2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 979-8-3503-8622-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/BIBM62325.2024.10822374

Medical Document Embedding Enhancement with Heterogeneous Mixture-of-Experts

Xiangyang Liu[1,2], Yi Zhu[1], Tianqi Pang[1], Kui Xue[2], Xiaofan Zhang[2,3], and Chenyou Fan[1,*]

[1]South China Normal University, Guangdong, China
[2]Shanghai AI Lab, Shanghai, China
[3]Shanghai Jiao Tong University, Shanghai, China
2022024952@m.scnu.edu.cn, xiaofan.zhang@sjtu.edu.cn, fanchenyou@scnu.edu.cn

***Abstract*—Retrieval-Augmented Generation (RAG) has emerged as a crucial technique to enhance the accuracy and reliability of large language models, particularly in specialized domains like medicine. However, the effectiveness of RAG heavily depends on the quality of text embeddings used for retrieval. In this paper, we introduce Med-MoE-Embed, a novel approach to improve medical text embeddings tasks. Med-MoE-Embed leverages a pretrained embedding backbone augmented with a trainable Mixture of Experts (MoE) network, allowing for efficient adaptation to specific medical subdomains and tasks. We design each expert to be a compact KANs or a MLP with heterogeneous activation functions such as GELU and SWIGLU. Furthermore, we propose a two-step fine-tuning process that optimizes expert training and selection, enhancing the model's adaptability across various medical datasets. Our extensive evaluation focuses on RAG tasks in the medical domain, demonstrating significant improvements in retrieval accuracy and generation quality. Med-MoE-Embed mitigates the challenges of limited data accessibility and domain-specific requirements in the medical field, offering a versatile and efficient solution for enhancing embedding quality in medical natural language processing applications.**

Keywords: Text Embedding Model, Mixture of Experts, Retrieval-Augmented Generation, Medical Document Retrieval

I. Introduction

The rapid development of large language models (LLMs), such as GPT-4 [26] and Deepseek-V2 [3], has significantly advanced natural language processing (NLP) capabilities, enabling impressive performance in tasks like question answering, text generation, and summarization. However, LLMs have limitations, particularly their tendency to generate plausible-sounding but factually incorrect information, a phenomenon commonly referred to as "hallucination" [11].

To mitigate this issue, Retrieval-Augmented Generation (RAG) [8], [18] techniques have been introduced. RAG leverages external knowledge sources to provide LLMs with relevant information during the inference process, thereby enhancing both accuracy and reliability. Nevertheless, the effectiveness of RAG is highly dependent on the relevance and quality of the retrieved information. In the medical domain, where knowledge is complex and constantly evolving, RAG has been widely adopted for its potential to keep models up-to-date. Nevertheless, the retrieval of inaccurate information can lead to harmful decisions, posing serious risks to patients and healthcare professionals. Therefore, we focus on improving the accuracy of RAG systems specifically in the medical domain to mitigate these risks and enhance the reliability of medical decision-making.

A key component in improving the retrieval process is the development of effective text embedding models. These models transform textual information into dense vector representations, which are crucial for accurately retrieving and ranking relevant documents. High-quality text embeddings ensure that more relevant information is accessed and utilized, thereby significantly enhancing the performance of RAG systems.

In the medical domain, text embedding models are used for tasks such as question-answering [2], [34], literature retrieval [33], [34], EHR analysis [7], [19], and clinical decision-making [2], [10]. These models improve information retrieval and data analysis, aiding in interpreting studies, supporting decisions, and enhancing patient care.

Our research aims to develop a text embedding model tailored for medical applications, improving the retrieval process in RAG systems to enhance the accuracy and reliability of language model outputs in medical settings.

The medical field consists of diverse subdomains, each with specific terminologies and data requirements. This diversity challenges general embedding models, which often struggle to capture the nuanced embeddings needed for specialized tasks. Additionally, strict privacy regulations restrict access to medical datasets, making it difficult to train a universal embedding model that serves all subdomains effectively.

