Image Processing, Electronics and Computers
L. Trajkovic et al. (Eds.)

doi:10.3233/ATDE240503

Designing an Interpretable Question Answering System for Vertical Domains Based on Large Language Model and Knowledge Graph

Xiaobin HUANG[a,1] and Chin Soon KU[b]
a Xiamen University of Technology, Xiamen, China
b Department of Computer Science, Universiti Tunku Abdul Rahman, Kampar 31900, Malaysia

Abstract. Given the low interpretability of large language models (LLMs) due to their extensive parameters and intricate features, this study aims to enhance the understandability and interpretability of automatic QA systems powered by LLMs, thereby addressing a critical gap in the field. To achieve this, we introduce an interpretable architecture for a domain-specific LLM-based question-answering (QA) system. The research decomposes the QA system into six modules: operation recognition, intent recognition, normalization, triplet structured data conversion, knowledge graph querying, and query result processing. Through this approach, the input and output of each module in the QA system are human-readable text data, enhancing the interpretability of the QA system's processing. The use of knowledge graph data increases the credibility of the answers provided by the QA system. The QA system architecture proposed in this study attempts to integrate the powerful natural language understanding capabilities of large language models with the data querying capacity of knowledge graphs, offering a reference for addressing the issue of low interpretability in automatic QA systems based on large language models (LLMs).

Keywords. Large language model, knowledge graph, question answering system, interpretable

1. Introduction

The Question Answering (QA) system automates responses to natural language queries from users, representing a convergence of information retrieval and natural language processing research. This system leverages fundamental technologies such as Natural Language Processing (NLP), machine learning [1], and knowledge graphs [2]. The advent of large language models has revolutionized this domain by exhibiting robust language comprehension and generation capabilities through extensive pre-training on vast datasets, often surpassing human-level performance in specific contexts [3,4]. Despite their remarkable achievements across various industrial sectors, large models are not without limitations. Operating on deep learning principles, these models remain opaque "black boxes" during training, leading to concerns regarding their interpretability. While large

[1] Corresponding Author: Xiaobin HUANG, moonhxb@qq.com.

models demonstrate interactive prowess in general domains, their responses exhibit significant unreliability and inconsistency when tested in standardized settings, potentially disseminating inaccurate information, particularly in critical fields like medicine or law [5,6]. Knowledge graphs offer structured knowledge representation and data integration functionalities, complementing the natural language processing and reasoning capabilities of large models to enhance overall utility [7,8].

Question answering systems based on large language models and knowledge graphs possess distinct strengths and weaknesses. Hu and colleagues' research [9] proposed a question answering system for wind turbine assembly processes that integrates multimodal knowledge graphs and large language models, necessitating domain-specific data for model refinement. Fan and colleagues' study [10] developed a retrieval question answering system based on knowledge graphs in a Pipeline approach, also requiring domain data for model optimization. Zhang and colleagues [11] designed a question answering system using LLM and knowledge graphs, where the processing workflow relies on domain knowledge bases for supplementary information. Conversely, the medical field question answering system by Yang and others [12] relies on entity recognition outcomes to discern query intent, potentially overlooking contextual cues and reducing intent recognition accuracy. Qin and colleagues' research [13] utilized local knowledge base vectorization and index construction for similarity matching with queries, enhancing domain relevance in answers but struggling with accuracy and interpretability assurance.

In light of these considerations, this study introduces an innovative interpretable vertical domain question answering system architecture design scheme. The research aims to seamlessly integrate large language models and knowledge graphs in a novel manner. This architecture preserves the high performance of large language models in language understanding and content generation, enhances the system's natural language comprehension, broadens the knowledge coverage, and upholds the interpretability and real-time update capabilities of knowledge graphs. By synergistically harnessing the strengths and capabilities of both technical frameworks, the study ultimately seeks to enhance the interpretability of large language model question answering systems.

