SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Dynamic Domain Knowledge Injection In Large Language Models Using Context-Gated Prompt Engineering

Dr. Abha Kiran Rajpoot[1,2] , Dr. Bal Virdee[3] , Dr. Ashish Khanna[4]

[1]Postdoctoral Fellow, London Metropolitan University, London, United Kingdom
[2]Professor, School of Computer Science & Engineering, Galgotias Univesrity, Greater Noida, UP. India
[3]Professor, London Metropolitan University, London, United Kingdom
[4]Associate Professor, Maharaja Agrasen Institute of Technology (MAIT), Delhi, INDIA

Abstract
Large Language Models (LLMs) have excelled in various NLP tasks, such as natural language understanding, question answering, and knowledge-intensive applications. But it is still difficult to be used in specialized fields, such as biomedicine, because the knowledge acquired in pre-training can be incomplete and outdated, and it may lack the knowledge for specific queries in this field. To include external knowledge, current methods typically rely on fine tuning or prompt engineering or Retrieval-Augmented Generation (RAG). Conventional RAG can include irrelevant or extraneous context in prompts, impacting the accuracy of responses and leading to higher token usage, and fine-tuning is time-consuming and demands significant computing power, with the need to update the model repeatedly. To overcome these challenges, this study introduces Context-Gated Knowledge Injection LLM (CGKI-LLM), a fine-tuning-free domain-specific customization framework. The proposed method decomposes a user query, fetches candidate knowledge, computes the semantic relevance of candidate knowledge, applies a context-gating process to filter out knowledge not related to the user query, and then generates the final prompt. PubMedQA is the main biomedical question-answering set on which experiments are conducted. The proposed framework is compared with a base LLM, conventional prompt engineering, and standard RAG in terms of accuracy, precision, recall, F1 score, hallucination rate and context-related measures. Results demonstrate that the use of CGKI-LLM with standard RAG results in 89.20% accuracy and an F1 score of 88.92% whereas using standard RAG only, the accuracy rate is 84.30%. CGKI-LLM decreases the hallucination rate from 7.20% (standard RAG) to 3.80%. These results show the effectiveness of selective context gating for reducing irrelevant knowledge injection and enhancing domain-specific question answering. The proposed framework offers a practical solution for integrating specialized knowledge into LLMs without altering the model's parameters.

Keywords: Large Language Models, Knowledge Injection, Context Gating, Prompt Engineering, Retrieval-Augmented Generation, PubMedQA.

1. Introduction

Large Language Models (LLMs) are ubiquitous for question answering, information retrieval, text generation, and decision support in different domains. Their capacity to understand and execute natural-language instructions is applicable to healthcare, financial, educational, legal and scientific fields. Even with these features, general-purpose LLMs might not provide accurate answers or even mention them if the query calls for knowledge they were not trained on or inadequately covered by their pre-training data [1]. It has, therefore, become crucial to use external domain knowledge at inference time to adapt LLMs to a specific application without iteratively updating the model parameters.

1.1. Large Language Models and Domain-Specific Knowledge

LLMs are typically trained on massive text corpora, which can include data from a variety of different sources and domains. Models learn linguistic patterns, contextual relationships and factual associations and can reason, via this training. A general-purpose training is provided, in which one model can be used in a wide range of different training tasks; however, the wide range of training does not necessarily imply that the model will gain enough knowledge for all specialized areas [3]. Biomedical applications are especially challenging in that the domain knowledge for medical terminology, scientific evidence, clinical concepts, and research literature must be understood precisely.

While some queries may be answered correctly by a general purpose LLM, some may require a query based on biomedical-specific terminology, evidence or biomedical relationships. Another restriction is the lack of dynamism in the knowledge learnt during the pre-training. After building models, newly published knowledge can't be automatically incorporated into the model parameters. Some of these domain-specific needs can be met by retraining or fine tuning, but there is a higher computational cost, training data requirements, model management, and repeated model updates [4][5].

Customization for a domain is an effort to minimize this dependency by providing relevant information during query processing. Rather than the model having to learn all of the specialized information, outside domain knowledge could be given as contextual evidence. This can be especially useful in biomedical question answering, for instance, where an answer needs to be anchored in the scientific context surrounding the question and should not only draw from the model's parametric memory.

PubMedQA provides a suitable benchmark for studying this problem. The dataset consists of biomedical research questions extracted from PubMed papers and their contexts and answers. It helps with the evaluation of the capability of an LLM to understand biomedical evidence and produce or choose a suitable response [6]-[8]. Therefore, PubMedQA offers a suitable experiment environment to study the domain knowledge injection at inference time.

1.2. Existing Approaches for LLM Domain Customization

Various methods have been used for domain adaptation with LLMs. One of the commonly used approaches is fine-tuning, where a pre-trained model is tuned with task or domain specific examples. Fine-tuning can be done to boost the performance, but it will need more training resources and enough good domain data. The resulted model may also need to be retrained when the domain knowledge changes [9][10]. The repeated domain customization is costly, especially for large models, because of these requirements.

In prompt engineering, instructions, examples, contextual information, or structures of reasoning are embedded in the input prompt. This method does not change the weights of the internal models and can thus be applied to an already available LLM. Its performance is however highly reliant on timely design and information provided to the model [11]. A fixed prompt might also not be appropriate for queries that require more or different domain knowledge.