To address these challenges, we propose a novel method called Med-MoE-Embed. Specifically, our method introduces a trainable MoE network module comprising multiple shared experts, multiple routed experts, and a gating network. During inference, shared experts are always activated, while the routed experts are selectively activated by the gating network. This design optimizes computational efficiency and enables targeted adaptation to specific tasks or datasets. The modularity of Med-MoE-Embed ensures that it can be fine-tuned on particular datasets or specialized tasks, providing tailored embeddings that align closely with domain-specific requirements.Additionally, the modularity of Med-MoE-Embed

* Corresponding Author.

allows for easy integration with existing embedding backbones, enhancing task-specific performance without requiring full parameter fine-tuning. This adaptability makes Med-MoE-Embed a highly effective solution for addressing the challenges posed by the diverse and specialized nature of medical data.

We summarize our contributions in this study as follows.

1) We propose a trainable MoE module called Med-MoE-Embed to enhance the capabilities of pretrained embedding backbones on private datasets and specific tasks.
2) We designed several compact MLP and KANs models as shared experts and routed experts to enable efficient task specialization and enhanced adaptability.
3) We propose a two-step fine-tuning method that ensures the experts within Med-MoE-Embed are fully trained and optimally selected.
4) We extensively evaluated the performance of Med-MoE-Embed in RAG tasks, demonstrating its effectiveness in this crucial application area within the medical domain.

II. Related Work

A. Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) [18] is a method that merges retrieval and generation models to create more informed text outputs. Unlike generative models like GPT-4 [26], which depend on internal knowledge, RAG accesses external knowledge bases for up-to-date or specialized information. The process involves using a retrieval model to find relevant documents, which are then inputted into a generative model to produce detailed answers. This approach has been successfully applied in various NLP tasks, including question answering systems [21], [32] and text generation tasks [37], demonstrating improved accuracy and relevance compared to generative models alone. RAG's adaptability allows it to be tailored to different domains and tasks, by altering the retrieval database, making it a versatile tool in the field of NLP.

B. Text Embedding Models

Text embedding models have seen significant advancements, with foundational models like BERT [4] and Sentence-BERT [29] demonstrating substantial improvements in tasks such as semantic similarity and clustering. Other notable contributions include Sentence T5 [24], which produces high-quality, general-purpose embeddings. Despite these advancements, embedding models often struggle to generalize across different tasks and domains. This limitation has prompted the development of unified models and benchmarks, such as the Massive Text Embedding Benchmark (MTEB) [23], which evaluates models on novel tasks and domains.

Recent efforts have focused on leveraging large, diverse datasets, such as the LLM-generated FRet dataset [16]. State-of-the-art models are typically fine-tuned on supervised data to enhance performance on downstream tasks. Notably, NV-Embed [15], which uses only publicly available data, has achieved a record-high score of 69.32, securing the top position on the MTEB benchmark.

C. Mixture-of-Experts

Mixture of Experts (MoE) models are widely used for their ability to scale model capacity efficiently. Initially proposed by Jacobs et al. [9] and refined by Jordan et al. [14], MoE models use a gating mechanism to route inputs to different experts, enhancing specialization while reducing interference.

In the domain of natural language processing, Shazeer et al. [31] incorporated an MoE layer into LSTM architectures, achieving strong results in NLP tasks like language modeling and machine translation. GShard [17] improved multilingual translation using sparse gating and automatic sharding.

Recent models like GLaM [5] have scaled MoE models to trillion parameters while optimizing training and inference costs. DeepSeek-V2 [3] further advances MoE by segmenting experts for greater specialization and isolating shared experts to reduce redundancy. Building on DeepSeek-V2, we propose the Med-MoE-Embed module, utilizing shared experts for comprehensive knowledge and routed experts for specific data.

III. Approach

In this section, we focus on introducing our proposed approach. We illustrate the overview of Med-MoE-Embed in Figure 1. Specifically, we first process the medical text through the Embedding backbone to generate intermediate embeddings. Med-MoE-Embed then refines these embeddings, producing more accurate representations that can be tailored to specific datasets and specialized tasks. As a result, downstream applications benefit from these enhanced embeddings, including literature retrieval, EHRs analysis, decision-making support, question-answering systems, and medical education and so on.

