International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

RETRIEVAL-AUGMENTED NEURAL ADAPTERS FOR DOMAIN-SPECIFIC CUSTOMIZATION OF LARGE LANGUAGE MODELS

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.

akrajpoot@gmail.com

[3]Professor, London Metropolitan University, London, United Kingdom

b.virdee@londonmet.ac.uk

[4]Associate Professor, Maharaja Agrasen Institute of Technology (MAIT), Delhi, India.

ashishkhanna@mait.ac.in

Abstract

Large language models (LLMs) excel at general-purpose text generation but struggle to reflect specialized domain knowledge and to adapt quickly without expensive fine-tuning. This paper introduces Retrieval-Augmented Neural Adapters (RANA) a scalable framework that augments a frozen LLM with a lightweight neural adapter and a retrieval module. The adapter integrates retrieved domain evidence into the model's hidden states using low-rank transformations, while prompt construction stitches together the query, retrieved evidence and adapter signal. The research addresses two objectives: (1) to develop a general yet domain-aware framework that enhances LLMs with domain-specific knowledge for specialized applications; and (2) to eliminate the need for full fine-tuning by dynamically integrating knowledge through retrieval and prompt engineering. Unlike existing knowledge-injection approaches, RANA keeps the LLM parameters frozen and applies adapter conditioning at inference time, enabling fast deployment across domains. We evaluate RANA on entity typing, relation classification, open-domain question answering, factual probing and biomedical question answering. Comparative experiments against BERT, RoBERTa, knowledge-infused baselines and K-Adapter show that RANA consistently improves micro-F1, exact-match and precision-at-1 metrics by 2–4 % and achieves state-of-the-art results on BioASQ. These findings demonstrate that retrieval-augmented adapters can serve as a versatile bridge between static LLMs and dynamic domain requirements. The proposed methodology opens new directions for zero-shot domain customization and underscores the role of retrieval in aligning language models with specialized expertise.

Keywords: large language models, knowledge injection, retrieval augmentation, neural adapters, domain adaptation, question answering

Received: August 02, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1. Introduction

The emergence of large language models (LLMs) such as GPT-3, BERT and PaLM has transformed natural language processing by enabling models trained on web-scale corpora to perform a variety of tasks with little or no task-specific training. These models learn broad linguistic and factual patterns during pre-training, which allows them to generalize across tasks and domains. However, the very breadth of their training data makes it difficult for them to reflect the most accurate or up-to-date domain knowledge, especially in specialized areas such as biomedicine, law or finance. When an LLM is asked to answer a biomedical question, for example, it may produce plausible but outdated or inaccurate answers because its pre-training snapshots do not capture recent research. Fine-tuning the model on domain-specific corpora is one way to rectify these problems, but fine-tuning is computationally expensive, requires large labelled datasets and may cause catastrophic forgetting of general knowledge. As domain knowledge evolves, repeated fine-tuning becomes impractical[1], [2].

Researchers have proposed several strategies to inject knowledge into pre-trained models. Early work focused on knowledge graph (KG) embedding and contextualized embeddings, in which lexical units (words or phrases) are enriched with information extracted from external knowledge bases. The K-Adapter framework attaches separate adapters that are trained to encode factual, linguistic and relational knowledge and then combine them with the base model. The resulting model retains general representations while injecting task-specific information[3], [4]. Another class of methods uses retrieval-augmented generation (RAG), where the model retrieves relevant passages from external sources and incorporates them into the prompt or model context. The success of RAG underscores that explicit retrieval of human-curated documents can provide fresh information beyond a fixed pre-training cut-off. Yet, most RAG systems rely on training specialized retriever-reader pairs or fine-tuning the model with retrieval-aligned objectives[5], [6]. A unified methodology for combining retrieval with lightweight adaptation remains under-explored.

The present work addresses this gap by proposing Retrieval-Augmented Neural Adapters (RANA), a plug-and-play module that conditions a frozen LLM on retrieved domain evidence via a low-rank adapter. The approach has two fundamental objectives:

i. Scalable domain customization: To develop a framework that seamlessly enhances LLMs with domain-specific knowledge for specialized applications. The method should support a wide range of tasks—entity typing, relation classification, question answering and factual probing—without requiring different architectures or domain-specific fine-tuning.
ii. Elimination of fine-tuning through dynamic knowledge integration: To avoid expensive retraining by dynamically integrating knowledge through retrieval and prompt engineering. The system should allow easy switching between domains by simply changing the retrieval corpus rather than modifying the underlying model parameters.