Retrieval-Augmented Generation (RAG) takes this idea one step further by fetching information from an external knowledge base prior to generating a response. A common RAG pipeline involves processing the user query into a representation, retrieving relevant documents or passages, and adding them to the prompt [12]. Then, with the query and provided context, the LLM generates an answer. With this mechanism, the external knowledge can be added without retraining the whole model [13].

Conventional RAG can have the issue that not all the retrieved passages are relevant to the question asked. Partially relevant / redundant / unrelated passages can be retrieved in top-k retrieval. These passages are directly added to a prompt, where the LLM might get more context information that doesn't help answer the prompt [14]. Too much context can lead to a higher number of input tokens, longer processing times, and the potential for the model to rely on weak or conflicting evidence. Hence, we should consider retrieval and injection as two distinct, but interdependent, acts [15]. Only retrieved passages that meet a suitable relevance criterion for the present query should be injected.

1.3. Research Gap and Proposed CGKI-LLM Framework

Current approaches to domain customization involve a compromise between adapting parameters of the model and providing external contextual information. Both fine-tuning and prompt-based or RAG-based solutions involve changing the model weights, but in the latter case, it might be necessary to provide the model with unnecessary information. This necessitates a mechanism to adapt an LLM with external domain knowledge and to determine which information to include in the prompt.

One of the main research areas is between retrieval and prompt construction. Typically retrieval methods return a set of passages that are sorted by similarity and return a fixed number of top passages. A rigid selection policy does not always consider the complexity of a query or the real relevance of each passage selected. Some

biomedical questions may involve just one evidence passage, while more specific questions may involve several. Using a set amount of context in either case can lead to either over-contextualization or lack of evidence.

To resolve this concern, this study presents the Context-Gated Knowledge Injection LLM (CGKI-LLM), an approach that allows for fine-tuning-free domain-specific customization of LLMs. The framework proposes a context gate in between knowledge retrieval and prompt construction. Candidates biomedical knowledge is first retrieved from the prepared knowledge repository for each incoming query. Then the semantic relevance of the query and the candidate knowledge chunks is computed. The context gate decides if a certain piece of knowledge has enough information about the query to be added to the final prompt or not.

Dynamically generated prompt: only knowledge passing the gating criterion is added to the prompt. The prompt is generated using the original user query, the retrieved biomedical evidence and task instructions, and then passed to the frozen LLM. The base LLM parameters are the same throughout the process. With this design, external knowledge selection is required to customize the domain instead of retraining the model.

Evaluation of the proposed framework is done on the basis of the main dataset named PubMedQA. It can be compared to the performance of a base LLM without external context, traditional prompt engineering, and typical RAG. Evaluation involves accuracy, precision and recall of answers, hallucination rate, retrieval relevance, token consumption and response time. These measures allow for ratings of answer quality and also the impact of controlling contextual information. The following are objectives of the study:

1. To establish a fine-tuning-free CGKI-LLM framework, which integrates domain-specific information into a Large Language Model based on query-dependent retrieval, relevance evaluation, relevance-based context gating and dynamic prompt construction.
2. To compare the performance of the proposed CGKI-LLM on PubMedQA biomedical dataset with base LLM, conventional prompt engineering and standard RAG technique in terms of Answer Quality, Hallucination reduction, contextual relevance, token consumption and response time.

For domain-specific applications of LLMs, it is not enough to have a vast amount of external information; it is also crucial to ensure that the right information is given to the model for specific queries. To meet this requirement, the proposed CGKI-LLM adds context gating as an intermediate step between retrieval and prompt construction. The framework dynamically chooses knowledge based on query relevance and builds a prompt, instead of tuning the parameters of LLM or adding a certain amount of retrieved information. The structured environment of the biomedical evaluation benchmark, PubMedQA, can be used to explore this approach. Hence, this study explores the potential of selective knowledge injection to improve the accuracy and evidence-based responses in domains without introducing hallucination and irrelevant contextual information.

2. Review of existing work

Large Language Models (LLMs) have advanced from being general-purpose text generators to knowledge-rich systems with the ability to assist in specialized tasks, including healthcare, cybersecurity, software engineering, and scientific reasoning. Their reliance on knowledge they have learnt during the pretraining, however, makes it challenging when a task involves a specific, up-to-date or organization-specific knowledge. Knowledge injection has become a crucial method for providing an LLM with external information without the need for retraining the entire model. The areas of knowledge graphs, Retrieval-Augmented Generation (RAG), long-context processing, prompt-based injection, hierarchical injection, and synthetic knowledge construction have been explored in various recent studies. According to the literature, external knowledge can be used to enhance domain-oriented responses, but there are still some problems to be solved, such as: context relevance, irrelevant information, computational overhead, hallucination and the selection of appropriate knowledge for each specific query. First, this section summarizes recent advancements in knowledge injection and RAG, as well as their applications in various fields and existing challenges that drive context-gated RAG.

2.1. Knowledge Injection and External Knowledge Integration

Knowledge injection offers a way to integrate structured (or unstructured) knowledge from outside the model that may not be included in the model itself. One important source is Knowledge Graphs that explicitly structure entities and relations. One recent study explored knowledge graph injection into LLMs and examined how structured knowledge can complement the knowledge represented in LLMs' model parameters. Such integration can help the domain-oriented reasoning, though it is important to know what graph information should be provided for a certain query [16].