A. Shared MoE

In conventional MoE models, experts often learn redundant knowledge, causing parameter inefficiency. We address this by using shared experts to capture comprehensive knowledge, assigned deterministically to embeddings. In our proposed MoE, the shared experts are responsible for learning more general knowledge and are pretrained on comprehensive datasets. The routed experts focus on learning more specialized domain-specific knowledge. By combining the comprehensive abilities of the shared experts with the specialized expertise of the routed experts, we enhance the capability of our MoE model.

Let $\mathbf{v}$ be the output of the embedding model. We compute the output $\mathbf{h}$ of the MoE layer as follows:

$$\mathbf{h} = \sum_{i=1}^{N_s} \mathrm{E}_i^{(s)}(\mathbf{v}) + \alpha \sum_{i=1}^{N_r} g_i \mathrm{E}_i^{(r)}(\mathbf{v}), \tag{1}$$

where N_s and N_r denote the numbers of shared experts and routed experts, respectively; $\mathrm{E}_i^{(s)}(\cdot)$ and $\mathrm{E}_i^{(r)}(\cdot)$ denote the i-th shared expert and the i-th routed expert, respectively; and α is a trainable parameter updated during training.

The gate value g_i for the i-th routed expert is defined as:

$$g_i = \begin{cases} s_i, & s_i \in \mathrm{Topk}(\{s_j \mid 1 \le j \le N_r\}, K_r), \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

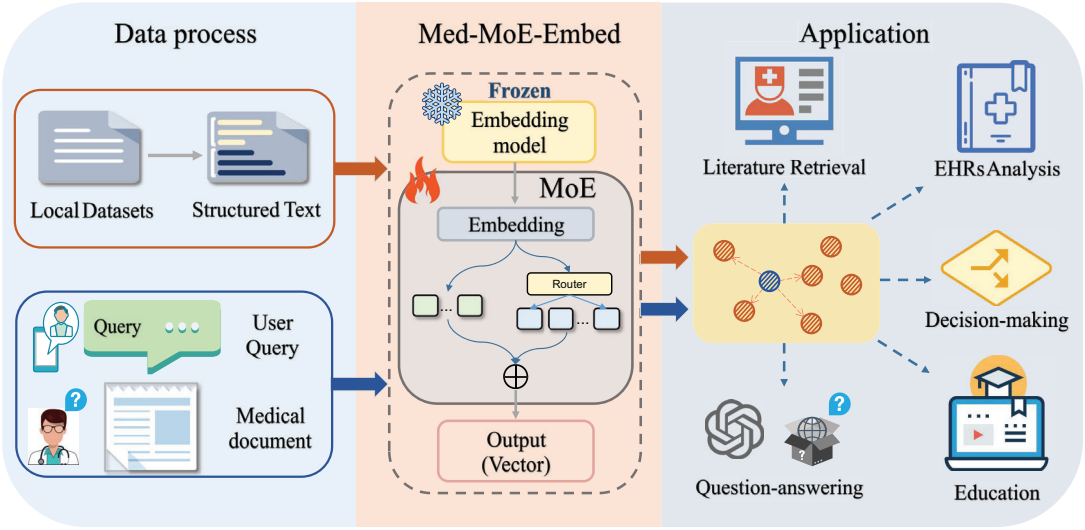

Fig. 1: **The Overview of Med-MoE-Embed.** Med-MoE-Embed enhances the embedding backbone's adaptability to sensitive medical data and improves its representational capabilities across various downstream tasks through the training of a MoE network module.

where K_r denotes the number of activated routed experts.

The token-to-expert affinity score s_i is computed as $s_i = \text{Softmax}_i\left(\mathbf{v}^T \mathbf{e}_i\right)$, where $\mathbf{e}_i$ is the centroid of the i-th routed expert in this layer. The function $\text{Topk}(\cdot, K)$ denotes the set containing the K highest scores among the affinity scores calculated for the embedding output $\mathbf{v}$ and all routed experts.

