

Article

CPEQA: A Large Language Model Based Knowledge Base Retrieval System for Chinese Confidentiality Knowledge Question Answering

Jian Cao and Jiuxin Cao *

School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China; 230199050@seu.edu.cn

* Correspondence: jx.cao@seu.edu.cn

Abstract: Large language models (LLMs) have exhibited remarkable performance on various natural language processing (NLP) tasks, particularly in the construction of the intelligent question-answering system. These systems, especially in specialized fields, usually rely on NLP through the retrieval of corpus and answering databases to efficiently provide accurate and concise answers. This paper focuses on the national confidentiality publicity and education field, aiming to address the dilemma of inaccurate knowledge retrieval in this field. Therefore, we design an intelligent confidentiality question-answering system CPEQA by comprehensively utilizing the LLMs platform and information retrieval technique. CPEQA is capable of providing professional answers to questions about Chinese confidentiality publicity and education raised by users. Additionally, we also integrate the conventional database retrieval technique and LLMs into the database query construction, enabling CPEQA to perform real-time queries and data analysis for both single-table and multi-table querying tasks. Through extensive experiments with generated query sentences, we show both methodological comparisons and empirical evaluations of CPEQA's performance. Experimental results indicate that CPEQA has achieved competitive results on answering precision, recall rate and other metrics. Finally, we explore the challenges of the CPEQA system associated with these techniques and outline potential avenues for future research in this emerging field.



Citation: Cao, J.; Cao, J. CPEQA: A Large Language Model Based Knowledge Base Retrieval System for Chinese Confidentiality Knowledge Question Answering. *Electronics* **2024**, *13*, 4195. <https://doi.org/10.3390/electronics13214195>

Academic Editor: Arkaitz Zubiaga

Received: 25 August 2024

Revised: 21 September 2024

Accepted: 21 October 2024

Published: 25 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: large language models; knowledge base retrieval; database query; confidentiality publicity and education

1. Introduction

The intelligent question-answering system has attracted the attention of researchers and engineers because of its great potential and business value. With the advent of pre-trained LLMs, the field of Nature Language Process (NLP) has witnessed a shift in methodologies. Currently, it has been widely used in intelligent customer service, smart speakers, intelligent vehicles and many other scenarios. LLMs (e.g., chatGPT) and information retrieval methods are also two mainstream technique roadmaps in the field of intelligent question answering. LLMs' remarkable capabilities like chain-of-thought reasoning [1] stemmed from the increasing number of parameters and training data. Nils et al. [2] proposed the Sentence-Bert method which took the Bert output as sentence embedding. Besides, it introduced the Siamese and triplet network structure to compare the similarity between the sentence representation and the manually labeled similarity, which made the sentence embedding output by BERT suitable for the semantic matching scene.

Based on the NLP, information retrieval adopts a retrieval corpus and question-answering database, which can quickly return accurate and concise answers, rather than the form of listing relevant web page information. Keyword search positioning is one of the commonly used techniques in the information retrieval field. Wordnet [3] defined synonyms and near-synonyms relations between words in semantic queries. Clever search [4]

selected a specific word from the Wordnet ontology base according to the reasoning algorithm and connected it with the query keyword through the AND operator to constrain the query content. Rocha et al. [5] performed graph traversal in the ontology knowledge base to find out the relevant information on users' query content. CIRI [6] provided a visual browser for traditional textual information, where users can browse the tree structure and select query concepts.

However, both methods have their weakness. LLMs struggle with the high cost of model updates and model hallucinations. Model hallucinations [7] refer to LLMs producing content that either deviates from the input context, contradicts the prior generated context, or is inconsistent with established factual knowledge. The information retrieval method poses challenges in significant differences between the corpus paragraph and the users' questions, which is likely to cause matching errors. Therefore, in some specific fields, we should leverage their strengths and overcome the limitations. In this paper, we focus on the national confidentiality publicity and education field. It is directly associated with political security, economic security, national defense security, diplomatic security and other important fields. By strengthening the work of national confidentiality, we can effectively prevent and combat the infiltration, subversion and sabotage activities of hostile forces, and safeguard the long-term peace and stability of the country. Therefore, we design an intelligent question-answering system, CPEQA oriented to confidentiality publicity and education, which not only tackles the sentence-matching dilemma but also improves the answering precision. CPEQA highlights the application of the word vector similarity matching and prompting sentence generation. Despite these applications, we also integrate conventional database retrieval and LLMs into the database query module construction of the CPEQA system. This function helps us realize the real-time data query in the database with the assistance of the LLMs whether single-table query tasks or multi-table query tasks. Experimental data illustrate that this technical roadmap has achieved excellent querying performance when the number of query sentences increases from 500 to 2500.

The conventional intelligent question-answering system based on knowledge base retrieval techniques, including keyword search, semantic retrieval, etc., can return accurate and concise matching answers by retrieving corpus and question answering database. However, due to the difference between the corpus database and users' questions, the answers received by the users may differ from the anticipated answers. Different from this, the utilization of the LLMs can significantly enhance the answer precision by bolstering the system's capacity for generalization. However, LLMs have been criticized for their lack of factual confidential general education knowledge. Specifically, LLMs memorize facts and knowledge contained in the training corpus [8]. Simultaneously, recent studies reveal that LLMs often suffer from model hallucinations and the high iteration costs caused by model fine-tuning [9,10].

To avoid the above issues, we completed the following work:

1. Combining knowledge base retrieval techniques and LLMs, we designed an intelligent question-answering system oriented to Chinese confidentiality publicity and education, called CPEQA. To enable CPEQA to better understand the question, we used a topic-word generation technique that embeds the keywords into the user's question vectorization which involves the provisions of confidentiality law, the confidentiality level, etc. Experimental data reflecting this design would improve the semantic search accuracy significantly.
2. Similarity calculation for unstructured text corpus is another highlight of our work. We used the pretraining model *Bert* [11] to vectorize the keywords and calculate the similarity value with the existing word vectors in the FAISS database. This process can mitigate the dilemma where we sometimes cannot obtain inaccurate similarity calculation results because of the difference between the length of the question text and the retrieved answers.
3. We also integrated the conventional database retrieval technique and LLMs into the database query system construction of CPEQA, enabling real-time query and data

analysis. Building upon the existing system architecture, we extracted the annotation set of the table creation statements, vectorized them, and stored them in the knowledge base alongside the table creation statements. For similarity computation, we compared the users' questions with the annotation set to obtain the corresponding table creation statements, which then helped us derive the prompt sentences. These derived prompt sentences were subsequently input into the LLMs to generate the SQL queries required to retrieve relevant data from the database. Experimental results demonstrated that our methods exhibit excellent performance for both single-table and multi-table query tasks.

We then introduce the structure of this paper. In the Introduction section, we discuss the dilemma of applying LLMs in intelligent question-answering systems and outline our contributions. The Related Work section provides an overview of existing work on intelligent question-answering systems and database query systems. In the Preliminaries on Large Language Models section, we review the preliminaries such as LLMs and prompt engineering that will be utilized in this survey. Additionally, we detail our technical roadmap for the question-answering system CPEQA and database query function in the Methodology section, respectively. The Experiment section presents the relevant experimental data, highlighting the precision values across various metrics. Finally, we discuss potential future applications and conclude our work and in the Future Work section and Conclusion section, respectively.