2. Large Language Model and Knowledge Graph

In the realm of artificial intelligence's rapid advancement, large language models have emerged as pivotal tools driving technological progress. Exemplified by the GPT series, these models harness deep learning techniques and extensive data training to comprehend and generate text at a level approaching human proficiency, showcasing remarkable language understanding and generation capabilities. They play a crucial role in various domains such as natural language processing, machine translation, and content creation. Concurrently, knowledge graphs, serving as a structured method for knowledge representation, furnish machines with a wealth of information by establishing intricate networks of relationships among entities. This approach substantially bolsters tasks like information retrieval, knowledge exploration, and intelligent reasoning, significantly augmenting machines' comprehension and decision-making abilities. Within QA systems, large language models excel in grasping user queries' intents and crafting coherent responses, while knowledge graphs furnish these responses with precise and extensive factual foundations.

2.1. Large Language Model

Large language models exhibit formidable language understanding and generation capabilities, often surpassing human performance in tasks such as language comprehension, reasoning, and dialogue generation. Nevertheless, these models confront several pressing challenges that necessitate resolution:

(1) Risk of factual inconsistency: Large language models lack mechanisms to verify the accuracy of information they generate, potentially leading to the dissemination of erroneous or false data. This poses significant risks to the widespread deployment of such systems.

(2) Limited model analysis and interpretability" Deeply rooted in complex neural networks, large language models' decision-making processes are inherently opaque and challenging to elucidate.

(3) Temporal constraints: The training data utilized by large language models frequently lags behind real-time information, impeding their ability to address time-sensitive queries. Moreover, the high costs associated with model training and fine-tuning hamper the pace of model updates.

(4) Expensive training and fine-tuning: Notably, the training process for models like GPT-3 demands substantial computational resources, involving over 10,000 GPU cores and datasets comprising hundreds of gigabytes. This training cycle spans weeks to months and necessitates iterative version updates to refine and adjust parameters. Fine-tuning operations typically require multi-gigabyte datasets and can extend over days to weeks. The advent of GPT-4 has further escalated model complexity, necessitating a proportional increase in hardware and data resources.

2.2. Knowledge Graph

The knowledge graph is a collection of interconnected entities and their attributes [14]. Rooted in the semantic web and ontology, it serves as a semantic network depicting relationships among entities. Leveraging principles from mathematics and information science, knowledge graphs visually represent resources and their carriers, finding utility in domains like question answering and recommendation systems. Depending on the application domain, knowledge graphs can be categorized into general and vertical fields. Vertical field knowledge graphs are tailored to specific industry data, boasting superior knowledge quality and accuracy despite their smaller scale. In contrast, general field knowledge graphs offer broader coverage and larger, more automated scales [15]. Knowledge graphs present several notable advantages:

(1) Real-time update capability: The knowledge graph not only houses data but also encapsulates intricate data relationships. Storage and updates within knowledge graphs predominantly rely on graph databases, such as the prevalent Neo4j graph database. Data can be dynamically added, updated, or removed through the database API, and can also be integrated via triggers and external data sources to establish a comprehensive automated data update process, facilitating real-time data tracking and updates.

(2) Interpretability: Knowledge graphs, especially in domains demanding exceptional accuracy like healthcare and engineering [16], often necessitate manual construction. This manual curation results in challenges such as high annotation costs, limited data volumes, and prolonged construction cycles.

However, knowledge graphs also exhibit inherent limitations:

(1) Challenges in processing complex semantic information: Complex questions, rich in semantic content, pose challenges as each component of the question can significantly influence triple selection. Consequently, effectively managing complex semantic information has emerged as a major research hurdle.

(2) Insufficiency in natural language generalization capability: Given the ever-evolving nature of natural language expressions, a single piece of knowledge may manifest in various linguistic forms. Accurately mapping these diverse natural language expressions to specific entities within the knowledge graph remains a focal point and challenge in research.