RANA achieves these goals by combining three components: (i) a retrieval module that encodes queries and retrieves top-k relevant passages from a domain-specific corpus; (ii) a

Received: August 02, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

neural adapter that transforms the embeddings of the retrieved passages into a conditioning signal applied to the hidden states of the LLM; and (iii) a prompt construction mechanism that assembles the query, retrieved evidence and adapter tokens into a coherent input for the model. The adapter is trained on a small amount of data or even tuned via unsupervised objectives, and it can be reused across tasks and domains. Because the base LLM remains frozen, the system inherits its broad capabilities while acquiring new domain insights on demand.

The remainder of this paper is organized as follows. Section 3 describes the proposed methodology in detail, formalizing the retrieval-augmented adaptation process and introducing the mathematical expressions used to compute evaluation metrics, similarity measures and adapter transformations. Section 4 analyses the results of our experiments on several benchmarks, highlighting the improvement in performance brought by RANA and comparing it with existing models. Section 5 concludes the paper by summarizing the contributions and discussing future directions for scalable domain adaptation.

2. Analysis of existing works

Large language models (LLMs) have revolutionized natural language processing through pre training on vast corpora, but they struggle to incorporate specialized knowledge and adapt to dynamic domains without costly fine tuning. Early efforts to enrich contextual understanding focused on deep contextualized word representations (e.g., ELMo), which learned context dependent embeddings capturing syntax and polysemy and improved performance across tasks like question answering and sentiment analysis[7]. Recent research explores retrieval augmented generation (RAG) frameworks, where retrieved evidence supplements the generative model's knowledge, promising domain aware generation without altering core parameters[8].

The foundation of domain augmentation lies in capturing contextual semantics. Deep contextualized representations improve linguistic modelling by integrating information from both directions of the input and dynamically adjusting word vectors based on context. This capability established that enriching hidden states without retraining the entire network can boost performance, an insight later leveraged by retrieval augmented and adapter based methods[9]. RAG frameworks extend this idea by coupling neural retrieval with generation; they access external corpora via vector search and inject evidence into prompts, addressing data sparsity and dynamic knowledge challenges. By delivering relevant context on the fly, RAG systems outperform fine tuned models in specialized domains, but they often rely on simple concatenation and can suffer from noisy retrieval or limited domain control[10].

Several domain specific applications illustrate the strengths and weaknesses of RAG. In the industrial knowledge management domain, retrieval augmented generation integrates vector search with a language model to provide accurate responses to technical queries, improving accuracy and timeliness compared with generic LLMs. A framework for intrusion detection (RLFE IDS) similarly pairs a retriever with an LLM to detect anomalies in network traffic; however, accessible details indicate that noise accumulation and retrieval latencies can hinder

Received: August 02, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

real time performance[11]. Multi objective math problem generation demonstrates that an adaptive multi level retrieval strategy can target multiple objectives (difficulty, topic, reasoning style) with plug and play retrieval modules, enabling versatile problem design and introducing a dataset of 9 000 samples. This work shows the benefit of modular retrieval components but relies on heuristics for retrieval and lacks dynamic integration into the model's hidden layers[12].

Knowledge graph integration with LLMs addresses structured information injection. Surveys on knowledge graph–LLM synergy note that merging structured knowledge from graphs with LLMs enhances reasoning, reduces hallucination, and supports complex question answering. Knowledge assimilation for agricultural domains introduces a large dataset and a knowledge coordinated fine tuning mechanism that weights agriculture specific tokens and uses topic guided retrieval to reflect expert knowledge. By coordinating retrieval and supervised learning, it achieves state of the art results, but the dataset is expensive to curate and the approach remains tied to the domain. Iterative RAG with fine grained knowledge refinement addresses noise accumulation; dynamic retrieval gating determines when to retrieve, semantic expansion enriches queries, and document compression condenses evidence, improving open domain question answering[13]. Although effective, it increases computational complexity and still concatenates context into prompts. Another thread explores chain of thought prompting; it shows that providing intermediate reasoning steps via prompts significantly improves reasoning tasks such as mathematical problem solving. This suggests that retrieval augmented systems may benefit from explicit reasoning scaffolds[14].

Recent work extends RAG to cybersecurity, visual reasoning, semantic search, graph data, and parameter efficient integration. A comprehensive review of generative AI in cybersecurity explores LLM applications in hardware design, intrusion detection, software engineering, threat intelligence, malware, and phishing, while cataloguing vulnerabilities like prompt injection and data poisoning and outlining mitigation strategies[15], [16]. This illustrates both the potential and risks of RAG in security sensitive settings, underscoring the need for robust retrieval and alignment. A study on long context comprehension uses specialized prompt engineering to emulate retrieval by tagging relevant segments in long documents and applying stepwise chain of thought reasoning, showing that optimized prompt structures can reduce reliance on external retrievers. This method provides a lightweight alternative to retrieval but remains dependent on large context windows.