2. Related Work

2.1. Knowledge Base Question Answering System

In recent years, the development of the intelligent question-answering system has garnered significant attention from researchers. Previous methods for this task can be categorized into knowledge retrieval-based approaches and LLMs-based approaches. The knowledge base defines one kind of database that contains some specific topics and information stored in structured and unstructured forms. Knowledge-Based Question Answering (KBQA) focuses on answering users' questions by utilizing a pre-built knowledge base as the source. Mainstream methods for KBQA include semantic parsing (SP-based) and information retrieval (IR-based). To parse the syntactic and semantic information of complex questions, Luo et al. [12] concatenated syntactic features and local semantic features by encoding directional dependency paths to form a global question representation. Zhu et al. [13] used structure-aware encoders to model the entity or relation context during the query process, which can facilitate the matching process between queries and questions. Miller et al. [14] employed the key-value memory network to implement dynamic instruction updates, representing the compositional semantics of complex problems.

2.2. LLMs-Based Question Answering System

With the widespread application of pre-trained models, LLMs play an increasingly important role in question-answering systems. Luo et al. [15] proposed a generation-retrieval knowledge question-answering framework based on the LLMs, named ChatKBQA. Their work addressed inherent issues in information retrieval, such as low retrieval efficiency and misleading results. Meta AI proposed Retrieval-Augmented Generation [16] that combined the knowledge vector base and approximate nearest neighbor searching algorithm to finish the searching task. This method inputted the relevant retrieval results along with users' questions into the LLMs to obtain the final answers. Semantic searching aims to match query content with the semantic content of documents [17]. In particular, the maximum inner product search ranks documents and matches semantics by maximizing the inner product between the query vectors and the document vectors [18]. However, this algorithm faces its own issues such as excessive computational complexity and high time costs. Therefore, FAISS [19] adopted the approximate nearest neighbor searching algorithms, which divided the original data into different vector spaces through a mapping method and returned the top K nearest neighbors. In addition, in numerous applications involving

professional domain question-answering systems, LLMs still employ a combination of data and fine-tuning methods to acquire expertise in the corresponding field. For example, PMC-LLaMA [20] proposed a pre-trained language model based on biomedical literature. The LLaMA model was meticulously fine-tuned and enriched with medical expertise to augment its proficiency in the domain of medicine. MedPaLM [21] proposed the Multi-MedQA medical question answering benchmark that covered the medical exams, medical research, and consumer medical questions. ChatDoctor [22] used the medical domain knowledge LLaMA model for fine-tuning to obtain a medical chat model. According to 100,000 real-world patient-doctor conversations from online medical consultations, this model added the autonomous knowledge retrieval function.

2.3. LLMs-Based Database Query

Intelligent synthesis of structured query language (SQL) technology refers to the process of automatically transforming users' input into database query statements. This technology helps users avoid manually writing SQL sentences during the database query process. Zhang et al. [23] designed a paradigm-based programming technique and its implementation tool SQLSynthesizer to help end users automate query tasks. Li et al. [24] proposed QFE, which can generate a series of candidate query statements and complete the preliminary filtering sample database and database-result pairs. To optimize the quality and efficiency of SQL synthesis while dealing with large databases and complex queries, Wang et al. [25] designed a SCYTHE system that can form SQL statements from I/O examples. Then, Thakkar et al. [26] presented the EGS algorithm to synthesize relational queries by exploiting patterns in a data structure called the "constant co-occurrence graph" and used this structure to efficiently enumerate candidate programs. SQL statement synthesis based on text input is one of the current research hotspots. The non-paradigm characteristic of natural language is likely to cause uncertainty, which may make the information-extracting process in the text more complex. Cheung et al. [11] first proposed the BRIDGE model based on BERT. BRIDGE utilized BERT for hybrid sequence encoding and generated a decoder combined with a pointer, which achieved advanced results on Spider and WikiSQL datasets after training and fine-tuning. Instead of fine-tuning, OpenAI's GPT series [27] and CodeX model [28] made use of massive datasets to solve the problem of over-dependence and data over-fitting of LLMs, which achieved great performance in translation, question answering and other fields. The emergence of ChatGPT greatly promoted the deep integration and technological change in databases and AI. ChatGPT has been widely applied to automatic code generation and auxiliary tuning. All of this effectively shortens the time required to program SQL statements.

3. Preliminaries on Large Language Models

3.1. Large Language Model

Language models (LMs) are computational models that have the capability to understand and generate human language. Typically, large language models (LLMs) refer to transformer language models that contain hundreds of billions of parameters, which are trained on massive text data [29]. The advent of the transformer architecture [30] marked a pivotal point in developing the LLMs, which concludes the encoder and decoder modules empowered by a self-attention mechanism (Figure 1). This breakthrough led to some state-of-the-art LLMs, such as Chatgpt [31], GPT-4 [32] and PaLM [33]. For auto-regressive language models, such as PaLM [33], given a context sequence X , we want to predict the following token y . The model is trained by maximizing the probability of the given token sequence conditioned on the context, i.e., $P(y|X) = P(y|x_1, x_2, \dots, x_n)$, where $x_1, x_2, \dots, x_n$ are the tokens in the context sequence, and n is the current position. Then, the conditional probability can be decomposed into a product of probabilities at each position. In Equation (1), where N is sequence length. The model predicts each token at each position in an auto-regressive method, generating a complete text sequence.

$$P(y|X) = \prod_{n=1}^N P(y_n|x_1, x_2, \dots, x_{n-1}) \quad (1)$$

Table 1 gives us a high-level view of the popular LLMs. We group early popular Transformer-based pre-language models, based on their neural architectures, into three main categories: encoder-only, decoder-only, and encoder–decoder models. Encoder-only LLMs only use the encoder to encode the sentence and understand the relationships between words, including Bert [11], RoBERTa [34] and RoBERTa [35]. Decoder-only LLMs only adopt the decoder module to generate target output text. The training paradigm for these models is to predict the next word in the sentence. ChatGPT [31] follows the decoder-only architecture. Encoder–decoder LLMs adopt both the encoder and decoder module architecture. The encoder module is responsible for encoding the input sentence into a hidden space, and the decoder is used to generate the target output text.

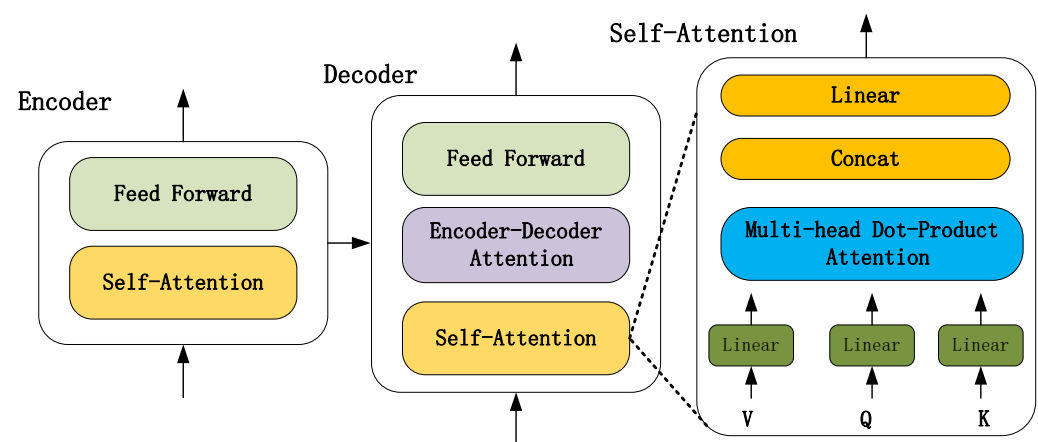


Figure 1. An illustration of the Transformer-based LLMs with self-attention mechanism.

Table 1. An overview of popular LLMs.