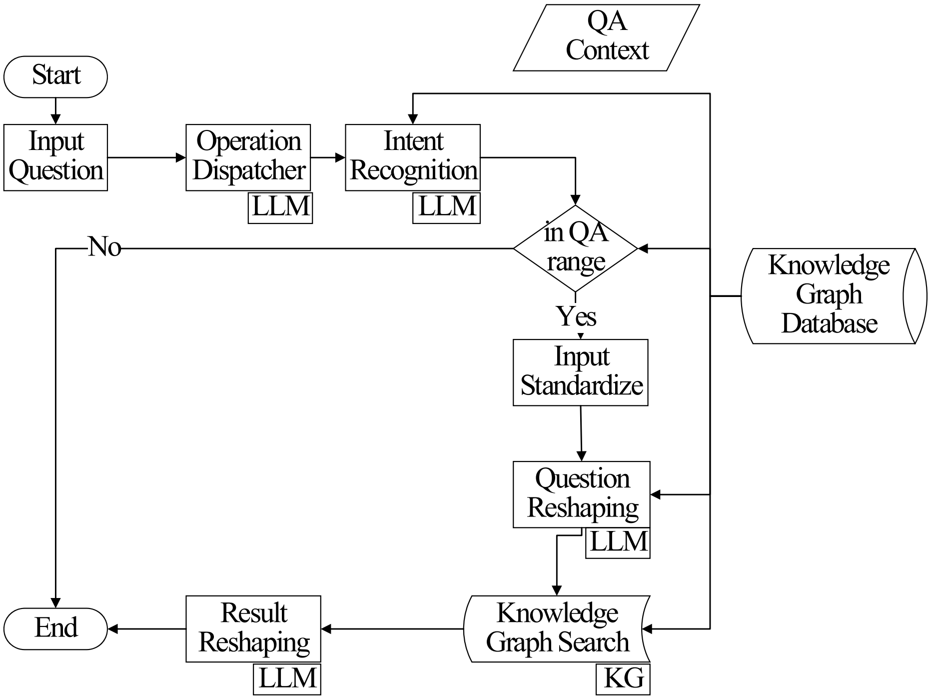

Figure 1. System architecture diagram

3. Interpretable Large Model Question Answering System in Vertical Fields

QA systems leveraging large language models and knowledge graphs exhibit distinctive strengths and limitations. In light of this, this study proposes an innovative and interpretable vertical domain question answering system architecture design scheme. The research endeavors to seamlessly integrate large language models and knowledge graphs in a novel manner[17]. Within this framework, the superior performance of large language models in natural language comprehension and content generation is preserved, enhancing the system's natural language understanding capabilities. Simultaneously, the knowledge coverage of the question answering system is expanded. Furthermore, the architecture upholds the interpretability and real-time update capabilities inherent in knowledge graphs, thereby synergizing the strengths and functionalities of both technical frameworks.

Ultimately, the objective is to enhance the interpretability of the large language model question answering system.

3.1. System Architecture and Module Segmentation

The QA system design outlined in this study comprises six modules. Figure 1 illustrates the interplay between each module and its function within the system. Modules denoted by "LLM" in the lower right corner predominantly leverage the large language model, while modules labeled "KG" in the lower right corner primarily rely on the knowledge graph.

3.2. Module Functions and Implementation.

Drawing from this research, the comprehensive large oracle model QA system is segmented into six modules. Each module's technical implementation operates independently of the others. The output of one module serves as the input for the subsequent module, and the sequential execution of all modules culminates in the completion of the QA process.

This study is anchored in the GPT-3 large language model. While the Knowledge Graph Search module relies on Neo4j's full-text index implementation, the remaining modules are GPT-3 based. Module functionalities are realized through tailored prompts and data inputs unique to each module. Table 1 delineates the function overview of each module, alongside the GPT-3 invocation code shared by each module and the bespoke prompt for each module.

Table 1. Module Description.