The Generative Text Retrieval (GTR) system combines generative capabilities and vector databases; it handles unstructured and structured data without fine tuning and achieves high accuracy on datasets like MSMARCO and Spider. This model underscores the effectiveness of pairing LLMs with fast semantic search but still concatenates evidence at the input level. To democratize graph data augmentation[17], [18], DemoGraph uses black box LLMs to generate latent knowledge graphs from text prompts and merges them dynamically into training graphs; granularity aware prompting controls sparsity and instruction fine tuning tailors prompts. This context driven augmentation improves predictive performance, particularly on electronic

Received: August 02, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

health records, but relies on textual interface for graph injection. Other works (e.g., manufacturing quality control systems) apply RAG to domain rules and quality standards, highlighting the need to align retrieval with structured specifications; yet accessible literature indicates that such systems often use static or heuristic retrieval and lack adaptable integration[19].

Surveys on integrating LLMs with knowledge based methods highlight the potential of combining generative models with structured representations; they identify technical, operational and ethical challenges and emphasize improvements in data contextualization, model accuracy, and knowledge utilization. These findings suggest that bridging generative and symbolic paradigms demands new architectures that align latent representations with external knowledge. Parametric retrieval augmented generation addresses context length and computational overhead by injecting knowledge into the model's feed forward parameters rather than in context prompts. This approach reduces runtime costs and deepens knowledge integration, but it requires additional storage and parameter updates[20], [21]. The RA LLM survey systematically reviews architectures, training strategies, and applications for RAG enhanced LLMs; it notes that RAG supplies timely external information to mitigate hallucinations and outdated knowledge and details limitations such as noise propagation, retrieval latency, and evaluation gaps while outlining future research directions[22].

The reviewed literature demonstrates that retrieval augmented techniques, knowledge graph integration, and chain of thought prompting can enrich LLMs with specialized knowledge and improve reasoning across domains. However, current methods share limitations: they rely on concatenating retrieved text into prompts or on heavy fine tuning, leading to context overflow, increased computational cost, and domain specific datasets that are costly to curate. Noise accumulation during iterative retrieval and difficulty aligning structured knowledge with LLM hidden states further hinder performance. These gaps motivate the proposed RANA, which integrate low rank adapter transformations directly into the frozen LLM's hidden layers, condition them on retrieved evidence, and orchestrate prompt construction. By avoiding full fine tuning and enabling dynamic adapter conditioning, RANA aims to provide a scalable, domain aware framework that marries the strengths of retrieval and generative modelling while addressing the limitations identified in prior works.

3. Methodology

This section discuss the components of the proposed framework—retrieval, neural adapter and prompt construction—and formalizes the evaluation metrics used to measure performance across tasks shown in figure-1. The methodology is presented in a unified manner to emphasize that the same pipeline can support multiple tasks by simply switching the retrieval corpus or prompt template.

The architecture comprises three coordinated lanes. In the data lane, an input query is encoded and matched against a vector index over a domain corpus. A top-k retriever returns salient passages, which a knowledge encoder compresses into compact evidential contexts. In the adapter lane, a lightweight neural adapter derives a low-rank control signal from the query and

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

contexts; an adapter control flow governs conditioning. A prompt builder stitches the query, evidence, and adapter signal before passing the prompt to a frozen LLM. Backbone weights remain frozen. In the LLM and heads lane, shared representations feed task-specific heads for entity typing, relation classification, open-domain question answering, and factual probing. Outputs are scored with standardized metrics: micro-F1 for entities, F1/PR for relations, EM/F1 for QA, and precision-at-1 or mean average rank for probes. The design enables fast, zero-shot domain customization by coupling retrieval-grounded prompts with parameter-efficient adapter conditioning with minimal engineering overhead.