B. The design of expert architecture in MoE

We use a compact MLP that expands an input from h dimensions to $4h$ dimensions via a linear transformation, followed by a non-linear activation to capture complex patterns. The features are then projected back to h dimensions through another linear transformation. Based on this structure, we designed four distinct network architectures as our experts.

MLP-SWIGLU SwiGLU is a variant of the Gated Linear Unit where the gating mechanism employs the Swish activation function instead of the traditional sigmoid function. This variant is designed to provide more flexibility and adaptivity in gating the flow of information within neural networks. The SwiGLU function is defined as $\text{SwiGLU}_\beta(x, W, V, \beta) = \text{Swish}_\beta(xW) \cdot (xV)$, where $\text{Swish}(x) = x \cdot \sigma(x)$. In this formulation, x represents the input to the SwiGLU function, while W and V are learnable weight matrices that allow the unit to adapt during training. $\sigma(x)$ is the sigmoid function. In our model, we adopt the MLP-SwiGLU as a MoE expert, following recent advancements in LLMs.

MLP-GELU GELU (Gaussian Error Linear Unit) is an activation function that is often used to enhance the expressiveness of neural networks by smoothly blending the properties of linear and non-linear activation. We use GELU as the activation function of MLP as an expert. The GELU function is defined as $\text{GELU}(x) = x\Phi(x) = x \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$.

KAN Kolmogorov–Arnold Networks (KANs) [22] offer an innovative alternative to traditional MLPs by leveraging the Kolmogorov-Arnold representation theorem. Unlike MLPs, which use fixed activation functions at each node, KANs feature learnable activation functions applied along the edges. This approach replaces the linear weight parameters typically used in MLPs with univariate functions that are parameterized as splines.

We set a compact KANs as one of our MoE routed experts. Specifically, the KANs structure we designed is similar to the previous MLP. It takes an input with h hidden dimensions, projects it to a 4h-dimensional space, and then projects it back to the original h-dimensional space.

MLP-KAN The core feature of KANs lies in their placement of learnable activation functions on the edges of the network (i.e., the weights), rather than on the nodes. Therefore, we replace the activation function in the MLP with a KAN layer that has the same input and output dimensions, serving as one of the experts.

For simplicity, we employ a MoE with a shared expert and four routed experts in the experiments. The shared expert is an MLP-SWIGLU model, while the routed experts consist of MLP-SWIGLU, MLP-GELU, KAN, and MLP-KAN models. However, the actual number of shared and routed experts can vary depending on the specific requirements.

C. Loss Function

In this paper, we employ contrastive learning to fine-tune Med-MoE-embed, ensuring that queries are embedded closer to positively related documents and further from negatively related documents within the embedding space. To realize the objectives of contrastive learning, we adopt the InfoNCE Loss [25], a prevalent loss function in contrastive learning. We adapt this loss function to suit the specific requirements of our task, resulting in the formulation of a tailored loss function. The equations for this loss function are presented as Eq. 3.

$$L(W) = -\log \frac{e^{\text{sim}(q,d_i^+)}}{\sum_{i=1}^{m} e^{\text{sim}(q,d_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q,d_j^-)}}, \quad (3)$$

where $\text{sim}(q, d)$ denotes the cosine similarity, d_i^+ and d_j^- are the positive and negative samples of q in the current batch.

D. Training Methodology

In our MoE framework, since experts are inherently different, we set the load balancing loss to a smaller value. This prevents the model from overusing specific experts while ensuring tasks are appropriately assigned to the most suitable experts, enhancing training efficiency.

Our method involves a two-step training approach. First, we pretrain shared experts independently and load them into the MoE. In the initial phase, the gating network randomly distributes input data to different experts, ensuring that each expert is adequately trained early on. In the second phase, normal training resumes, with the gating network learning to assign tasks to the most suitable experts until convergence. We find this method helps maintain a balanced training process.