Type	Model Name	Release Time	Training Dataset
Encoder-Only	Bert [11]	2018	BookCorpus, Wikipedia
	ALBert	2019	BookCorpus, Wikipedia
	XLNet	2019	BookCorpus, Wikipedia
Decoder-Only	GPT-1 [36]	2018	BookCorpus
	GPT-2 [37]	2019	Reddit outbound
Encoder–decoder	T5(Base)	2019	Common Crawl
	MT5(Base)	2020	New Common Crawl-based dataset
	BART(Base) [38]	2019	Corrupting text
GPT Family	GPT-3 [27]	2020	Common Crawl, WebText2
	GPT-4 [32]	2023	—
	WebGPT [39]	2021	ELI5
PaLM Family	PaLM [33]	2022	Github Code, Web documents
	PaLM-2 [40]	2023	Web documents
	Med-PaLM	2022	HealthWSearchQA
LLaMA Family	LLaMA1	2023	Online Sources
	LLaMA2 [41]	2023	Online Sources
	LongLLaMA	2023	—
	Koala	2023	—
	Alpaca	2023	GPT-3.5

These models can generate content with human-like fluency and comprehend intricate contexts. As LLMs have revolutionized the way how we develop AI algorithms, they pose a significant impact on the research community and domains. In recent years, fine-tuning and reinforcement learning from human feedback has enhanced model conversational and reasoning abilities in diverse settings [31]. Therefore, Wen et al. [42] pointed out that these approaches have optimized the performance and helped produce more advanced capabilities.

3.2. Prompt Engineering

The notion of prompt engineering was initially investigated and popularized in the LLMs [43,44]. Prompt engineering involves strategically designing task-specific instructions to guide model output without altering parameters, i.e., creating and refining the prompt sentences to maximize the effectiveness of the LLMs. Recently, prompt engineering has emerged as a crucial technique for enhancing the capabilities of pre-trained large language models (LLMs). It provides a convenient way to utilize the potential functions of LLMs without the need for fine-tuning. The goal is to enhance the capacity of LLMs (e.g., ChatGPT) in various complex tasks, including question answering, sentiment classification, and common sense reasoning [45]. Liu et al. [46] integrated external knowledge to design better knowledge-enhanced prompts. Dong et al. [47] proposed an approach involving multiple demonstrations to aid LLMs in mastering and executing downstream tasks. To further improve prompt effectiveness, Zhou et al. [48] introduced the Automatic Prompt Engineer (APE), an automatic prompt generation approach. However, enabling advanced reasoning through context learning remains a great difficulty. Chain-of-thoughts (COT) [49] and Reason-and-Act (ReAct) [50] both aimed to solve this problem. In summary, proficiency in prompt engineering leads to a better understanding of the strengths and weaknesses of LLMs. Table 2 demonstrates the taxonomy of prompt engineering techniques, organized by different application domains.

Table 2. Taxonomy of prompt engineering techniques.

Application	Prompt Technique	Comparison Scope		
		LLMs	Dataset	Metrics
Reasoning and Logic	Self-Consistency [51] CoT [49] Auto-CoT [52] Chain of Table [53] Cos [54] LogicCoT [55]	PaLM GPT-4 Llama 2-70B T5-large GPT 3.5 GPT-3	GSM8K Game of 24 GSM8K GSM8K TabFact Arithmetic	Precision Success Rate Precision Rouge BLEU, Rouge Precision
Reduce Hallucination	CoVe [56] ReAct [50] RAG [16] CoN [57] CoK [58]	Llama 65B PaLM-540B RAG-Token Llama 2 GPT 3.5	Wikidata HotpotQA MSMARCO TriviaQA MMLU Physics and Biology	Precision Precision Rouge, BLUE F1 Score Precision
New Tasks without Training Data	Zero-shot [59] Few-shot [27]	GPT-2 GPT-3	Arithmetic, Symbolic NaturalQS, WebQS	Rouge Precision
Code Generation and Execution	SCoT [60] PoT [61] CoC [58] Scratchpad Prompting [62]	ChatGPT, Codex GPT-3.5-turbo text-davinci-003, GPT-3.5-Turbo GPT-3	HumnaEval, MBPP, MBCPP GSM8K, FinQA BIG-Bench Hard MBPP	pass@k Exact Match(EM) Score Precision Precision
User Interaction	Active Prompt [63]	text-davinci-003	Arithmetic, Symbolic	Self-confidence
Fine-Tuning and Optimization	APE [48]	text-davinci-002	BBII, TruthfulQA	Log Probability Execution Accuracy
Knowledge-Based Reasoning and Generation	ART [64]	GPT-3(175B)	BigBench, MMLU	Precision
Optimization and Efficiency	OPRO [65]	PaLM 2-L-IT	GSM8K, BIG-Bench Hard	Precision

4. Methodology

4.1. Question Answering System Architecture and Details

Our system retrieves the potential answers from the knowledge base to construct prompt statements in the vector retrieval method depending on the context-learning ability of the LLMs. Finally, the prompt statement is input into the LLMs to generate the answers. The system architecture consists of two main parts: (1) the construction of the Chinese confidential education knowledge base, and (2) the question-answering module. The construction of the confidential education knowledge base involves text pre-processing of natural language and word vectorization. The word vectors would be stored in the vector database. The question-answering process would conduct the vector similarity matching work and create the prompt sentences. Finally, the system inputs the constructed prompt statements into the LLM to generate the final answers. We summarize the technique roadmap in Figure 2.

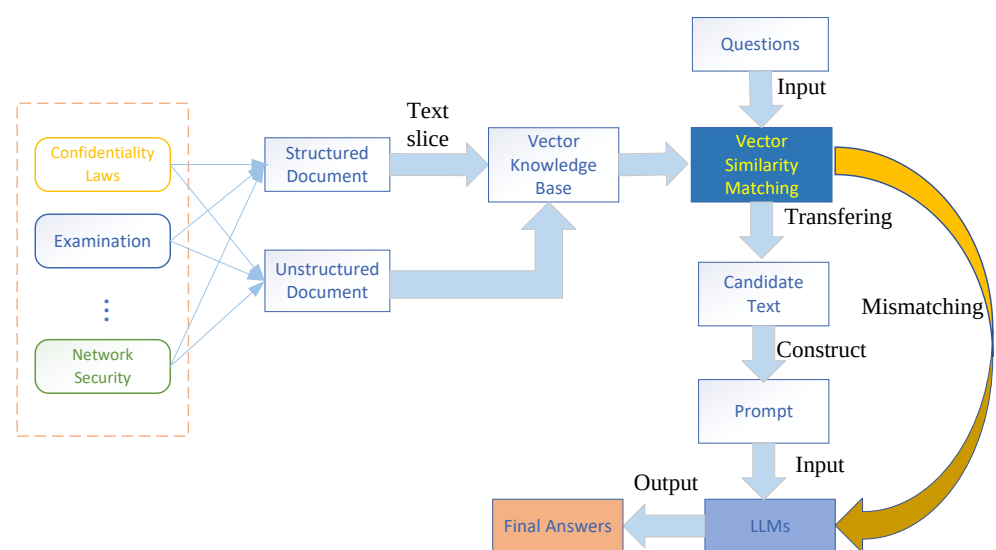


Figure 2. The architecture of the intelligent question answering system CPEQA.