Considerable attention has been paid to RAG in biomedical applications, where medical knowledge is constantly updated and answers are frequently based on information from external sources. An assessment of 10 LLMs

was carried out to investigate the potential for enhanced medical fitness assessment using retrieval-supported generation and to assess the transferability of retrieval-based assistance among models. The findings suggested that providing relevant information from external sources can enhance task performance, but the extent of the enhancement depends on the specific LLM and retrieval setup used [17].

The importance of external evidence has also been investigated in evidence-based neurology. Model variants with retrieval-based document and online information were compared with the base LLMs. The study showed that the source and quality of supporting information affects the performance of an LLM in answering medical questions requiring specialized expertise, highlighting the need to not rely solely on the knowledge contained within the model.The study confirmed that the source and quality of supporting information affects LLM performance in answering medical questions requiring specialised expertise, reinforcing the importance of using external domain information rather than relying solely on the knowledge within the LLM.

Another important parameter in retrieval-supported medical question answering is context length. Long-context RAG explored the impact of larger retrieved medical content on the question-answering capability. The results indicated that the larger contextual inputs can be helpful to offer evidence; however, larger context does not necessarily lead to better results. How evidence is retrieved and provided to the model is thus still important design concerns [19].

2.2. Domain-Specific and Task-Oriented Knowledge Injection

Outside of biomedical QA, knowledge-based AI systems are increasingly shifting to models that communicate with other information collection resources and computational elements. An expanded post-LLM paradigm has been put forward, where knowledge-supported, collaborative and co-evolving models are used instead of standalone language models. This trend points towards an increasing need for explicit knowledge integration for specific AI systems [20]. Also, Knowledge injection has been taken into consideration in multi-agent reinforcement learning for defense of cloud storage. In this method, the knowledge obtained using LLM was integrated with multi-agent decision-making for defense resource allocation in varying scenarios. The work shows that knowledge injection can be done not only for question answering but can be useful for decision-oriented applications where knowledge of the domain should be taken into account during the application operation [21].

Meanwhile, reliance on external knowledge repositories poses security issues. Memory attacks against memory and knowledge bases fed into LLM agents have shown that external information can shape the behaviour of LLM agents. This results in the following key requirement for knowledge-injection frameworks: “just because it is there” does not mean that retrieved knowledge should be accepted automatically. The selection, validation, and controlled entry of outside context can thus both help to improve response quality and the safer use of knowledge [22]. Previous research on informed machine learning investigated when knowledge injection makes a difference in model performance. The results of this study indicated that the learning system can be improved using prior knowledge, but the improvements obtained depend on how the prior knowledge is represented and used in the learning process. This indicates that the availability of domain knowledge is not only important but also the mechanism of injection [23].

Knowledge injection has now been extended to formal protocol modelling with LLMs. The study assessed the impact of extra knowledge on the formal modeling task performed with the support of LLMs and found that in specialized software-related domains, external knowledge can enhance task understanding. The results, however, also show that knowledge injection cannot be thought of as a fixed procedure to be used universally based on the characteristics of the task [24]. Prompt based knowledge injection has also been studied for knowledge-based visual question answering. To help LLMs understand questions that require information outside of the visual content, external knowledge was added to the prompts. This is especially applicable to fine tuning-free customization as domain information can be provided at inference time without changing the model parameters [25].

2.3. Controlled, Hierarchical, and Retrieval-Based Knowledge Injection

More structured mechanisms to control the injected information are being studied recently. Hierarchical knowledge injection has also been used in the field of program repair with LLMs, where the knowledge is arranged and provided hierarchically based on the needs of the repair task. The hierarchical structure shows that not all external information needs to be considered equal, and that structured selection can enhance the usefulness of provided information [26]. Knowledge Injection has also been applied to ensure retrieval augmented code generation. To mitigate the limitations of code generation by LLMs, security-specific

information was provided to LLMs to direct the model's output to specific requirements. This study offers additional support that knowledge injection can be realized as an inference-time control mechanism, without the need for full model retraining [27].

For domain-oriented RAG, systematic knowledge injection via various augmentations has been explored. Various augmentation methods were employed to present different representations of relevant domain knowledge to an LLM. As shown in this work, it is beneficial to carefully prepare domain knowledge before retrieval and generation. Nevertheless, systematic augmentation can boost the quantity of candidate information, making it ever more significant to select it effectively by means of query-dependent way [28]. There has been some success in zero shot and cross-target stance detection with knowledge injection. To help predict targets that are not sufficiently represented in the training examples, external information was generated or selected by an LLM. The results indicate that injected knowledge can be used to compensate for lacking task knowledge and to enhance generalization performance without the traditional task-specific retraining [29].