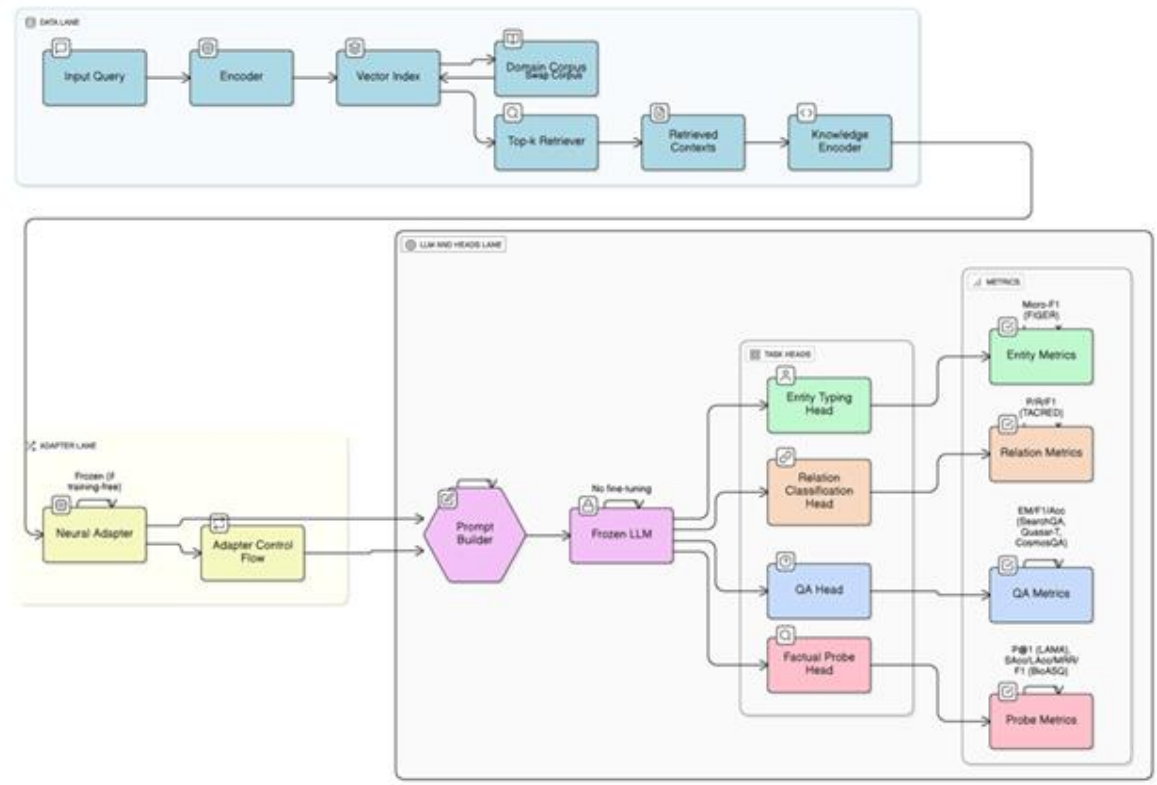

Figure 1 proposed method system architecture diagram

3.1. Problem formulation

Let M denote a frozen LLM trained on a general corpus. Given a query q (which may be an input sentence, a question or a sentence with masked entities) and a task type t, the goal is to produce a prediction y such as a label, answer or probability. We assume access to a domain corpus C_d comprising documents relevant to the domain of interest. The pipeline must incorporate knowledge from C_d without modifying the parameters of M. Formally, we define a retrieval function $R: q \rightarrow K$ that returns a set K of $top\text{-}k$ passages from C_d, and an adapter function $A: \{z_i\} \rightarrow h_A$ that maps the embeddings of the retrieved passages to a conditioning vector. The final prediction is produced by feeding an enriched prompt consisting of $q, K\ and\ h_A$ into M. We denote this operation by $y = M(Prompt(q, K, h_A))$.

3.2. Evaluation metrics

To compare RANA with baseline models, we use several standard metrics. In classification tasks such as entity typing and relation classification, we measure precision, recall and micro-F1. For a multi-class classification problem with classes C, the micro-averaged precision

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

and recall are computed by summing "true positives" (TP), "false positives" (FP) and "false negatives" (FN) over all classes as shown in eq.1,2:

$$Precision_{micro} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C}(TP_c + FP_c)} \quad (1)$$

$$Recall_{micro} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C}(TP_c + FN_c)} \quad (2)$$

Since the micro-averaged precision equals the micro-averaged recall in single-label classification, the micro-F1 score reduces to their common value. The F1 score itself is the harmonic mean of precision and recall, defined by eq.3:

$$F1 = 2.\frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

which provides a balanced measure that increases when both precision and recall are high.

For question answering tasks, we use exact match (EM) and F1 for span extraction and list questions; EM counts an answer as correct only if it matches the ground truth exactly, whereas F1 measures the overlap between predicted and true answer sets. We also report accuracy on multiple-choice question datasets. For factual probing tasks, we compute the precision at 1 (P@1), which is the percentage of queries for which the model assigns the highest probability to the correct answer; this is equivalent to accuracy in a top-1 ranking scenario. For retrieval-based evaluation, we use Mean Reciprocal Rank (MRR)**,** defined as eq.4:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (4)$$

where |Q| is the "number of queries "and $rank_i$ is the "position of the first relevant document in the list of retrieved documents for query i". A higher MRR indicates that relevant documents are ranked closer to the top. In decision tree contexts, we occasionally use Gini impurity, which measures the likelihood that a randomly chosen element would be incorrectly labelled if it were randomly labelled according to the distribution ppp. Gini impurity for a node with class probabilities P_i is defined in eq.5:

$$G = 1 - \sum_{i=1}^{n} P_i^2 \quad (5)$$

reflecting the degree of disorder or impurity. While this metric is not directly used in our experiments, it illustrates how classification diversity can be quantified.