From the Figure 2, we find that when users ask confidential relevant questions to the CPEQA system, such as the latest content of the confidentiality laws, the system will leverage the pre-trained model *Bert* to vectorize the user's questions. In detail, We use the dictionary library corresponding to *Bert* to digitize the questions posed by users. For example, to the question raised by the user: What is the purpose of the enactment of the confidentiality law? We use the dictionary library to convert the words into tokens with corresponding numeric indices, which will be used as the input of the *Bert*. Figure 3 demonstrates the corpus digitization process, where *CLS* and *SEP* are placeholders for the head and tail of the text corpus. In addition, the numbers corresponding to the *CLS* and *SEP* placeholders are 101 and 102, respectively. Then, the *Bert* model will convert the input numbers into word vectors through its Embedding module, which are $E_1, E_2, \dots, E_n$. We feed the word vectors into the transformer encoder to compute the output vector group, $T = [T_1, T_2, \dots, T_n]$. After the average pooling operation for the output vector group, as shown in Equation (2), we can obtain the sentence vector.

$$A = (T_1 + T_2 + \dots + T_n) / n \quad (2)$$

where n is the number of vectors and A is the sentence vector.

After the pre-processing phase, the generated word vectors will be put into the FAISS [19] which is an open-source vector library created by Meta. In the text similarity calculation phase, we compute the cosine similarity value between the vector of the

generated question and the existed vectors in the FAISS database. As shown in Equation (3), where $\vec{Q\_vector}$ is the storage vector in the database and $\vec{Question\_vector}$ is the input question vector.

$$\cos \theta = \frac{\vec{Question\_vector} \bullet \vec{Q\_vector}}{\|\vec{Question\_vector}\| \bullet \|\vec{Q\_vector}\|} \quad (3)$$

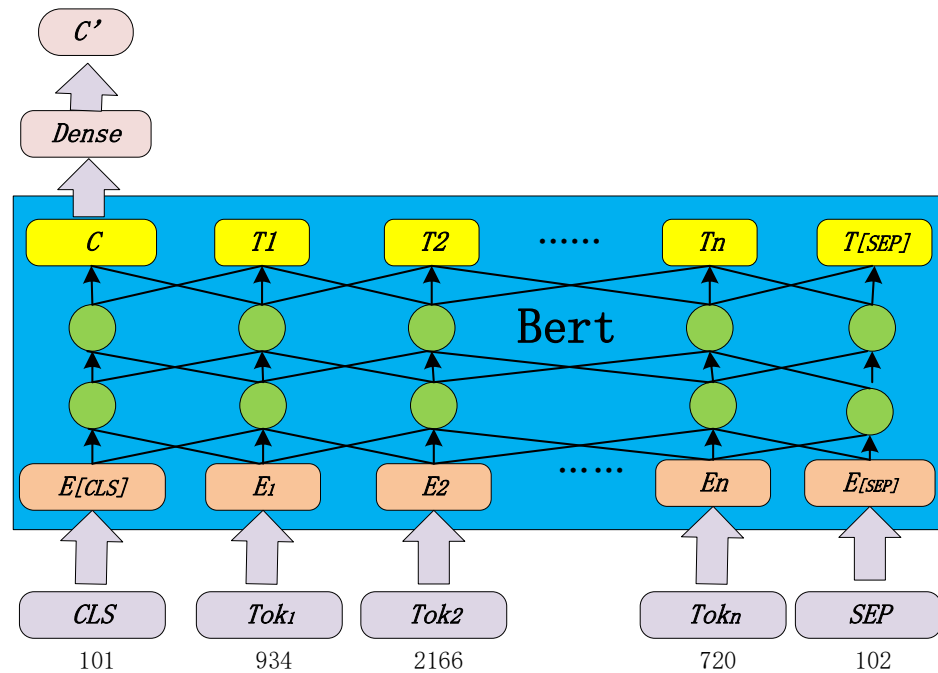


Figure 3. Question pre-Processing in the CPEQA system.

If the cosine similarity value $\cos \theta$ overcomes the threshold, the system will directly return the corresponding answers in the FAISS library to the question-answering interface. Afterward, the system can acquire the candidate text according to the similarity computation results, which is the basis of the generated prompt sentences. As a result, the final answers can be generated after the prompt sentences are transmitted into the LLMs. It is remarkable that the topic-word generation technique is one feature of our system. Our system leverages a topic-word generation model in corpus retrieval to embed keywords into the user's questions. Experimental data in Table 3 will demonstrate the pros of enhancing the semantic information of the user's questions and improving the information retrieval accuracy. The details are shown in Algorithm 1.

Table 3. Key Words Embedding Results.

	Top1_acc	Top5_acc	Top10_acc
No Key Words Embedding	0.634	0.768	0.856
Key Words Embedding	0.743	0.844	0.907

Algorithm 1 LLMs Base Question Answering System Architecture

Input:**unstructured_doc****structure_QA_pairs** = $[(Q_1, A_1), (Q_2, A_2), \dots, (Q_n, A_n)]$ **Output:****Final_answer.**

```

1: Knowledge Base Constrction Process
2: • Unstructured text Pre-processing
3: tokens = word_tokenize(unstructured_doc)
4: slices = slice_tokens(tokens)
5: • Text vectorization and storage
6: vectors = vectorizer.fit_transform(slices)
7: for i, vector  $\in$  enumerate(vectors) do
8:   VectorDatabase.add('slices_' + vector)
9: end for
10: • Structure_QA_pairs stored in database
11: for Q, A  $\in$  Structure_QA_pairs do
12:   Q_vector = vectorizer.transform(Q)
13:   A_vector = vectorizer.transform(A)
14:   VectorDatabase.add(Q_vector)
15:   VectorDatabase.add(A_vector)
16: end for

17: Knowledge Base Question Answering
18: • Initilization
19: model = VectorModel()
20: knowledge_base = KnowledgeBase()
21: • Input Questions and Vectorization
22: Q=Questions
23: Question_vector=model.encode(Q)
24: • Similarity Comparison and candidate_texts Acquisition
25: similar_knowledge = knowledge_base.search_similar(Question_vector, top_k=5)
26: for text  $\in$  similar_knowledge do
27:   candidate_texts = text
28: end for
29: • Construct Prompt and Generate Answers
30: for text  $\in$  candidate_texts do
31:   prompts = Do you know about text
32: end for
33: for prompt  $\in$  prompts do
34:   Final_answer = model.generate_answer(prompt)
35:   print("Prompt : prompt,   Answer : Final_answer")
36: end for

```

4.2. Database Query Architecture and Details

We also apply this technique roadmap in the database query task, Figure 4 demonstrates the whole process.

Database query function comprises two parts: knowledge base construction and data query process. The details are given in Algorithm 2. In the knowledge base construction part, we use Data Definition Language (DDL) to create tables and extract the annotation of table creation statements. To facilitate our vectorization of annotation and DDL statements, we split the statements by words into word fragments. After the vectorization process, all the word vectors would be stored in the knowledge base as comparison objects.

Once the knowledge base is built, the CPEQA performs a similarity comparison between the input questions and the annotation vector set. Based on the similarity comparison results, it returns the corresponding table creation statements. This design enables the

system to establish prompt sentences based on the creation statements. During each query process, these prompt sentences are input into the LLMs, assisting them in generating the query statement. Unlike the intelligent question-answering system, the database query system requires re-inputting the query statements generated by the LLMs into the database to retrieve the final data.

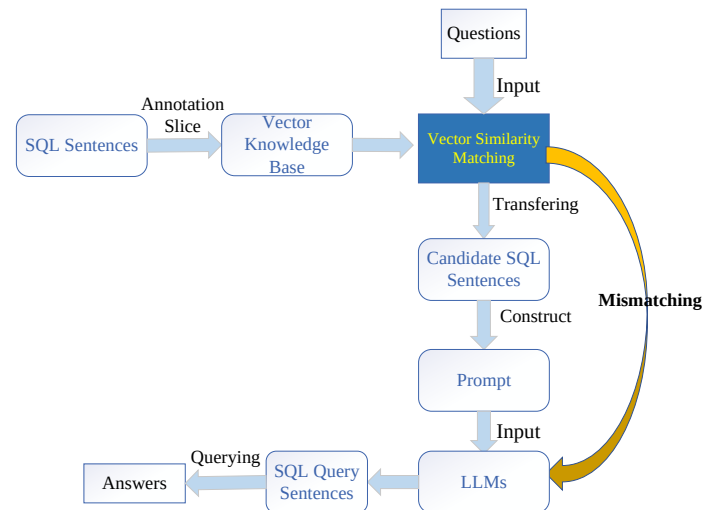


Figure 4. Intelligent Database Query Architecture.