Synthetic knowledge ingestion is another flow of directions in which the knowledge is initially created or refined and then fed into an LLM. This is an indirect way of getting external content into the model, in that it seeks to enhance the quality of the information an informer can use. The knowledge refinement process can eliminate certain irrelevant information, but will add more processing steps and lead to research into whether the synthetically produced knowledge is correct or not [30]. Following table-1 summarized the analysis of existing work:

Table 1 Summary of Representative Studies Related to the Proposed Work

Application	Knowledge Method	Main Contribution	Limitation Relevant to CGKI-LLM
Medical assessment	RAG with multiple LLMs	Evaluates retrieval support across various LLMs	No explicit query-dependent context gate
Medical QA	Long-context RAG	Studies effect of extended retrieved context	Longer context may contain unnecessary information
Program repair	Hierarchical knowledge injection	Supplies task knowledge at multiple levels	Designed specifically for software repair
Domain-specific RAG	Diverse knowledge augmentation	Uses multiple augmentation strategies for domain knowledge	Additional candidate knowledge requires careful selection

2.4. Research Gap and Concluding Remarks

The reviewed literature shows a clear shift from the use of LLMs solely based on their parametric knowledge to leveraging external domain knowledge at the time of inference. Various methods have been explored for providing extra information, such as knowledge graphs, RAG, long context prompting, hierarchical structure, augmentation, and synthetic knowledge ingestion. Overall, these studies validate that external knowledge is useful for enhancing the performance of LLMs in a specific domain. But, they also reveal a common pitfall: that retrieval or availability of knowledge does not imply the injection of all retrieved knowledge into the prompt.

Conventional RAG typically retrieves the top k passages, and long-context RAG can retrieve significantly more context. While hierarchical and augmentation-based methods enhance knowledge organization, they do not fully solve the problem of deciding whether the knowledge unit is in the final prompt (query-dependent). Excessive or unrelated or not very related information can lead to more tokens consumed, more response time, and more contextual interference. This issue is especially significant in biomedical QA where there is a need for the participant to provide a response that is supported by carefully selected and accurate evidence.

Based on the reviewed studies, the proposed Context-Gated Knowledge Injection LLM (CGKI-LLM) is thus conceived. In contrast to the above two retrieval methods, CGKI-LLM introduces an explicit relevance gate between knowledge retrieval and prompt construction. Candidates' biomedical knowledge is evaluated based on the current query, and only the relevant information that meets the gating criterion will be used to present in the final prompt. The proposed approach can be compared with base LLM, prompt engineering, and traditional RAG in terms of accuracy, F1-score, hallucination rate, contextual relevance, token utilization and response time using PubMedQA as the benchmark. This design covers the identified gap and a seamless domain customization process without fine tuning.

3. Proposed methodology

Context-Gated Knowledge Injection Large Language Model (CGKI-LLM) is proposed to inject the biomedical domain knowledge in the LLM without changing the internal parameters of the LLM. The methodology involves preparing the dataset, preprocessing the text, constructing the knowledge, retrieving the text semantically, selecting the knowledge based on context, dynamically constructing the prompt, and generating the response. PubMedQA is chosen as a major benchmark for it has biomedical questions with scientific contexts and reference answers. While traditional RAG typically appends a set number of retrieved passages to the prompt, CGKI-LLM assesses each knowledge chunk and only includes it if it meets a relevance threshold defined by the query. This helps to remove irrelevant context information and preserve evidence needed to answer the biomedical query.

The proposed CGKI-LLM framework as shown in figure-1 starts with the PubMedQA dataset and then goes through data preparation, biomedical knowledge repository construction, chunking, and vector embedding. Semantic retrieval involves discovering the knowledge chunks that are relevant and ranking them based on the relevance with the input query. The proposed context-gating mechanism filters out irrelevant or redundant information and keeps only appropriate domain knowledge for dynamically building the prompt. This selected knowledge and query is then fed into the "frozen" LLM, without the need for fine-tuning. Finally, the performance, retrieval quality, hallucination and computation metrics of the biomedical answers generated are evaluated against standard metrics.

Figure 1 Proposed workflow of the Context-Gated Knowledge Injection LLM (CGKI-LLM) framework

3.1. About dataset

The PubMedQA dataset is the primary one used to assess the proposed framework. It is a biomedical question-answering dataset based on scientific contexts extracted from PubMed research articles, which consist of questions, the scientific contexts from which the questions are drawn and answers. The dataset can be used to make a yes, no, maybe decision and is appropriate for domain-oriented reasoning. This expert-annotated subset can serve as the main evaluation set, since the answers to these questions are manually labeled and are deemed to be more suitable as a reference for controlled comparison. Every instance of data is of the form:

$$D = \{(Q_i, C_i, Y_i)\}_{i=1}^{N} \quad (1)$$

where D is the entire set of data, the ith biomedical question is represented by Q_i, the scientific context of $the\ i^{th}$ biomedical question is represented by C_i, the reference label/answer of the i^{th} biomedical question is represented by $Y_i, and\ N$ represents the total number of instances.

To build the external knowledge repository, the biomedical contexts are isolated from the question-answer pairs. With this separation, the LLM is not able to directly access the reference answer but instead gets the domain information as a result of the proposed retrieval and gating pipeline.

