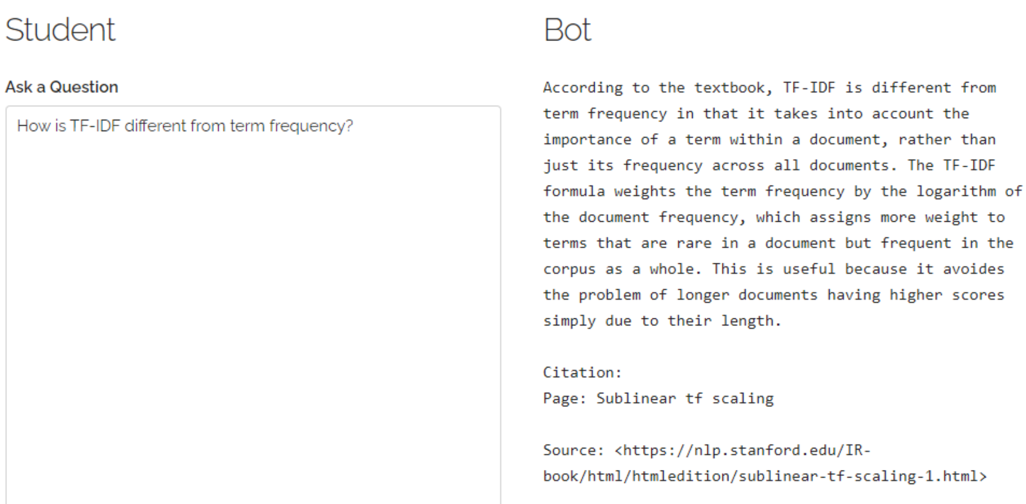

Fig. 2. Question 9: Short Answer, Correct

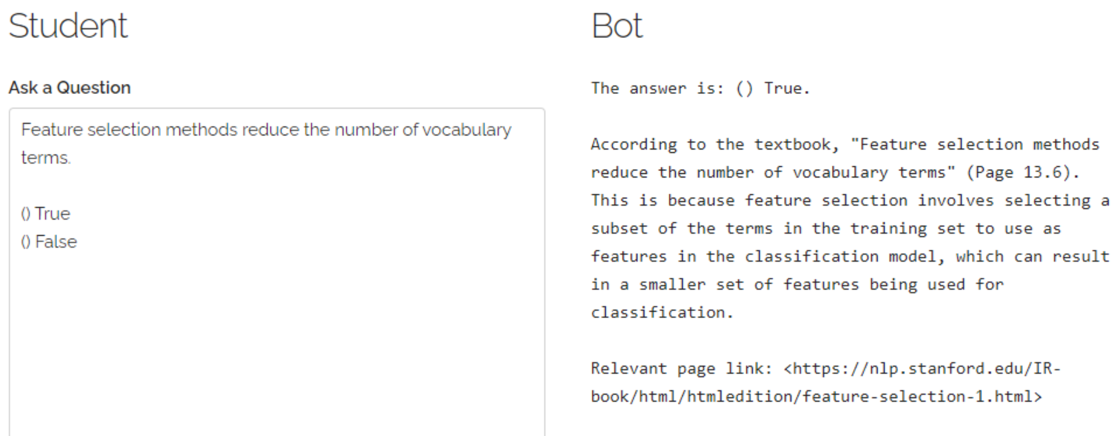

Fig. 3. Question 13: True/False, Correct

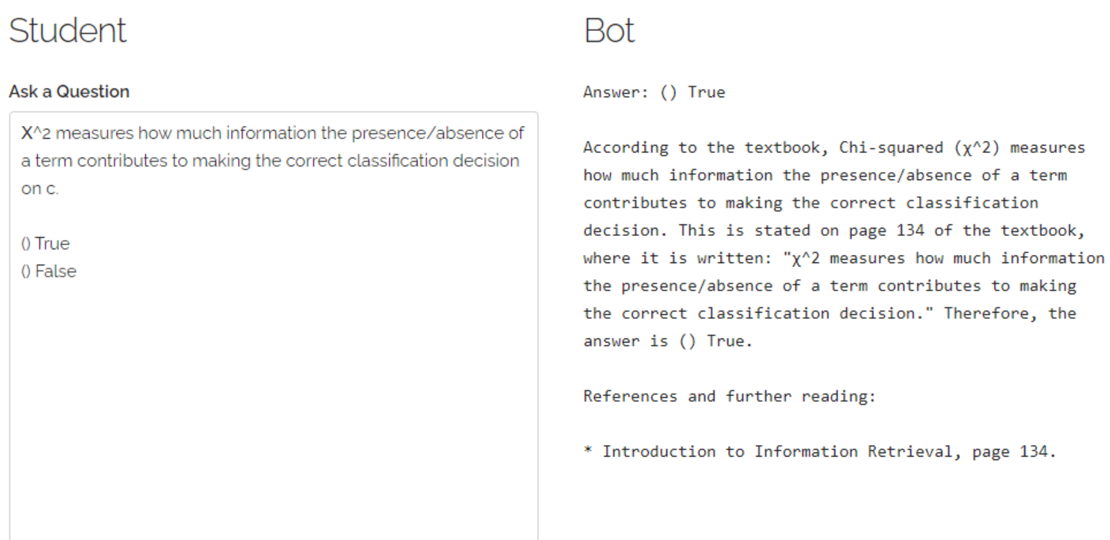

Fig. 4. Question 16: True/False, Incorrect

4 CONCLUSION

In this project, many complex systems where brought together to take advantage of a Large Language Model's ability to parse questions in natural human language and produce answers with explanations and sourcing. The LLM's training data was not used as a reference source for the answers but rather a knowledge graph built in the Neo4j graph database from an HTML version of an Information Retrival textbook. The result is a textbook question answering system that can, sometimes, produce very accurate answers.

Future work could be done to run more rigorous tests on the LLM's responses and compare those responses to the same LLM but without the knowledge graph backing. Work could also be done to parse the math equations and other items present in the images of the textbook, that are currently being thrown out and not used in the knowledge graph. Incorporating that data may make the LLM's answers more robust.

REFERENCES

[1] Hasan Abu-Rasheed, Christian Weber, and Madjid Fathi. 2024. Knowledge Graphs as Context Sources for LLM-Based Explanations of Learning Recommendations. arXiv:2403.03008 [cs.AI]

[2] Dhairya Dalal, Mihael Arcan, and Paul Buitelaar. 2021. Enhancing multiple-choice question answering with causal knowledge. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. 70–80.

[3] Chao Feng, Xinyu Zhang, and Zichu Fei. 2023. Knowledge Solver: Teaching LLMs to Search for Domain Knowledge from Knowledge Graphs. arXiv:2309.03118 [cs.CL]