Module	Module Introduction	Type	Content
Operation recognition Inten t Recognition I nput Stardardize Question Reshaping R esult Reshaping	The module function is implemented based on GPT3 . It is called through python code and the customized prompt and data of each module are passed in.	core code	import openai # Set API key openai.api_key = "YOUR_API_KEY" # Define prompt containing ${text} prompt = """${text}""" # Call GPT-3.5-turbo model response = openai.Completion.create(engine="text-davinci-003", prompt=prompt, max_tokens=100) # Get the output of the model output = response.choices[0].text.strip() print(output)
Operation Recognition	Identify the operation type and determine whether the input request complies with the application scope of the system, that is, determine whether it is the type of service that the current QA system can provide. For requests that exceed the system's capabilities, the module will filter them at this step and feed back relevant information to the user. The implementation of this function relies on the prompts preset in the large model.	prompt	The question type classification ${cagegory} includes [symptoms, diseases, medical treatment, drugs, body, nutrition, health care, medical knowledge, health knowledge] ${cateogry} is an array containing multiple types. Here is a piece of text ${text} marked by ``` You are a doctor in a clinic. A visiting patient asks you a question ${question}. It is up to you to determine the intention of the question. As long as it is related to one or more of ${cateogry}, ${answer} will be given. ${answer} is equal to the category to which the current ${question} in ${category} belongs. If it does not belong to any category, ${answer} is set to "out of range". Each element in ${category} is a category, and each line in this ${text} is a ${question}. Please reply ${answer} to ${question} in order from top to bottom, And save it into an array ${rs} in the form of ${answer}-->${question}, Finally return and print out the ${rs} array, When outputting, each element in ${rs} is on a separate line. ``` ``

Intent Recognition	Extract the metadata of the vertical domain knowledge graph, and limit the answer knowledge range of the large language model to the target domain through prompts to prevent the system from producing "illusions" that are not related to the vertical domain. For user issues that exceed the scope of the domain, the system will clearly report that they are not supported.	prompt	You are a Chinese medicine expert, Each line in the text ${text} marked with ``` below is a question ${question} Do the following for each question ${question}: 1. What is the purpose of asking this question? 2. Classify the target of the question into one of the types: [medicinal materials, books, quotes, theories, chapters, causes of illness, treatment methods]. If it cannot be classified, set the type to [Unknown] Output data in json format, including two keys: target and type. ``` ` `
Input Stardardize	In vertical domain knowledge graphs, due to the relatively small amount of data, the diversity of natural languages may cause some interference to the retrieval results. In order to ensure the accuracy and efficiency of subsequent processing, we need to normalize the data before processing. This step can be achieved by presetting prompts in the large model.	prompt	You are a text editor. For the following ``` marked text ${text} , Each line in ${text} is a ${question} , and for each ${question}, Do the following: 1. Convert the numbers into Arabic numerals 2. Symbols are uniformly converted into English symbols, 3. Divide a ${question} into one or several short sentences as needed, 4. Remove all characters and spaces in short sentences except Chinese, English, and Arabic numerals. Finally, the processed results are returned. Each row in the results corresponds to a ${question}. Each ${question} processing result in the result is followed by a newline character \n, Only one processing result of ${question} is printed in each line. Your task is to delete text except Chinese, English, Arabic numerals, commas, and periods. All other symbols except Print a newline character at the end of each line ``` ` `
Question Reshaping	Extract the corresponding data type and relationship type information from the metadata of the knowledge graph, integrate these metadata information in the form of Prompt and call the GPT3 model, guiding GPT3 to convert natural language questions into knowledge graph triple data in a specific field	prompt	You are a traditional Chinese medicine expert and an expert in the field of knowledge graph. Each line of the text ${text} marked with ``` below is a question ${question} raised by the patient. The node types ${nodeType} that limit the knowledge graph include: [Causes of disease, Chinese medicine prescriptions, medicinal materials, symptoms, diseases, book chapters] Limit the relationship types ${relType} in the knowledge graph to include: [yes, no, suitable, including] Please do the following for each ${question}: 1. Mark the entities and relationships in the question in the form of knowledge graph triples 2. Limit entity types and relationship types to ${nodeTypes} and ${relType}. For those outside the range, mark them as [Unknown] 3. Save the results of the same ${question} annotation in an array The final result is the annotated knowledge graph data Example output: question:What should I do if I have a stomachache? Triplet: (stomachache:symptom)-[:yes] The results for each ${question} are formatted as: Question: ${question} Triplet: {(entity1:type)-[:relationship type]->(entity2:type)} ``` ` `
Knowledge Graph Search	Query in the knowledge graph according to the query conditions in the form of triples[18]. There are several ways to achieve this functionality. In this study, the full-text indexing function of Neo4j knowledge graph database is used to implement the query function.	Neo4j full text index	cypher_query = f''' CALL db.index.fulltext.queryNodes("all_names_full_text", "{nodeName}") YIELD node, score with node as n,score MATCH (n:{nodeType})--(t:{targetNodeType}) WHERE score>5 return t.name limit 10; '''
Result Reshaping	This module calls the large language model through prompt for result rendering. On the basis of keeping the content of the triplet data unchanged, a large language model is used to transform it into a more natural and suitable form for human understanding.	prompt	You are the customer service staff of a medical platform. Each line in the following text ${text} marked with ```` is an answer to the customer: for the answers 1. If it is "none", please remind: It is beyond the scope of business, please consult me about medical-related issues! 2. If it is "no results", please remind: Sorry, there is no relevant data on the platform, please ask me other questions! 3. For other text content, increase the introduction of prescriptions, explanations and explanations of symptoms, etc., please make the expression more smooth, natural and friendly while maintaining the meaning of the source text. ``` ` `