3.1.1. Text Preprocessing

Preprocessing is mainly done on the external knowledge documents before embedding & indexing. Unicode normalization, whitespace fixing, stripping of non-informative formatting symbols, preservation of sentence-boundaries and duplicate-content removal is done. Aggressive text cleaning will remove information that is needed for medical reasoning and for that reason, biomedical terminology, abbreviations, numerical measurements and clinically meaningful punctuation are retained. Normalized similarity is used to determine exact or near duplicates of contexts. The duplicated decision is stated as in eq.1:

$$Dup(C_i, C_j) = 1, if\ Sim(C_i, C_j) \geq \delta;\ otherwise\ 0 \quad (2)$$

Here, $C_i\ and\ C_j$ are two context records, $Sim(C_i, C_j)$ is the similarity between the two contexts, δ is the duplicate threshold and $Dup(C_i, C_j)$ is the binary duplicate indicator. Records that have been identified as duplicates are only saved once in the knowledge repository.

The cleaned documents are then broken down into smaller chunks of knowledge. Chunking based on arbitrary splitting of characters is not preferred, as biomedical sentences should be semantically meaningful. It is possible to have a limited overlap between neighbouring chunks to avoid loss of information near chunk boundaries. This becomes a knowledge repository in the form shown in eq.2:

$$K = \{K_1, K_2, \ldots, K_m\} \quad (3)$$

Here, K is the complete external biomedical knowledge repository, $K_j\ is\ the\ j-th$ biomedical knowledge chunk after preprocessing and m is the number of biomedical knowledge chunks after preprocessing.

3.2. Semantic Representation and Candidate Knowledge Retrieval

In CGKI-LLM, both query and all knowledge chunks are mapped to a common semantic vector space by sentence embedding model when a biomedical query is entered. The LLM is still frozen during this process. The embedding operation is shown in eq.4:

$$E_Q = f_e(Q), \quad E_{K_j} = f_e(K_j) \quad (4)$$

where Q is the input biomedical query, K_j is a candidate knowledge chunk, $f_e\ is$ the embedding function, E_Q is the vector representation of the biomedical query and E_{K_t} is the vector representation of the knowledge chunk K_j. The similarity between query and each knowledge chunk is measured in terms of Semantic similarity, which is computed as Cosine similarity as in eq.5:

$$S_j = (E_Q \cdot E_{K_j}) / (\|E\_Q\| \, \|E\_K_j\|) \quad (5)$$

where S_j is the semantic relevance score for the j^{th} knowledge chunk, $E_Q \cdot E_{K_j}$ refers to the dot product between the two vectors and $\|E_Q\|\ and\ \|E_{K_j}\|$ are the vector magnitudes. The higher S_j is, the more semantically similar the question is with the candidate biomedical knowledge.

The chunks are then sorted by S_j and an initial candidate pool of the top-ranked chunks is produced. The retrieved chunks are not put in the prompt automatically as with normal RAG. These are forwarded to the context gate, which is proposed for further review.

3.3. Proposed Context-Gated Knowledge Injection

The key of CGKI-LLM is context gating. To decide if there is enough information about the actual question in a retrieved knowledge chunk to include it in the final prompt. A normalized relevance probability is first calculated from the semantic score shown in eq.6:

$$P_j = 1/(1 + e^{-zj}) \quad (6)$$

where P_j is the normalized relevance probability of chunk $K_j\ and\ z_j$ is the transformed relevance score of chunk K_j. The score is then passed through a sigmoid operation which maps it to the range [0, 1] so that different candidate chunks can be evaluated on the same gating scale. CGKI-LLM adopts the query-dependent threshold for each biomedical question rather than a single threshold. It is figured out as eq.7:

$$\tau_Q = \tau^0 + \lambda(1 - \bar{S}_Q) \quad (7)$$

where τ_Q is the threshold for query Q, τ_0 is the initial gating threshold, λ is the amount by which the threshold is adjusted, and $\bar{S}_Q$ is the average similarity of the retrieved candidate chunks for the query. The threshold thus

reacts to the evidence relevance distribution. Then the context gate decides if the individual chunks are to be admitted as in eq.8:

$G_j = 1, if\ P_j \geq \tau_Q;\ otherwise\ G_j = 0$ (8)

where G_j is the binary gating decision, P_j is the relevance probability of $K_j, and\ \tau\_Q$ is the query-dependent threshold. If $G_j = 1$, then the knowledge chunk is accepted, otherwise $G_j = 0$, it is excluded. Finally, the contextual evidence is built up from only the admitted chunks shown in eq.9:

$C = \sum_{j=1}^{k} G_j K_j$ (9)

Here, C represents the biomedical context of interest provided to the LLM, G_j is the gating decision and K_j the corresponding knowledge chunk being retrieved of which k is the total number of retrieved candidate chunks. When it comes to implementation, the summation is not numerical addition, but ordered concatenation of the textual chunks accepted.

3.4. Dynamic Prompt Construction and LLM Response Generation

Gated, CGKI-LLM dynamically generates the final input prompt. The prompt includes task instructions, the original biomedical question, the selected evidence C, and a response constraint to ask for a response that is supported by the provided context. The prompt that has been built is stated as in eq.10:

$P = I \oplus Q \oplus C \oplus R$ (10)

Here P is the final prompt, I is the domain/task instruction, Q is the original question text, C is the gated biomedical context, R is the response format, and $\oplus$ is ordered textual concatenation. Then the prompt is provided to the frozen LLM:

$\hat{Y} = LLM(P;\ \theta_f)$ (11)

where $\hat{Y}$ is the answer being generated, P is the dynamically-generated prompt, LLM is the language model being used, and θ_f are the frozen parameters of the model. The subscript f means that the parameters do not change during domain customization and no gradient based fine tuning is done.