Algorithm 2 LLMs Based Database Query System Architecture

Input:

unstructured_doc: *Question*

Output:

Data

- 1: **Knowledge Base Construction**
 - 2: • **Extract Table Creation Sentences Annotation**
 - 3: *Annotation* = *extract_annotation(DDL_statement)*
 - 4: • **Split the Annotation by word**
 - 5: *segmented_annotation* = *split_annotation(Annotation)*
 - 6: • **Vectorization and Storage in database**
 - 7: *DDL_vector* = *create_vector(DDL_statement)*
 - 8: *Annotation_vector* = *create_vector(segmented_Annotation)*
 - 9: **Data Query Process**
 - 10: • **Vector Similarity Matching between Questions and Annotation**
 - 11: *match_index* = *matching(Question, Annotation_corpus)*
 - 12: • **Acquiring corresponding DDL base annotation**
 - 13: *DDL* = *get DDL from annotation_index(Annotation_corpus, DDL_corpus, match_index)*
 - 14: • **Prompt Construction**
 - 15: *Prompt* = *construct_prompt(DDL)*
 - 16: • **LLMs Generate SQL and Query**
 - 17: *SQL* = *generate_sql(Prompt)*
 - 18: *Data* = *query_database(SQL)*
-

5. Experiments

5.1. Implementation Details

In this section, we design experiments to explore the practical effects of the CPEQA system. Our experiment is divided into three parts. The first part is the keyword generation experiment, which measures the influence of the embedded keywords on answering precision. We select 1000 pieces of testing sentences containing various types of confidential knowledge questions and answers. The evaluation criteria include Top1_accuracy,

Top5_accuracy, and Top10_accuracy. The second part involves testing the precision performance of the CPEQA system. In this part, we conduct comparative experiments between our scheme and fine-tuning condition to evaluate the respective precision trend as the number of query sentences increases. The third part is the LLMs-based database query performance test, which consists of single-table and multi-table queries. We propose between 100 and 2500 pieces of confidentially relevant query sentences to evaluate the experimental results.

Evaluation Metrics We use the open source large language model Tongyi Qianwen from Alibaba in the CPEQA system as a base model and adopt the ROUGE-L method and BLEU method as evaluation metrics.

1. ROUGE-L [66] measures recall by how much the words in reference sentences appear in predictions using Longest Common Subsequence-based statistics.
2. BLEU [67] measures precision by how much the words in predictions appear in reference sentences. BLEU-1(B1), BLEU-2(B2), BLEU-3(B3), and BLEU-4(B4) use 1-gram to 4-gram for calculation, respectively.

5.2. LLMs-Based Question Answering System

5.2.1. Demonstration of Question Answering System

Figure 5 demonstrates the interface of the CPEQA that contains several function areas, including document analysis, text annotation, and knowledge modeling. Additionally, the interface features detectors that display the number of related entities and documents. At the top, there is an intelligent search box capable of performing semantic searches, document searches, entity searches, and other functions.

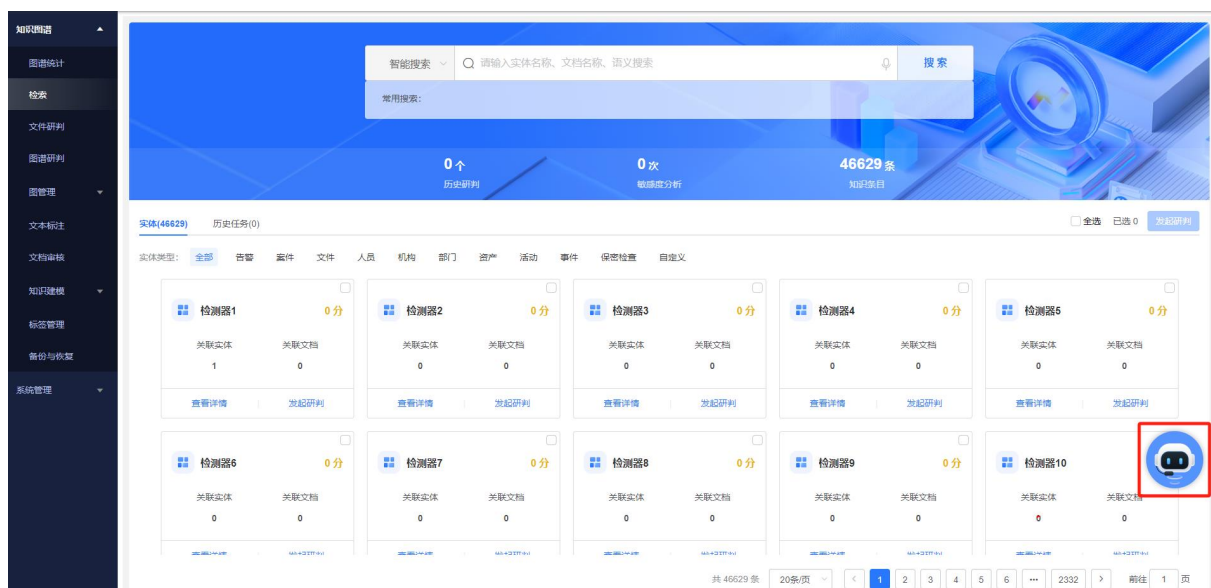


Figure 5. Interface of confidential comprehensive supervision system.

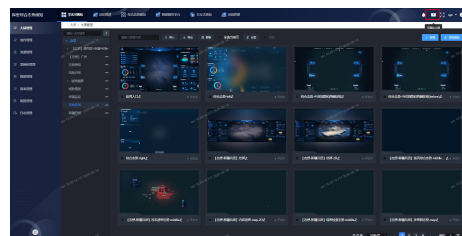
When we click on the chatbot on the right side of the interface, the system presents the CPEQA interaction interface. For mode selection, we choose the LLMs-based mode to conduct the confidentiality question-answering task. Here, we propose three typical questions (Figure 6). For example, we ask the system whether is it necessary for us to back up data off-site in the field of information security. The CPEQA system subsequently provides details of answers that affirm the necessity for remote data backup. Besides, we also ask what actions can we take to protect sensitive information. In response, the system assistant advises us to use complex passwords to encrypt information and a series of other measures to protect sensitive information.



Figure 6. The interface of the question-answering system CPEQA.

5.2.2. Case Study Process

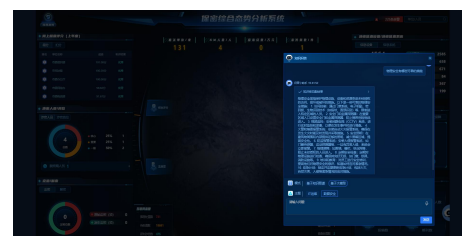
Here, we give one case study about how to use the question-answering system CPEQA. The whole process is shown in Figure 7. Firstly, we should login to the CPEQA system with an account and password. Then we can enter the confidentiality situational awareness interface which contains the resource management, data management, configuration and so on, as shown in Figure 7a. If we want to conduct question answering operation, we can click the visualization screen on the right top of the interface that is circled by the red square. Figure 7b demonstrates the confidentiality comprehensive situation analysis that consists of technical service and guarantee parts. Once clicking the chatbot, we enter the questioning interface Figure 7c which allows us to propose several questions and select the mode including knowledge graph-based and LLMs-based. Finally, after finishing all the settings, the system would return us relevant answers to the proposed questions.