4. Experimental Process and Result Analysis

4.1. Experiment Procedure

The experiment is conducted using Python 3.8, Neo4j graph database version 5.17.0, GPT-3.5, and Claude3 Opus (Claude 3 Opus being an advanced AI model recently introduced by Anthropic). Initially, a collection of traditional Chinese medicine consultation knowledge graph (TCM-KG), network QA data, and related resources is curated and structured into test data. Subsequently, this data is input into the QA system to capture the output data and resultant QA outcomes generated during the execution of each module. Ultimately, by scrutinizing the intermediate data and final outcomes, the study aims to validate the feasibility and interpretability of the question answering system as proposed.

4.2. Experimental Data

This experiment is based on our own manually annotated TCM prescription knowledge graph (TCM-KG) . On the basis of TCM-KG , network QA data was introduced to verify the filtering ability of this study on questions unrelated to vertical fields. The CMID data set was introduced to verify the ability of this study to limit the content of the QA system's answers to specific vertical fields.

(1) Traditional chinese medicine prescriptions knowledge graph (TCM-KG): This data set contains 20 entity categories, 19 relationship categories, a total of 4430 nodes, and 20255 relationship connections. It is compiled by field experts based on "Treatise on Febrile Diseases" and "Golden Chamber". Summary" A knowledge map in the field of traditional Chinese medicine constructed by manual annotation. TCM-KG only has annotated knowledge graph data and no corresponding question data set. This experiment randomly selects part of the data based on TCM-KG and inputs it into Claude3 Opus as prompt parameters to generate questions.

(2) Internet QA data: 30 system operation tips and domain-independent questions are randomly generated based on Claude3 Opus.

(3) Chinese medical QA intent understanding data set Chinese Medical Intent Dataset (CMID): This data set is used for Chinese medical quality assurance intent understanding tasks. It is provided by Haodafu online and includes diseases, drugs, treatment plans, other four main categories and 36 Secondary classification. This data set contains questions and corresponding answers.

Based on the above three different data sources, a total of 180 experimental data were finally generated , of which 30 came from the CMID data set, 30 came from the network QA, and 120 came from TCM-KG .

4.3. Experimental Process Output Data

This experiment is based on python3.8 and GPT3.5 . Except for the Knowledge Graph Search module, which is based on the full-text indexing function of the neo4j database, other modules are based on the GPT3.5 interface. The prompts and data customized by each module are passed in , and the response of GPT3.5 is received. The result is used as the output of the module. Table 2 shows the output data and final results of each module during the QA process. The module execution order is from top to bottom. The output of the previous module is the input of the next module.

Table 2. Data Processing Process.

Module	Output
Question	What medicinal materials are included in Zeqi Soup?
Operation Recognition	drug
Inten t Recognition	Medicinal materials
I nput Stardardize	What medicinal materials are included in Zeqi Soup?
Question Reshaping	[{'nodeType': 'Chinese medicinal prescription', 'nodeName': 'Zeqi Tang', 'relType': 'Includes', 'targetNodeType': 'Medicine', 'targetNodeName': '[Unknown]'}]
Knowledge Graph Search Cypher	['CALL db.index.fulltext.queryNodes("all_names_full_text", "zeqitang") YIELD node, score with node as n,score MATCH (n:prescription)--(t:medicinal materials) WHERE score>5 return t.name limit 10; ']
Knowledge Graph Search Result	Zeqi, cassia twig, ginseng, skullcap, licorice, white root, ginger, purple ginseng, pinellia
Result Reshaping	Zeqi, cinnamon twig, ginseng, skullcap, licorice, white root, ginger, purple ginseng, pinellia. This is a combination of Chinese medicinal materials that may be used to prepare specific Chinese medicine prescriptions. The medicinal materials have their own effects and functions. , can be used under the guidance of a doctor.