3.5. Performance Evaluation

CGKI-LLM is evaluated alongside a base LLM, traditional prompt engineering, and traditional RAG with the same evaluation set. The parameters such as accuracy, precision, recall, F1-score, hallucination rate, contextual relevance, token consumption, and response time can be recorded.

Besides predictive performance, token consumption is analyzed, as a goal of context gating is to prevent irrelevant prompt content. The number of "hallucinated" answers, or those for which there is no biomedical evidence in the provided sources, is used to measure the hallucination rate. Retrieval relevance is assessed independently to assess if the gate is able to weed out weakly related information.

4. Results and discussion

The results in Table 2 show that the proposed CGKI-LLM outperforms Zero-shot Base LLM, Prompt Engineering, and Standard RAG. The proposed CGKI-LLM is able to achieve the highest accuracy of 89.20% with 88.75% precision, 89.10% recall and 88.92% F1-score. The accuracy of standard RAG is 84.30%, prompt engineering is 78.60%, and zero-shot is 72.40%. Moreover, CGKI-LLM also achieves the lowest hallucination rate at 3.80%, while Standard RAG is at 7.20%. The results show that the context-gated knowledge selection method can not only improve the answer quality but also decrease the number of unsupported answers.

Table 2 Overall Performance of Different Knowledge-Injection Strategies

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Hallucination Rate (%)
Base LLM – Zero-shot	72.4	71.85	70.92	71.38	14.8
Prompt Engineering	78.6	77.94	77.31	77.62	10.7
Standard RAG	84.3	83.75	83.1	83.42	7.2
Proposed CGKI-LLM	**89.2**	**88.75**	**89.1**	**88.92**	**3.8**

Table 3 assesses the quality of the retrieved knowledge, in terms of Recall@5, Precision@5, context relevance, faithfulness and irrelevant context. The proposed CGKI achieves the best Recall@5 of 0.918 and Precision@5 of 0.903, which shows identification of more relevant biomedical information related to the query. It also has context relevance and faithfulness scores of 0.921 and 0.934. The irrelevant context is reduced to only 4.30%, as opposed to 18.60% for BM25, 14.20% for Dense Retrieval and 9.70% for Hybrid Retrieval. This indicates that the context gate is quite successful in filtering out weakly related knowledge from the final prompt.

Table 3 Knowledge Retrieval and Context Quality

Method	Recall@5	Precision@5	Context Relevance	Faithfulness	Irrelevant Context (%)
BM25 Retrieval	0.812	0.746	0.781	0.793	18.6
Dense Retrieval	0.846	0.801	0.824	0.831	14.2
Hybrid Retrieval	0.881	0.842	0.861	0.872	9.7
Proposed CGKI	**0.918**	**0.903**	**0.921**	**0.934**	**4.3**

Table 4 compares computational characteristics of various methods of injecting context using input tokens, response time, knowledge usage, and hallucination. CGKI-LLM has an average of 1,640 input tokens, while Full-Context Prompt has 3,850 and Standard RAG has 2,760. It also shows the lowest Response Time of 2.28 sec and the highest Knowledge Utilization of 91.20%. Meanwhile, the hallucination rate for Hallucination in the Loop falls to 3.80%, while Full-Context Prompt, Standard RAG and Top-k RAG record 9.80%, 7.20% and 5.90% respectively. Results show that selective knowledge admission can lower the number of useless prompt content while keeping useful domain evidence.

Table 4 Computational and Prompt-Cost Analysis

Method	Avg. Input Tokens	Response Time (s)	Knowledge Utilization (%)	Hallucination (%)
Full-Context Prompt	3,850	3.82	62.4	9.8
Standard RAG	2,760	3.14	71.8	7.2
Top-k RAG	2,180	2.71	78.5	5.9
Proposed CGKI-LLM	**1,640**	**2.28**	**91.2**	**3.8**

A comparison of the proposed CGKI-LLM with the existing KG-RAG approach using the same PubMedQA dataset is shown in Table 5. The accuracy of the KG-RAG method is 81.30% while the proposed CGKI-LLM is 89.20%. This means it is 7.90 percentage points more accurate than the current method. The enhancement could be related to query-dependent context gating, which filters retrieved knowledge before adding to the LLM prompt. The comparison validates the effectiveness of the controlled knowledge injection for biomedical question answering.