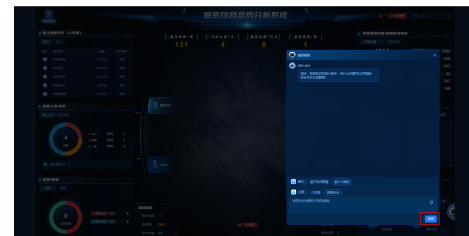
(a) Confidentiality Situational Awareness Interface



(b) Confidentiality Comprehensive Situation Analysis



(c) Questioning



(d) Answering

Figure 7. The case study of question answering process.

5.2.3. Influence of Generated Keywords

We embed the topic word in the user's confidentiality-related questions to enhance the answering precision. For example, if a user asks: "Which units and individuals can be commended in the critical information infrastructure security preserving work?" the system generates the keyword "critical information infrastructure security preserving". We

then conduct a comparative experiment to test the effect of the keyword presence. From the Table 3, we observe that if we do not embed the keywords, such as “critical information infrastructure security preserving” in the questions, the top1_acc only reaches 0.634 which is significantly lower than the case with keyword embedding. When we introduce the keyword in the question sentences, the top1_acc grows to 0.743 and the top10_acc reaches 0.907.

5.2.4. The Performance Question Answering System CPEQA

In this part, we conduct a comparative experiment between our scheme and fine-tuning method. We select the 500, 1000, 2000 to 5000 testing sentences to test the precision of the two methods, and evaluate them from the ROUGE-L and BLEU metrics. Apart from this, we also substitute the Tongyi Qianwen with Chatglm to test the robustness of the system and analyze their respective performance.

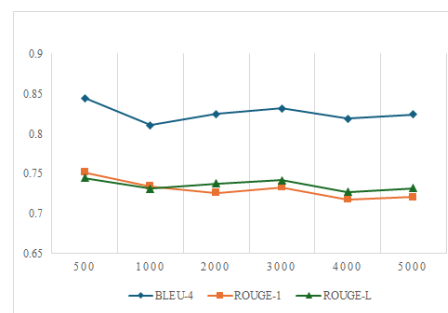
From Tables 4 and 5, we can find the precision of the fine-tuning approach is higher than our scheme. For example, BLUE-4 is only 0.845 with 500 sentences, which is lower than Fine-tuning 0.997. However, as the number of sentences increases from 500 to 5000, the precision value of the fine-tuning scheme declines is more significant than that of our scheme. Figure 8 illustrates that the BLEU-4 value without fine-tuning remains stable between 0.8 and 0.85, while the precision with fine-tuning conditions decreases by 15%. Simultaneously, the decrease in ROUGE-1 is less than 5%, which is significantly lower than the approximately 15% declines observed in the fine-tuning approach.

Table 4. Question Answering System Precision Testing Results without Fine-tuning.

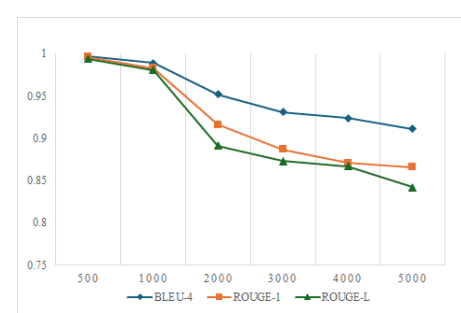
	500	1000	2000
BLUE-4	0.845	0.811	0.825
ROUGE-1	0.752	0.734	0.726
ROUGE-L	0.754	0.731	0.738

Table 5. Question Answering System Precision Testing Results with Fine-tuning.

	500	1000	2000
BLUE-4	0.997	0.989	0.952
ROUGE-1	0.996	0.983	0.916
ROUGE-L	0.994	0.981	0.891



(a) Precision trend without fine-tuning



(b) Precision trend with fine-tuning

Figure 8. Question answering precision trend results between our scheme and fine-tuning.

Table 6 demonstrates us the precision results without fine-tuning when we select Chatglm as the testing model. From Table 6, we can find that the BLUE-4 value decreases as the number of testing sentences increases from 500 to 2000, as do the other two metrics. Simultaneously, compared with the case where we choose the Tongyi Qianwen model, the precision value of it is lower. Figure 9 shows us the comparison results when there

are 500 testing sentences. The BLUE-4 value of Tongyi Qianwen is slightly higher than that of Chatglm. The other two metrics demonstrate similar comparison results. While the testing sentences increase to 1000 and 2000, the comparison results are displayed in Figures 10 and 11. In conclusion, selecting Tongyi Qianwen as the large language model can achieve higher testing precision than Chatglm.

Table 6. CPEQA Precision Testing Results under Chatglm model without Fine-tuning.

	500	1000	2000
BLUE-4	0.831	0.801	0.795
ROUGE-1	0.732	0.726	0.700
ROUGE-L	0.744	0.730	0.735

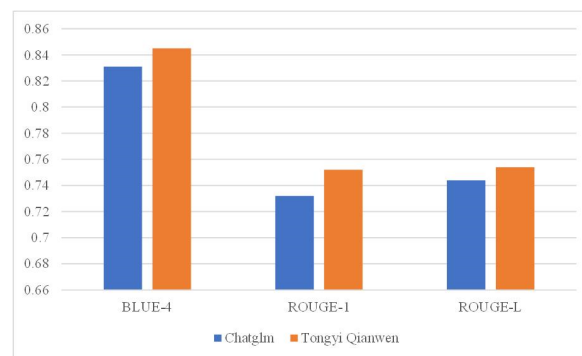


Figure 9. 500 testing sentences.

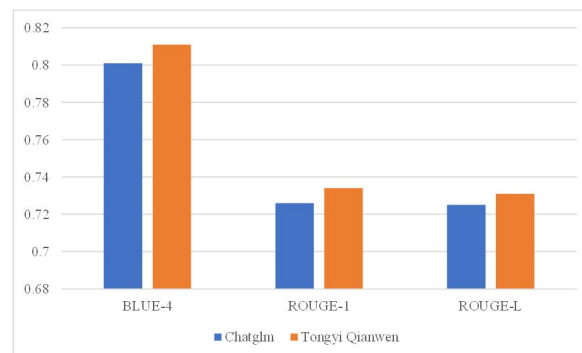


Figure 10. 1000 testing sentences.

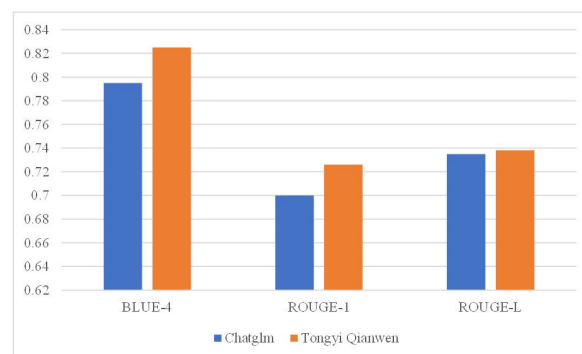


Figure 11. 2000 testing sentences.

5.3. LLMs-Based Database Query

5.3.1. Demonstration of Database Query Interface

Figure 12 shows the interface of the database query system. The bottom of the interface is divided into multiple options such as knowledge graph-based, LLMs-based, and database-based question answering. We choose database-based question answering and propose three example questions. For instance, we ask the latitude and longitude of an enterprise that engaged in confidential work to the question-answering system assistant. Then, the assistant returns the latitude and longitude values.