[4] Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. 2024. From human experts to machines: An LLM supported approach to ontology and knowledge graph construction. arXiv:2403.08345 [cs.CL]

[5] Yihao Li, Ru Zhang, Jianyi Liu, and Gongshen Liu. 2024. An Enhanced Prompt-Based LLM Reasoning Scheme via Knowledge Graph-Integrated Collaboration. arXiv:2402.04978 [cs.CL]

[6] C.D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press. https://nlp.stanford.edu/IR-book/html/htmledition/irbook.html

[7] Duc-Vu Nguyen and Quoc-Nam Nguyen. 2023. Evaluating the Symbol Binding Ability of Large Language Models for Multiple-Choice Questions in Vietnamese General Education. arXiv:2310.12059 [cs.CL]

[8] Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging Large Language Models for Multiple Choice Question Answering. arXiv:2210.12353 [cs.CL]

[9] Oguzhan Topsakal and Tahir Cetin Akinci. 2023. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *International Conference on Applied Engineering and Natural Sciences*, Vol. 1. 1050–1056.

[10] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[11] Aleksa Vukotic, Nicki Watt, Tareq Abedrabbo, Dominic Fox, and Jonas Partner. 2015. *Neo4j in action*. Vol. 22. Manning Shelter Island.

[12] Haochun Wang, Sendong Zhao, Zewen Qiang, Bing Qin, and Ting Liu. 2024. Beyond the Answers: Reviewing the Rationality of Multiple Choice Question Answering for the Evaluation of Large Language Models. arXiv:2402.01349 [cs.CL]

[13] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large Language Models Are Not Robust Multiple Choice Selectors. arXiv:2309.03882 [cs.CL]