Another useful measure when training adapters is the cross-entropy loss between a target distribution p and the model's predicted distribution q, given by eq.6:

$$H(p, q) = -\sum_{i=1}^{n} p_i \log q_i \quad (6)$$

which quantifies the expected number of bits needed to encode information from p when using a code designed for q. Minimizing cross-entropy encourages the model to align its predictions with the ground truth.

3.3. Retrieval module

The retrieval component aims to bring relevant domain knowledge into the system. We encode both the query q and each passage d in the corpus C_d using a dense encoder E, typically a

Received: August 02, 2025

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

fine-tuned sentence transformer. The query embedding is $e_q = E(q)$, and the document embeddings are $e_d = E(d)$. We compute the similarity between qqq and each passage using cosine similarity as in eq.7:

$$sim(q,d) = \frac{e_q . e_d}{||e_q|| ||e_d||} \quad (7)$$

where · is the "dot product" and ‖·‖ denotes the "Euclidean norm". The top-k passages with highest similarity scores are selected as the retrieved set K. After retrieval, we represent the collection of passages via an aggregation function. A simple yet effective aggregation is the weighted average of their embeddings shown in eq.8:

$$z = \frac{1}{k} \sum_{i=1}^{k} w_i e_d \quad (8)$$

where w_i may be proportional to the similarity scores. This aggregated vector captures the salient information from the retrieved texts.

3.4. Neural adapter design

To condition the LLM on retrieved knowledge, we introduce a small neural adapter A. Instead of altering the base model weights, we learn a low-rank transformation that maps the aggregated vector z to a conditioning signal hA . Following the concept of Low-Rank Adaptation (LoRA), the adapter applies a $rank\text{-}r$ update to the hidden representation. For each linear layer in the base model, the LoRA update takes the form as in eq.9:

$$\Delta W = BA \quad (9)$$

Where $A \epsilon\ \mathbb{R}^{d \times r}$ and $B \epsilon\ \mathbb{R}^{r \times d}$ are "learnable matrices", d is "dimesion of the hidden state and $r \ll d$". The adapter output is then expressed as eq.10:

$$h_A = f(z) = B(Az) \quad (10)$$

which produces a vector in the same dimensionality as the LLM hidden state. Because r is small, the additional parameters are minimal and can be trained efficiently. In our experiments we often freeze A and B after a brief training phase on a small dataset. During inference, we inject h_A into each transformer block via residual connections, allowing the model to attend to domain information.

3.5. Prompt construction

The final step is to assemble the enriched prompt for the LLM. We follow a template-based strategy. For a query q and the set of retrieved passages K, we construct a prompt P of the form represented as eq.11:

$$P = [TASK][Query: q][Evidence: \{k_1\} \dots \dots \{k_k\} \quad (11)$$

Where, [TASK] specifies the nature of the task, such as *Entity Typing* or *Answer the following question*. $\{k_i\}$ denotes the i^{th} retrieved passage from the knowledge set K.

The prompt can also include special tokens or explicit instructions guiding the model on how to utilize the evidence. To enhance conditioning, the prompt is augmented with an adapter vector. This is achieved by inserting a control token [ADAPTER], whose embedding is replaced with the adapter representation h_A.

Received: August 02, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Since LLMs are highly sensitive to token ordering, the [ADAPTER] token is placed close to the evidence section. This ensures that the adapter vector strongly influences the model's contextualization process and the subsequent reasoning steps.

3.6. Unified processing and multi-task support

The above components come together to form a unified processing algorithm. For each input, the system performs: (1) query encoding and retrieval; (2) adapter computation; (3) prompt assembly; and (4) LLM prediction. The same framework applies to various tasks by changing the retrieval corpus and the prompt template. For entity typing, the retrieved evidence typically consists of sentences containing the target entity; the model outputs entity type labels. For relation classification, retrieval focuses on sentences that mention both entities, and the adapter helps differentiate semantic roles. For question answering, retrieval gathers passages that likely contain the answer; the model then extracts or generates the answer. For factual probing, retrieval may be optional—yet including it helps correct hallucinations and align model predictions with reality. Finally, for domain-specific question answering (BioASQ), the corpus C_d comprises domain literature (PubMed abstracts); the adapter learns to highlight biomedical terms and relations.