Figure 12. The interface of the database query.

5.3.2. Database Query System Performance

To demonstrate the effectiveness of the database querying system based on the LLMs, we conduct a testing experiment that covers examples involving 1000 pieces of query data. Besides, we exhibit experimental results of single-table query and multi-table query, respectively, which are shown in Tables 7 and 8. From the performance of Single-Table Query results, we can find that with the increase in the number of query sentences, the answering precision rate fluctuates at 93%. Apart from this, when we only input 100 query sentences, the recall rate exceeds 90%, which is still stable at this level while the query sentences increase. Considering the above metrics, the value of F1-Score in the Single-Table querying process is close to 1, which shows our database query system achieves excellent performance between precision rate and recall rate. For the performance of a multi-table query, we can see that although the precision rate and recall rate have decreased, they are still stable around 87% and 85%, respectively (Figure 13).

Table 7. Single-table query results.

	100	500	1000
Precision	0.948	0.927	0.933
Recall Rate	0.912	0.872	0.895
F1-Score	0.929	0.897	0.914

Table 8. Multi-table query results.

	100	500	1000
Precision	0.881	0.872	865
Recall Rate	0.853	0.824	0.831
F1-Score	0.867	0.847	0.848

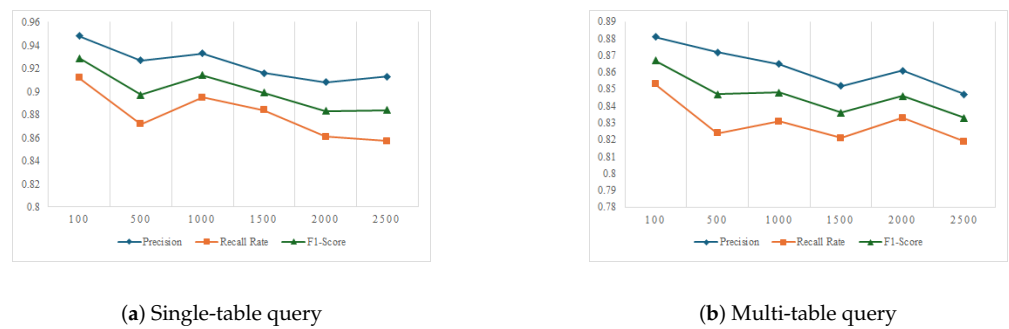


Figure 13. Metrics changing trend between single-table query and multi-table query as testing sentences increasing.

6. Discussion and Future Work

Our summarization inspires us to redesign a wide spectrum of aspects related to evaluation in the field of LLMs. In this section, we present several challenges.

Robustness Evaluation It is crucial for LLMs to maintain robustness against a wide variety of inputs to perform optimally for end-users in the CPEQA and other similar systems, given their extensive integration into Confidentiality publicity and education. For instance, the same prompts but with different grammar and expressions could lead LLMs to generate diverse results, indicating that current LLMs are not robust to the inputs. While there is some prior work on robustness evaluation [68,69], there is much room for advancement, such as including more diverse evaluation sets, and examining more evaluation aspects.

Principled and Trustworthy Evaluation When introducing a question-answering system, it is important to ascertain its integrity and trustworthiness. Therefore, the necessity for trustworthy computing extends to the requirement for reliable question-answering systems as well. This poses a challenging research question that intertwines with measurement theory, probability, and numerous other domains. For example, how can we ensure that dynamic testing truly generates out-of-distribution question examples about the confidentiality field? There is a scarcity of research in this domain, and it is hoped that future work will remedy these deficiencies.

LLM-Enhanced Data Management Developing a robust LLM-enhanced data management system is essential to utilize the full potential of LLMs. There are several main challenges. Firstly, how to effectively utilize data sources (e.g., tabular data and various document formats) to reduce LLM hallucination problems (e.g., through knowledge-augmented answering). Secondly, it is difficult to call LLMs for every request. Therefore, how to accurately interpret the intent behind user requests and capturing the domain knowledge to reduce the iterations with LLMs will be our future research direction.

7. Conclusions

In this paper, to promote confidentiality education, we contribute an LLMs-based intelligent confidentiality knowledge question answering system, CPEQA. It consists of two parts, including knowledge base construction and knowledge base question answering. We have successfully established a confidential knowledge platform and integrated keyword embedding, word vector similarity comparison, and prompt engineering that greatly enhance the precision of knowledge base question answering. Apart from this, CPEQA transforms the candidate text into effective prompts for LLMs, improving the LLMs' performance on knowledge-intensive tasks. In the experiment part, we evaluate the CPEQA system by answering precision, recall rate and other metrics. Experimental results illustrate that the CPEQA system is well compatible with the advantages of LLMs and knowledge base as the data size increases, which enables the users to obtain ideal answers.

Author Contributions: Conceptualization, J.C. (Jian Cao) and J.C. (Jiuxin Cao); methodology, J.C. (Jian Cao); software, J.C. (Jian Cao); validation, J.C. (Jian Cao) and J.C. (Jiuxin Cao); formal analysis, J.C. (Jian Cao); investigation, J.C. (Jian Cao); resources, J.C. (Jian Cao); data curation, J.C. (Jian Cao); writing—original draft preparation, J.C. (Jian Cao); writing—review and editing, J.C. (Jiuxin Cao); supervision, J.C. (Jiuxin Cao). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number No. 62172089.

Data Availability Statement: The data that support the finding of this study are available on request from the corresponding author. They are not publicly available due to privacy or ethical restrictions.

Acknowledgments: First and foremost, we would like to express our sincere gratitude to the Southeast university for their research support, for their generous financial support in the National Natural Science Foundation of China under Grants No. 62172089, which made this research possible. We are deeply indebted to our supervisor, Cao, for his invaluable guidance, insightful comments, and constant encouragement throughout this research. His expertise and dedication have been instrumental in shaping this work. Finally, we are grateful to our colleagues in the laboratory for their support and cooperation. Their input and suggestions have greatly enriched this research.

Conflicts of Interest: The authors declared that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large language models are zero-shot reasoners. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 22199–22213.
2. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
3. Fellbaum, C. *WordNet: An electronic lexical database*; MIT press: Cambridge, MA, USA, 1998.
4. Kruse, P.M.; Naujoks, A.; Rösner, D.; Kunze, M. Clever search: A wordnet based wrapper for internet search engines. *arXiv* **2005**, arXiv:cs/0501086.
5. Rocha, C.; Schwabe, D.; Aragao, M.P. A hybrid approach for searching in the semantic web. In Proceedings of the 13th International Conference on World Wide Web, New York, NY, USA, 17–20 May 2004; pp. 374–383.
6. Airio, E.; Järvelin, K.; Saatsi, P.; Kekäläinen, J.; Suomela, S. Ciri-an ontology-based query interface for text retrieval. In Proceedings of the Web Intelligence: Proceedings of the 11th Finnish Artificial Intelligence Conference, Vantaa, Finland, 2–3 September 2004; Citeseer: Princeton, NJ, USA, 2004.
7. Wang, H.; Shu, K. Explainable claim verification via knowledge-grounded reasoning with large language models. *arXiv* **2023**, arXiv:2310.05253.
8. Petroni, F.; Rocktäschel, T.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A.H.; Riedel, S. Language models as knowledge bases? *arXiv* **2019**, arXiv:1909.01066.
9. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 248:1–248:38. [[CrossRef](#)]
10. Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv* **2023**, arXiv:2302.04023.
11. Cheung, A.; Kamil, S.; Solar-Lezama, A. Bridging the gap between general-purpose and domain-specific compilers with synthesis. In Proceedings of the 1st Summit on Advances in Programming Languages (SNAPL 2015). Schloss-Dagstuhl-Leibniz Zentrum für Informatik, Asilomar, CA, USA, 3–6 May 2015.
12. Luo, K.; Lin, F.; Luo, X.; Zhu, K. Knowledge base question answering via encoding of complex query graphs. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2185–2194.
13. Zhu, S.; Cheng, X.; Su, S. Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing* **2020**, *372*, 64–72. [[CrossRef](#)]
14. Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.H.; Bordes, A.; Weston, J. Key-value memory networks for directly reading documents. *arXiv* **2016**, arXiv:1606.03126.
15. Luo, H.; Tang, Z.; Peng, S.; Guo, Y.; Zhang, W.; Ma, C.; Dong, G.; Song, M.; Lin, W.; Zhu, Y.; et al. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *arXiv* **2023**, arXiv:2310.08975.
16. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
17. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*.