IV. Experiment

Our evaluation focuses on the performance of Med-MoE-Embed in RAG tasks as a representative and challenging application in medical natural language processing. We demonstrate its effectiveness across various downstream tasks that rely on high-quality embeddings in the medical domain.

A. Datasets

The MPD dataset [28] has 1,000 curated papers from NCBI, resulting in 886 retained papers and 79,966 entries, split into MPD-Title and MPD-RP. The MMD dataset [28] includes over 200,000 drug records from "WHO Medicine" and "National Pharmacopoeia," with relevance assessments for 100 medical questions, divided into 573 training and 205 test instances. PubMedQA [12] features 1,000 expert-annotated QA instances and 211,300 generated instances from PubMed abstracts. MedMCQA [27] has over 194,000 multiple-choice questions from India's AIIMS and NEET PG exams, covering more than 2,400 healthcare topics across 21 specialties.

We created the MCQA-Retri dataset from MedMCQA to assess retrieval capabilities. We extracted medical questions and labeled their answer explanations as positive examples. For negative examples, we classified questions by subject and used the BM25 [30] retrieval algorithm to rank answer explanations, selecting those ranked 6th to 10th. After removing overly short explanations, the final dataset contains 182,822 training samples and 4,183 test samples, each with a question, one positive example, and five negative examples.

B. Embedding Backbones

We utilize five different embedding models as backbones: GTE-base-en-v1.5, GTE-base-zh, M3E, E5-base-v2 and PubMedBert-base-embedding, all of which output 768-dimensional embeddings.

GTE models [20] are based on the BERT framework and support both Chinese and English. M3E [36] is a robust model trained using the UniEM framework. It has been extensively evaluated on the MTEB-zh benchmark and trained on over 22 million Chinese sentence pairs. E5 [35] is a universal text embedding model known for its adaptability across retrieval, clustering, and classification tasks. PubMedBert-base-embedding [6] is an embedding model based on the PubMedBERT model trained on medical datasets.

C. Baselines and Settings

For the "base" method, we directly use the pre-trained embedding model to encode both the query and document, performing the retrieval task via Faiss [13]. In the "FPFT" (Full Parameter Fine-Tuning) method, we fine-tune the embedding model on datasets. The "GELU/SWIGLU" method involves processing the output of the embedding model through our proposed MLP with different activation functions to obtain an optimized embedding representation, followed by fine-tuning the MLP. In the experimental section, we refer to our method, "Med-MoE-Embed", as MME for simplicity.

D. Evaluation Metrics

We use various evaluation metrics, including **Precision, Recall, F1 Score, mAP, and Accuracy**. Precision measures the proportion of relevant documents among the retrieved ones, while Recall calculates the proportion of relevant documents that are successfully retrieved. The F1 Score provides a harmonic mean of these two metrics, giving equal weight to both precision and recall. Additionally, we use mAP to evaluate the ranking quality, and Accuracy to measure the overall correctness of the retrieved results.

E. Results and Analysis

TABLE I: Performance evaluation of different methods on MPD (Top-$K = 10$).

	Recall	Precision	F1 Score	mAP	Accuracy
MPD-Title					
E5-base-v2 [28]	0.212	0.526	0.302	0.541	0.545
E5-GELU	0.340	0.835	0.483	0.890	0.651
E5-SWIGLU	0.295	0.704	0.416	0.763	0.611
E5-MME-Title	**0.346**	**0.915**	**0.502**	**0.917**	**0.680**
GTE-en-base	0.210	0.514	0.298	0.566	0.542
GTE-en-GELU	0.315	0.828	0.456	0.840	0.645
GTE-en-SWIGLU	0.330	0.866	0.478	0.899	0.662
GTE-MME-Title	**0.363**	**0.978**	**0.529**	**0.984**	**0.703**
MPD-RP					
E5-base-v2	0.229	0.562	0.325	0.600	0.566
E5-GELU	0.336	0.829	0.478	0.877	0.647
E5-SWIGLU	0.328	0.823	0.469	0.864	0.664
E5-MME-RP	**0.352**	**0.869**	**0.501**	**0.902**	**0.676**
GTE-en-base	0.227	0.555	0.322	0.581	0.565
GTE-en-GELU	0.319	0.785	0.454	0.787	0.653
GTE-en-SWIGLU	0.338	0.830	0.480	0.828	0.667
GTE-MME-RP	**0.367**	**0.922**	**0.525**	**0.921**	**0.686**