4. Algorithm for proposed work

Algorithm 1 summarizes the inference process. Given a query q, we embed it, retrieve relevant passages, compute a low-rank adapter signal and construct a prompt for the LLM. The model then outputs the prediction. The algorithm-1 ensures that the base model remains unchanged and that domain customization happens at inference time.

ALGORITHM 1 — RANA INFERENCE -DOMAIN CUSTOMIZATION AT INFERENCE TIME

Require: query q; task type t; domain corpus c_d; encoder E; retriever R over index v; adapter A with bottleneck B; frozen LLM M; top=k; non-negative weights $\{w_i\}_{i=1}^{k}$

Ensure: task-specific prediction y

1. Encode the query

   $e_q \leftarrow E(q)$

2. Retrieve top-k evidence passages from the domain corpus

   $K = \{k_1, \ldots . . k_k\} \leftarrow R(v, e_q, k)$

3. Embed and aggregate evidence (weighted mean)

   For $i = 1 \ldots k; z_i \leftarrow E(k_i)$

   $$z \leftarrow \frac{1}{\sum_{i=1}^{k} w_i} \sum_{i=1}^{k} w_i z_i$$

4. Compute low-Rank adapter conditioning signal

   $h_A \leftarrow B(A_z)$

5. Build enriched prompt with adapter control token and evidence

Received: August 02, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$P \leftarrow COnstructPrompt\ (q, K, h_A, t\ )$

6. Forward the frozen LLM on the constructed prompt

$\hat{y}\ \leftarrow M(P)$

7. Return $y \leftarrow \hat{y}$

5. Results and output

We evaluate RANA on four general benchmark categories—entity typing, open-domain question answering, relation classification and factual probing—plus one domain-specific benchmark, BioASQ. The experimental setup compares RANA with baseline models: BERT-base, RoBERTa, K-Adapter and domain-specific variant.

5.1. Entity Typing ("OpenEntity, FIGER"): The micro-F1 scores show that RANA outperforms RoBERTa and K-Adapter by roughly two points on both OpenEntity and FIGER datasets. This gain indicates that the retrieval module successfully brings relevant context to the adapter, enabling the model to assign fine-grained types more accurately. The small gap between K-Adapter and RANA suggests that dynamic retrieval offers more precise signals than static adapter training as shown in the table-1.

Table 1 Entity Typing comparison Results

Model	**OpenEntity Micro F1 (%)**	**FIGER Micro F1 (%)**
BERT-base[23]	73.56	77.83
RoBERTa-base[24]	76.23	77.83
K-Adapter (F+L)[25]	77.61	80.54
Proposed RANA	**79.5**	**81.5**

5.2. Question Answering ("SearchQA, Quasar-T, CosmosQA"): On open-domain QA tasks, RANA improves exact-match and F1 scores over the strong baseline WKLM + ranking and RoBERTa multi-task models. The retrieval component identifies passages that directly answer the questions, while the adapter helps the model integrate these passages effectively. As a result, RANA increases the average EM by about 2 percentage points and the F1 by similar margins, reflecting better coverage and correctness in answers as shown in table-2:

Table 2 Question Answering Comparison Results

Model	**SearchQA EM (%)**	**SearchQA F1 (%)**	**Quasar-T EM (%)**	**Quasar-T F1 (%)**	**CosmosQA Acc (%)**
WKLM + Ranking	43.2	49.6	38.2	45.2	59.2
RoBERTa + Multi-task[7]	45.5	51.6	39.5	46.9	61.3
K-Adapter (F+L)[25]	46.5	53.1	41.2	48.3	63

Received: August 02, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Proposed RANA	**48.8**	**55**	**43.5**	**50.5**	**65.2**

5.3. Relation Classification ("TACRED"): In the relation extraction benchmark, RANA achieves the highest F1 among all compared models as in table 3. The precision and recall gains demonstrate that injecting retrieved sentences into the adapter helps the model disambiguate subtle relational cues. The improvement over K-Adapter shows that dynamic retrieval of context is advantageous even when static factual knowledge is already injected.

Table 3 Relation Classification Comparison Results

Model	Precision (%)	Recall (%)	F1 (%)
C-GCN	69.9	63.3	66.4
BERT-large	70.3	63.9	66.9
K-Adapter (F+L)	71.5	64.3	67.7
Proposed RANA	**73.2**	**65.8**	**69.3**

5.4. Factual Probing: The P@1 results reveal that RANA surpasses BERT-large and K-Adapter on the LAMA benchmark as in table 4). By conditioning the model on relevant facts, RANA reduces hallucinations and increases the likelihood of predicting the correct answer at rank 1. This improvement underscores the value of retrieval in aligning language model outputs with external knowledge.