Table 3 shows the output data and QA results of each module during the experimental process, which were analyzed manually and statistically.

Table 3. Experimental Result.

Evaluation indicators	**Operation recognition**	**Intent Recognition**	**Question Reshaping**	**K nowledge Graph Search**	**Final QA Answer**
Accuracy	0.97	0.96	0.92	0.87	0.87

The experimental findings of this study indicate that the accuracy of the QA system achieves 0.87, demonstrating a high level of feasibility. The data generated during the experiment and the final results illustrate our successful restriction of the QA scope within the vertical domain knowledge graph to ensure answer reliability. The output data and final results of the QA process exhibit high interpretability, aiding in error identification and debugging within the QA process. Despite the system being divided into six modules, the ultimate accuracy of the QA system does not surpass the lowest accuracy value among the individual modules. The "Question Reshaping" and "Knowledge Graph Search" modules exhibit the lowest accuracy among the six modules.

Within the QA system, the "Question Reshaping" and "Knowledge Graph Search" modules demonstrate the lowest accuracy levels. This can be attributed to the complexity of their task, involving the conversion of natural language questions into final answers, which is intricate and challenging, hence resulting in lower accuracy. The other modules primarily handle data preparation, processing for these two modules, and initial noise filtering operations, which are comparatively less challenging, leading to higher accuracy. Consequently, these two modules constitute the core components of this research. The "Question Reshaping" module necessitates the identification of entities and relationships highly relevant to the natural language questions, with its performance contingent upon the quality of input from the large language model and natural language questions. The efficacy of the "Knowledge Graph Search" module relies on the accuracy of the "Question Reshaping" module in converting natural language questions and the specific knowledge graph query algorithm. Theoretically, its accuracy does not exceed that of the "Question Reshaping" module.

Based on the experimental outcomes, it is evident that enhancing the overall system accuracy necessitates a focus on optimizing the "Question Reshaping" and "Knowledge

Graph Search" modules to elevate their performance, thereby effectively boosting the overall system accuracy.

In addition to high interpretability, the language model question answering system proposed in this study offers the following advantages:

1) Debuggable system: The input and output data of each module in this architecture are human-readable, enabling swift identification of anomalies or errors for targeted optimization.

2) Data updatability: Through observation of intermediate processing results, errors in the knowledge graph data can be detected during system operation, facilitating performance enhancement through data updates.

3) Rapid domain migration: The vertical domain question answering system designed in this framework primarily relies on general large language models, with domain-specificity limited to the knowledge graph data. Seamless domain migration can be achieved by: replacing the knowledge graph database, synchronously updating stored knowledge graph metadata, and fine-tuning the large language model prompts..

4) High cohesion and low coupling: The system design is not restricted to specific large oracle models. Compliance with interface interaction specifications allows for diverse algorithm implementations within modules, varied large oracle model selections, and flexible data storage and retrieval methods, enabling adaptable configurations tailored to practical development needs.

5. Conclusion

The proposed solution of module segmentation and knowledge graph fusion in this study enhances the interpretability of the large language model QA system in vertical domains. It facilitates the debugging of the QA process, supports data updates, rapid field migration, flexible module optimization, and demonstrates high feasibility.

Due to experimental constraints, this article solely validates the interpretability and feasibility of the research within simple QA scenarios. Future endeavors will focus on exploring and researching the application of this approach in complex and multi-hop problem settings, conducting comprehensive evaluations with various large language models to identify optimal performance in entity and relationship labeling tasks, thereby enhancing the study's efficacy.

Large language models are poised to serve as the foundational cornerstone for artificial intelligence applications across diverse domains. Consequently, enhancing their interpretability stands as a crucial prerequisite. By refining the accuracy of mapping entities and relationships from natural language to knowledge graphs, a viable pathway is established to augment the interpretability of large language models.