Table 5 Comparison with existing work with similar dataset

Study	Method	Dataset	Accuracy
Wang et al. [31]	KG-RAG	PubMedQA	**81.3 %**
Proposed Work	**CGKI-LLM**	**PubMedQA**	**89.2 %**

5. Conclusion, limitation and future scope

As domain-specific LLM responses become more crucial, external knowledge is vital to address this, but model fine-tuning can be computationally expensive. The proposed work introduced a Context-Gated Knowledge Injection (CGKI) framework for customizing Large Language Models (LLMs) in a domain-specific manner

without fine-tuning the LLMs' internal parameters. The framework integrates semantic retrieval, candidate ranking, query-dependent context gating, selective knowledge injection and dynamic construction of prompts, and PubMedQA is used as the biomedical benchmark. Unlike traditional RAG, retrieved information is not directly fed into the LLM, the context gate filters out weakly related and redundant information before the response generation process. The results of the experiments show that the accuracy of the CGKI-LLM was 89.20% with an F1-score of 88.92%, surpassing the accuracy of standard RAG which was 84.30%. The proposed method also decreased the hallucination rate by 3.80% against the standard RAG (7.20%). In retrieval view, CGKI got a score of context relevance as 0.921 and irrelevant context as 4.30%. Additionally, the framework needed less input tokens and exhibited improved knowledge utilization as compared to full-context and traditional retrieval methods. These results suggest that the post-retrieval knowledge control is useful for Biomedical QA. Instead of providing more contextual information to an LLM, CGKI-LLM aims to choose evidence that is close to each query. The proposed framework offers a practical solution for fine-tuning general-purpose LLMs on specialized domains without requiring finetuning, enhancing the quality of answers, grounding them in evidence, and reducing computational demands.

5.1. Limitations and Future Scope

The current study is restricted to the biomedical domain PubMedQA dataset, and may not generalize to other specialized domains or more complex question types. Another factor that influences the effectiveness of CGKI-LLM is the correct embedding quality, retrieval accuracy, chunk configuration, and selected gating threshold. Future studies can test the framework on more legal, financial and cyber security and medical datasets. Multi-stage evidence verification, credibility assessment of knowledge sources, and threshold selection based on query-complexity are further extensions to the context gate. Future studies with other open-source LLMs and larger knowledge bases would be useful to explore the genericity and computation of CGKI-LLM in domain-specific scenarios.

References

[1]. S. Wang, H. Yang, and W. Liu, “Research on the construction and application of retrieval enhanced generation (RAG) model based on knowledge graph,” Sci Rep, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-21222-z.

[2]. Y. Xu, D. Pi, and Y. Xu, “CADI: A cross-source alignment and dynamic knowledge injection framework for medical QA,” Neurocomputing, vol. 697, Oct. 2026, doi: 10.1016/j.neucom.2026.134184.

[3]. S. Wang, H. Yang, and W. Liu, “Research on the construction and application of retrieval enhanced generation (RAG) model based on knowledge graph,” Sci Rep, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-21222-z.

[4]. Z. Su et al., “MedKit: Multi-level feature distillation with knowledge injection for radiology report generation,” Expert Syst Appl, vol. 296, Jan. 2026, doi: 10.1016/j.eswa.2025.129003.

[5]. S. K. Ramareddy, "Integrating AI-Driven Data Visualization and Data Warehouse Analytics for Predictive Decision-Making in Financial and Stock Market Systems," 2026 International Symposium of Systems, Advanced Technologies and Knowledge (ISSATK), Hammamet, Tunisia, 2026, pp. 1-6, doi: 10.1109/ISSATK69667.2026.11566934.

[6]. Y. Peng, H. Hu, F. Li, Y. Jiang, J. Tang, and Y. Liu, “LLM4Game: Multi-agent reinforcement learning with knowledge injection for dynamic defense resource allocation in cloud storage,” Computer Networks, vol. 273, Dec. 2025, doi: 10.1016/j.comnet.2025.111748.

[7]. L. Yao et al., “A flooding fault diagnosis PEMFC based on self-supervised learning and knowledge injection pre-training,” Int J Hydrogen Energy, vol. 231, May 2026, doi: 10.1016/j.ijhydene.2026.154795.

[8]. Y. Zhang, X. Ren, S. Fuchs, and R. Amor, “A knowledge injection method for supporting automated compliance checking of shield tunnel designs,” Advanced Engineering Informatics, vol. 68, Nov. 2025, doi: 10.1016/j.aei.2025.103781.

[9]. X. Zhao, X. Li, R. Jiang, and B. Tang, “Resolving multimodal ambiguity via knowledge-injection and ambiguity learning for multimodal sentiment analysis,” Information Fusion, vol. 115, Mar. 2025, doi: 10.1016/j.inffus.2024.102745.

[10]. Y. Zhao, K. He, Y. Zhang, Y. Zhou, T. Zhou, and D. Liang, “Structured prompt-guided knowledge injection for medical image segmentation,” Pattern Recognit, vol. 179, Nov. 2026, doi: 10.1016/j.patcog.2026.113852.