18. Ram, P.; Gray, A.G. Maximum inner-product search using cone trees. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 931–939.
19. Aumüller, M.; Bernhardsson, E.; Faithfull, A. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Inf. Syst.* **2020**, *87*, 101374. [\[CrossRef\]](#)
20. Wu, C.; Zhang, X.; Zhang, Y.; Wang, Y.; Xie, W. Pmc-llama: Further finetuning llama on medical papers. *arXiv* **2023**, arXiv:2304.14454.
21. Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S.S.; Wei, J.; Chung, H.W.; Scales, N.; Tanwani, A.; Cole-Lewis, H.; Pfohl, S.; et al. Large language models encode clinical knowledge. *Nature* **2023**, *620*, 172–180. [\[CrossRef\]](#)
22. Yunxiang, L.; Zihan, L.; Kai, Z.; Ruilong, D.; You, Z. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv* **2023**, arXiv:2303.14070.
23. Zhang, S.; Sun, Y. Automatically synthesizing sql queries from input-output examples. In Proceedings of the 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), Silicon Valley, CA, USA, 11–15 November 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 224–234.
24. Li, H.; Chan, C.Y.; Maier, D. Query from examples: An iterative, data-driven approach to query construction. *Proc. VLDB Endow.* **2015**, *8*, 2158–2169. [\[CrossRef\]](#)
25. Wang, C.; Cheung, A.; Bodik, R. Synthesizing highly expressive SQL queries from input-output examples. In Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, Barcelona, Spain, 18–23 June 2017; pp. 452–466.
26. Thakkar, A.; Naik, A.; Sands, N.; Alur, R.; Naik, M.; Raghothaman, M. Example-guided synthesis of relational queries. In Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual, 20–25 June 2021; pp. 1110–1125.
27. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
28. Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.D.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. *arXiv* **2021**, arXiv:2107.03374.
29. Shanahan, M. Talking about large language models. *Commun. ACM* **2024**, *67*, 68–79. [\[CrossRef\]](#)
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
31. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 27730–27744.
32. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
33. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
34. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
35. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
36. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018, *Preprint, Work in Progress*. Available online: <https://api.semanticscholar.org/CorpusID:49313245> (accessed on 20 October 2024).
37. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
38. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
39. Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv* **2021**, arXiv:2112.09332.
40. Anil, R.; Dai, A.M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. Palm 2 technical report. *arXiv* **2023**, arXiv:2305.10403.
41. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288.
42. Wen, Y.; Wang, Z.; Sun, J. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv* **2023**, arXiv:2308.09729.
43. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [\[CrossRef\]](#)
44. Tonmoy, S.; Zaman, S.; Jain, V.; Rani, A.; Rawte, V.; Chadha, A.; Das, A. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv* **2024**, arXiv:2401.01313.
45. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3580–3599. [\[CrossRef\]](#)

46. Liu, J.; Liu, A.; Lu, X.; Welleck, S.; West, P.; Bras, R.L.; Choi, Y.; Hajishirzi, H. Generated knowledge prompting for commonsense reasoning. *arXiv* **2021**, arXiv:2110.08387.
47. Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; Sui, Z. A survey on in-context learning. *arXiv* **2022**, arXiv:2301.00234.
48. Zhou, Y.; Muresanu, A.I.; Han, Z.; Paster, K.; Pitit, S.; Chan, H.; Ba, J. Large language models are human-level prompt engineers. *arXiv* **2022**, arXiv:2211.01910.
49. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
50. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv* **2022**, arXiv:2210.03629.
51. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv* **2022**, arXiv:2203.11171.
52. Zhang, Z.; Zhang, A.; Li, M.; Smola, A. Automatic chain of thought prompting in large language models. *arXiv* **2022**, arXiv:2210.03493.
53. Wang, Z.; Zhang, H.; Li, C.L.; Eisenschlos, J.M.; Perot, V.; Wang, Z.; Miculicich, L.; Fujii, Y.; Shang, J.; Lee, C.Y.; et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv* **2024**, arXiv:2401.04398.
54. Hu, H.; Lu, H.; Zhang, H.; Song, Y.Z.; Lam, W.; Zhang, Y. Chain-of-symbol prompting elicits planning in large language models. *arXiv* **2023**, arXiv:2305.10276.
55. Zhao, X.; Li, M.; Lu, W.; Weber, C.; Lee, J.H.; Chu, K.; Wermter, S. Enhancing zero-shot chain-of-thought reasoning in large language models through logic. *arXiv* **2023**, arXiv:2309.13339.
56. Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-verification reduces hallucination in large language models. *arXiv* **2023**, arXiv:2309.11495.
57. Yu, W.; Zhang, H.; Pan, X.; Ma, K.; Wang, H.; Yu, D. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv* **2023**, arXiv:2311.09210.
58. Li, X.; Zhao, R.; Chia, Y.K.; Ding, B.; Joty, S.; Poria, S.; Bing, L. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. *arXiv* **2023**, arXiv:2305.13269.
59. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
60. Li, J.; Li, G.; Li, Y.; Jin, Z. Structured chain-of-thought prompting for code generation. *arXiv* **2023**, arXiv:2305.06599.
61. Chen, W.; Ma, X.; Wang, X.; Cohen, W.W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv* **2022**, arXiv:2211.12588.
62. Nye, M.; Andreassen, A.J.; Gur-Ari, G.; Michalewski, H.; Austin, J.; Bieber, D.; Dohan, D.; Lewkowycz, A.; Bosma, M.; Luan, D.; et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv* **2021**, arXiv:2112.00114.
63. Diao, S.; Wang, P.; Lin, Y.; Zhang, T. Active prompting with chain-of-thought for large language models. *arXiv* **2023**, arXiv:2302.12246.
64. Paranjape, B.; Lundberg, S.; Singh, S.; Hajishirzi, H.; Zettlemoyer, L.; Ribeiro, M.T. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv* **2023**, arXiv:2303.09014.
65. Yang, C.; Wang, X.; Lu, Y.; Liu, H.; Le, Q.V.; Zhou, D.; Chen, X. Large language models as optimizers. *arXiv* **2023**, arXiv:2309.03409.
66. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004; pp. 74–81.
67. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
68. Wang, J.; Hu, X.; Hou, W.; Chen, H.; Zheng, R.; Wang, Y.; Yang, L.; Huang, H.; Ye, W.; Geng, X.; et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv* **2023**, arXiv:2302.12095.
69. Zhu, K.; Wang, J.; Zhou, J.; Wang, Z.; Chen, H.; Wang, Y.; Yang, L.; Ye, W.; Zhang, Y.; Gong, N.Z.; et al. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv* **2023**, arXiv:2306.04528.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.