References

[1] H. Alami, A. El Mahdaouy, A. Benlahbib et.al, "DAQAS: Deep Arabic Question Answering System based on duplicate question detection and machine reading comprehension," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 8, pp. TBD, 2023.

[2] YZ Zheng, DJ Zhu, HL Wu et al., "Overview on Knowledge Graph Question Answering," Computer Systems & Applications, vol. 31, no. 4, pp. 1-13, 2022 .

[3] HP Zhang, LH Li, CJ Li, "ChatGPT Performance Evaluation on Chinese Language and Risk Measures," Data Analysis and Knowledge Discovery, vol. 7, no. 3, pp. 16-25, March 18, 2023.

[4] G. Polverini and B. Gregorcic, "How understanding large language models can inform the use of ChatGPT in physics education," European Journal of Physics, vol. 45, no. 2, pp. TBD, 2024.
[5] JM Zhang, JB Chen, JX Hu et al., "Research on Integrated Application Methods in the Field of Standards Based on Foundation Models and Knowledge Graphs," China Standardization, vol. 23, no. 8 , pp. 39-46, 2023
[6] L. Zhu, W. Mou, and P. Luo, "Potential of Large Language Models as Tools Against Medical Disinformation," JAMA Internal Medicine, vol. TBD, no. TBD, pp. TBD, 2024.
[7] ZR Chen, X Wang, L Wang et al., "Survey of Open-Domain Knowledge Graph Question Answering," Journal of Frontiers of Computer Science and Technology, vol. 15, no. 10, pp. 1843-1869, 2021
[8] MH Wang, T Yin, HJ Yang et al., "Knowledge Graphs and Large Language Models technology development and application," Cyber Security And Data Governance, vol. 42, no. S1, pp. 126-131, 2023.
[9] ZQ Hu, XY Pan, SJ Wen et al., "Assembly process question answering system of wind turbines combining multi-modal knowledge graphs with LLMs," Journal of Machine Design, vol. 40, no. S2, pp. 20 -26, 2023
[10] J. Fan, H. Ma, and X. Liu, "Research on Intelligent Question-Answering Services for Military Knowledge Graphs Based on Open Source Intelligence in the Era of Digital Wisdom," Data Analysis and Knowledge Discovery, pp. 1-15, Oct. 2023.
[11] H. Zhang, X. Wang, L. Han et.al, "Research on Question Answering System on Joint of Knowledge Graph and Large Language Models," Journal of Frontiers of Computer Science and Technology , vol. 17, no. 10, pp. 2377-2388, 2023.
[12] B. Yang, X. Sun, J. Dang et al., "Named Entity Recognition Method of Large Language Model for Medical Question Answering System," Journal of Frontiers of Computer Science and Technology, vol. 17, No. 10, pp. 2389-2402, 2023.
[13] S. Qin, Z. Zheng, Y. Gu et al., "Exploring and Discussion on the Application of Large Language Models in Construction Engineering," Industrial Construction, vol. 53, no. 09, pp. 162-169, 2023.
[14] Y Belinkov, J Glass. "Analysis methods in neural language processing:a survey, " Trans Assoc Comput Linguist, 2019,7 :49–72
[15] JZ Pan, G Vetere, JM Gomez-Perez et al. "Exploiting Linked Data and Knowledge Graphs in Large Organizations," Berlin, Heidelberg: Springer, 2017
[16] A. Zafar, SK Sahoo, H. Bhardawaj et al., "KI-MAG: A knowledge-infused abstract question answering system in medical domain," Neurocomputing, vol. 571, no. TBD, pp. 127141-TBD, 2024.
[17] T. Andreasen, G. Bordogna, G. De Tré et al, "The power and potentials of Flexible Query Answering Systems: A critical and comprehensive analysis," Data & Knowledge Engineering, vol. 149, no. TBD, pp. 102246-TBD, 2024.
[18] J. Qi, C. Su, Z. Guo et al, "Enhancing SPARQL Query Generation for Knowledge Base Question Answering Systems by Learning to Correct Triplets, " Applied Sciences, vol. 14, no. 4, pp. TBD, 2024.