[11]. S. Elmitwalli, J. Mehegan, S. Braznell, and A. Gallagher, "Scalable evaluation framework for retrieval augmented generation in tobacco research using large Language models," Sci Rep, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-05726-2.
[12]. Q. Yang et al., "Dual retrieving and ranking medical large language model with retrieval augmented generation," Sci Rep, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-00724-w.
[13]. Sunkara, S. P. (2025). Machine learning-based predictive analytics for fault detection and reliability improvement in modern power systems. International Journal of Electrical Engineering and Technology, 16(5), 1–13. https://doi.org/10.34218/IJEET_16_05_001
[14]. S. Liu, A. B. McCoy, and A. Wright, "Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines," Journal of the American Medical Informatics Association, vol. 32, no. 4, pp. 605–615, Apr. 2025, doi: 10.1093/jamia/ocaf008.
[15]. I. Lopez et al., "Clinical entity augmented retrieval for clinical information extraction," NPJ Digit Med, vol. 8, no. 1, Dec. 2025, doi: 10.1038/s41746-024-01377-1.
[16]. Y. H. Ke et al., "Retrieval augmented generation for 10 large language models and its generalizability in assessing medical fitness," NPJ Digit Med, vol. 8, no. 1, Dec. 2025, doi: 10.1038/s41746-025-01519-z.
[17]. A.Cadeddu et al., "A comparative analysis of knowledge injection strategies for large language models in the scholarly domain," Eng Appl Artif Intell, vol. 133, Jul. 2024, doi: 10.1016/j.engappai.2024.108166.
[18]. E. Coppolillo, "Injecting Knowledge Graphs into Large Language Models," May 2025, [Online]. Available: http://arxiv.org/abs/2505.07554
[19]. Y. H. Ke et al., "Retrieval augmented generation for 10 large language models and its generalizability in assessing medical fitness," NPJ Digit Med, vol. 8, no. 1, Dec. 2025, doi: 10.1038/s41746-025-01519-z.
[20]. L. Masanneck, S. G. Meuth, and M. Pawlitzki, "Evaluating base and retrieval augmented LLMs with document or online support for evidence based neurology," NPJ Digit Med, vol. 8, no. 1, Dec. 2025, doi: 10.1038/s41746-025-01536-y.
[21]. G. Zhang et al., "Leveraging long context in retrieval augmented language models for medical question answering," NPJ Digit Med, vol. 8, no. 1, Dec. 2025, doi: 10.1038/s41746-025-01651-w.
[22]. F. Wu et al., "Knowledge-Empowered, Collaborative, and Co-Evolving AI Models: The Post-LLM Roadmap," Engineering, vol. 44, pp. 87–100, 2025, doi: https://doi.org/10.1016/j.eng.2024.12.008.
[23]. Y. Peng, H. Hu, F. Li, Y. Jiang, J. Tang, and Y. Liu, "LLM4Game: Multi-agent reinforcement learning with knowledge injection for dynamic defense resource allocation in cloud storage," Computer Networks, vol. 273, p. 111748, 2025, doi: https://doi.org/10.1016/j.comnet.2025.111748.
[24]. Z. Chen, Z. Xiang, C. Xiao, D. Song, and B. Li, "AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases," in Advances in Neural Information Processing Systems, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., Curran Associates, Inc., 2024, pp. 130185–130213. doi: 10.52202/079017-4136.
[25]. S. Gajula, "AI-Powered Forecasting Models, Optimizing Working Capital, Supply Chain Financing," 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM), Bhopal, India, 2025, pp. 1-6, doi: 10.1109/RCSM67767.2025.11507813.
[26]. L. von Rueden, J. Garcke, and C. Bauckhage, "How Does Knowledge Injection Help in Informed Machine Learning?," in 2023 International Joint Conference on Neural Networks (IJCNN), 2023, pp. 1–8. doi: 10.1109/IJCNN54540.2023.10191994.
[27]. Y. Lin et al., "How Well Does Knowledge Injection Enhance LLM-Aided Formal Protocol Modeling?," in 2026 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2026, pp. 1–6. doi: 10.1109/SANER67736.2026.00103.
[28]. Z. Hu, P. Yang, F. Liu, Y. Meng, and X. Liu, "Prompting Large Language Models with Knowledge-Injection for Knowledge-Based Visual Question Answering," Big Data Mining and Analytics, vol. 7, no. 3, pp. 843–857, 2024, doi: 10.26599/BDMA.2024.9020026.
[29]. R. Ehsani, E. Parra, S. Haiduc, and P. Chatterjee, "Hierarchical Knowledge Injection for Improving LLM-based Program Repair," in 2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2025, pp. 1440–1452. doi: 10.1109/ASE63991.2025.00122.
[30]. B. Lin, S. Wang, Y. Qin, L. Chen, and X. Mao, "Give LLMs a Security Course: Securing Retrieval-Augmented Code Generation via Knowledge Injection," in Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, in CCS '25. New York, NY, USA: Association for Computing Machinery, 2025, pp. 3356–3370. doi: 10.1145/3719027.3765049.

[31]. K. Bhushan et al., "Systematic Knowledge Injection into Large Language Models via Diverse Augmentation for Domain-Specific RAG," in Findings of the Association for Computational Linguistics: NAACL 2025, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 5937–5958. doi: 10.18653/v1/2025.findings-naacl.329.
[32]. Z. Zhang, Y. Li, J. Zhang, and H. Xu, "LLM-Driven Knowledge Injection Advances Zero-Shot and Cross-Target Stance Detection," in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), K. Duh, H. Gomez, and S. Bethard, Eds., Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 371–378. doi: 10.18653/v1/2024.naacl-short.32.
[33]. J. Zhang, W. Cui, Y. Huang, K. Das, and S. Kumar, "Synthetic Knowledge Ingestion: Towards Knowledge Refinement and Injection for Enhancing Large Language Models," in Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 21456–21473. doi: 10.18653/v1/2024.emnlp-main.1196.
[34]. Balasingam, K. (2026). Large Language Model-Based Intelligent Code Generation and Automated Software Engineering Framework. Research Journal of Computer Systems and Engineering, 44–49.