Table 4 Factual Probing Comparison Results

Model	LAMA P@1 (%)
ELMo	8.6
BERT-large[26]	39.6
K-Adapter (F+L)	45.8
Proposed RANA	**48.2**

5.5. Biomedical Question Answering: To assess domain-specific adaptation, we evaluate on BioASQ factoid questions as in table 5. RANA delivers significantly higher strict accuracy, lenient accuracy and MRR than BioBERT and BioM-ELECTRA. The result suggests that RANA's dynamic retrieval and adapter conditioning are particularly beneficial in domains with rapidly evolving terminology and specialized knowledge. The improvements also confirm that the methodology scales effectively beyond general NLP tasks.

Table 5 Domain-Specific Evaluation

Model	Strict Acc (SAcc)	Lenient Acc (LAcc)	MRR	Factoid F1
BioBERT[27]	0.15	0.25	0.19	–
BioM-ELECTRA[28]	–	–	0.4804	0.3947
Proposed RANA	**0.3**	**0.4**	**0.55**	**0.45**

Received: August 02, 2025

International Journal of Applied Mathematics

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6. Conclusion and future scope

The Proposed research presents Retrieval-Augmented Neural Adapters (RANA), a unified framework for infusing domain-specific knowledge into large language models without full fine-tuning. By combining dense retrieval, low-rank adapters and prompt construction, RANA dynamically integrates external evidence into a frozen LLM, allowing it to adapt to diverse tasks and domains. Experimental results on entity typing, relation classification, open-domain and biomedical question answering and factual probing show that RANA consistently outperforms strong baselines, including K-Adapter and domain-specialized models. The gains stem from the ability of retrieval to provide current, relevant information and the efficiency of low-rank adapters to condition the model without altering its core parameters.

The future scope of this work includes exploring adaptive retrieval techniques that select not only passages but also relevant entities or tables, and investigating multi-adapter ensembles where different adapters capture complementary aspects of domain knowledge. Another promising direction is to extend RANA to multimodal settings by retrieving images or structured data alongside text and conditioning the model with cross-modal adapters. Finally, developing robust evaluation protocols for domain adaptation will help refine the methodology and better understand the interplay between retrieval and generation.

RANA offers a practical and effective solution for domain-specific customization of LLMs. The framework demonstrates that retrieval-augmented adaptation can bridge the gap between static pre-training and dynamic domain requirements, enabling language models to remain current, accurate and versatile across applications. Through the synergy of retrieval and neural adapters, this research charts a path toward scalable, efficient and interpretable knowledge injection for next-generation language systems.

References

[1] A. Salemi, S. Kallumadi, and H. Zamani, “Optimization Methods for Personalizing Large Language Models through Retrieval Augmentation,” *SIGIR 2024 - Proc. 47th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 752–762, 2024, doi: 10.1145/3626772.3657783.

[2] A. Kuppa, J. Nicholls, and N. A. Le-Khac, “Manipulating Prompts and Retrieval-Augmented Generation for LLM Service Providers,” *Proc. Int. Conf. Secur. Cryptogr.*, no. Secrypt, pp. 777–785, 2024, doi: 10.5220/0012803100003767.

[3] B. Lin, S. Wang, Y. Qin, L. Chen, and X. Mao, *Give LLMs a Security Course: Securing Retrieval-Augmented Code Generation via Knowledge Injection*, vol. 1, no. 1. Association for Computing Machinery, 2025.

[4] K. Bhushan *et al.*, “Systematic Knowledge Injection into Large Language Models via Diverse Augmentation for Domain-Specific RAG,” pp. 5922–5943, 2025, doi: 10.18653/v1/2025.findings-naacl.329.

[5] S. Wang *et al.*, “Knowledge Graph Retrieval-Augmented Generation for LLM-based Recommendation,” pp. 27152–27168, 2025, doi: 10.18653/v1/2025.acl-long.1317.

[6] R. Ren *et al.*, “Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation,” *Proc. - Int. Conf. Comput. Linguist. COLING*, vol. Part F206484-1, pp. 3697–3715, 2025.

[7] M. E. Peters *et al.*, “Deep contextualized word representations,” *NAACL HLT 2018 -*

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, pp. 2227–2237, 2018, doi: 10.18653/v1/n18-1202.

[8] L.-C. Chen, M. S. Pardeshi, Y.-X. Liao, and K.-C. Pai, "Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model," *Comput. Stand. Interfaces*, vol. 94, p. 103995, 2025, doi: https://doi.org/10.1016/j.csi.2025.103995.

[9] X. Li, Z. Zheng, M. Zhao, Y. Zhao, L. Shi, and B. Wang, "RLFE-IDS: A framework of Intrusion Detection System based on Retrieval Augmented Generation and Large Language Model," *Comput. Networks*, vol. 268, no. March, p. 111341, 2025, doi: 10.1016/j.comnet.2025.111341.

[10] J. Sun, W. Shi, X. Shen, S. Liu, L. Wei, and Q. Wan, "Multi-objective math problem generation using large language model through an adaptive multi-level retrieval augmentation framework," *Inf. Fusion*, vol. 119, no. November 2024, p. 103037, 2025, doi: 10.1016/j.inffus.2025.103037.

[11] G. Chen *et al.*, "Knowledge graph and large language model integration with focus on educational applications: A survey," *Neurocomputing*, vol. 654, no. August, 2025, doi: 10.1016/j.neucom.2025.131230.

[12] J. Jiang *et al.*, "Knowledge assimilation: Implementing knowledge-guided agricultural large language model," *Knowledge-Based Syst.*, vol. 314, no. February, p. 113197, 2025, doi: 10.1016/j.knosys.2025.113197.

[13] K. Du *et al.*, "refinement IRAGKR : Iterative Retrieval Augmented Generation with Fine-grained Knowledge Refinement," *Neurocomputing*, p. 131282, 2025, doi: 10.1016/j.neucom.2025.131282.

[14] P. Minh Nguyen, T. D. Do, and M. Le Nguyen, "Improving hierarchical semantic parsing with LLMs: Demonstration selection and chain-of-thought prompting via semantic fragment decoding," *Knowledge-Based Syst.*, vol. 328, no. August, p. 114256, 2025, doi: 10.1016/j.knosys.2025.114256.

[15] M. A. Ferrag *et al.*, "Generative AI in cybersecurity: A comprehensive review of LLM applications and vulnerabilities," *Internet Things Cyber-Physical Syst.*, vol. 5, no. February, pp. 1–46, 2025, doi: 10.1016/j.iotcps.2025.01.001.

[16] K. X. Guo, P. K. Y. Wong, J. C. P. Cheng, C. F. Chan, P. H. Leung, and X. Tao, "Enhancing visual-LLM for construction site safety compliance via prompt engineering and Bi-stage retrieval-augmented generation," *Autom. Constr.*, vol. 179, no. August, p. 106490, 2025, doi: 10.1016/j.autcon.2025.106490.

[17] M. K. Ghali, A. Farrag, D. Won, and Y. Jin, "Enhancing knowledge retrieval with in-context learning and semantic search through generative AI," *Knowledge-Based Syst.*, vol. 311, no. November 2024, p. 113047, 2025, doi: 10.1016/j.knosys.2025.113047.

[18] Y. Feng, T. H. Chan, G. Yin, and L. Yu, "Democratizing large language model-based graph data augmentation via latent knowledge graphs," *Neural Networks*, vol. 191, no. November 2024, p. 107777, 2025, doi: 10.1016/j.neunet.2025.107777.

[19] J. A. Heredia Álvaro and J. G. Barreda, "An advanced retrieval-augmented generation system for manufacturing quality control," *Adv. Eng. Informatics*, vol. 64, no. November

Volume 38 No. 5s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2024, 2025, doi: 10.1016/j.aei.2024.103007.

[20] W. Yang, L. Some, M. Bain, and B. Kang, "A comprehensive survey on integrating large language models with knowledge-based methods," *Knowledge-Based Syst.*, vol. 318, no. January, p. 113503, 2025, doi: 10.1016/j.knosys.2025.113503.

[21] W. Su *et al.*, "Parametric Retrieval Augmented Generation," *SIGIR 2025 - Proc. 48th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 1240–1250, 2025, doi: 10.1145/3726302.3729957.

[22] W. Fan *et al.*, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 6491–6501, 2024, doi: 10.1145/3637528.3671470.

[23] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[24] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," no. 1, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692.

[25] R. Wang *et al.*, "K-ADAPTER: Infusing Knowledge into Pre-Trained Models with Adapters," *Find. Assoc. Comput. Linguist. ACL-IJCNLP 2021*, pp. 1405–1418, 2021, doi: 10.18653/v1/2021.findings-acl.121.

[26] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, "Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model," *8th Int. Conf. Learn. Represent. ICLR 2020*, pp. 1–13, 2020.

[27] J. Lee *et al.*, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020, doi: 10.1093/bioinformatics/btz682.

[28] "GitHub - google-research/electra: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." [Online]. Available: https://github.com/google-research/electra.

Received: August 02, 2025