In Table I, we present the performance of E5-base-v2 and GTE-base-en-v1.5 on the MMD datasets using several

evaluation metrics, focusing on the Top-10 retrieval results. The metrics include Recall, Precision, F1 Score, mAP and Accuracy as described in Section IV-D. From the table, we can draw the following conclusions:

Our method significantly enhances the retrieval capability of text embedding models. As shown in the table, our approach achieves approximately a 2 to 5.1% increase in F1 Score, a 2.7% to 9.3% improvement in mAP, and a 2.9% to 4.1% increase in Accuracy. These results demonstrate the robustness of our method across various evaluation metrics.

GTE-MME performs best on all evaluation metrics. From the table, we can see that GTE-MME demonstrates the best performance across all metrics. Compared to E5-MME, GTE-MME exhibits superior results, likely due to GTE's robust performance capabilities established during pretraining. Given that the embedding backbone in our method can be easily replaced, our approach offers great flexibility and adaptability, further illustrating its broad applicability.

TABLE II: Performance evaluation of different methods on MMD (Top-$K = 10$).

Setting	Recall	Precision	F1 Score	mAP	Accuracy
M3E	0.379	0.592	0.462	0.551	0.568
M3E-FPFT	0.372	0.254	0.302	0.318	0.231
M3E-GELU	0.531	0.514	0.522	0.569	0.617
M3E-SWIGLU	**0.575**	0.493	0.531	0.493	0.612
M3E-MME	0.477	**0.640**	**0.547**	**0.645**	**0.638**
GTE-zh-base	0.398	0.644	0.492	0.582	0.614
GTE-zh-GELU	0.423	0.648	0.512	0.588	0.617
GTE-zh-SWIGLU	0.467	0.647	0.542	0.592	0.615
GTE-zh-MME	**0.502**	**0.668**	**0.573**	**0.653**	**0.628**

In Table II, We show the performance of different settings of M3E and GTE-base-zh on the MMD dataset using several evaluation metrics, focusing on the Top-10 retrieval results.

MME method has a significant improvement effect on both M3E and GTE-zh models. As illustrated in Table II, our approach outperforms the second-best method, yielding enhancements in F1 Score ranging from 1.6% to 3.1%, mAP from 6.1% to 7.6%, and accuracy from 1.1% to 2.1%. Furthermore, our model attains the highest F1 Score, mAP, and Accuracy among all compared methods.

Table III shows the performance of different settings of E5 and GTE on the MCQA-Retri dataset using several evaluation metrics. From the table, we can see that the MME method also shows good performance in various metrics.

TABLE III: Performance evaluation(%) of different models on MCQA-Retri dataset (Top-$K = 10$)

Setting	Recall	Precision	F1 Score	mAP	Accuracy
E5-base	91.98	50.33	65.06	88.24	84.43
E5-MME	**92.10**	**59.66**	**72.41**	**90.95**	**84.58**
GTE-base	91.03	60.58	72.75	90.31	84.45
GTE-MME	**93.54**	**66.69**	**77.87**	**91.10**	**86.53**
PubMedBert	92.66	63.06	75.05	95.12	86.47
PubMedBert-MME	**95.30**	**78.56**	**86.12**	**97.35**	**91.91**

The MME method demonstrates enhanced performance on the MCQA-Retri dataset generated based on Medical QA dataset. As shown in Table III, models utilizing the MME approach achieved notable gains in F1 score, mean Average Precision (mAP), and accuracy. Specifically, the MME method led to an enhancement of approximately 5% to 11% in F1 score and a 1% to 2% increase in mAP. These improvements highlight the effectiveness of MME in enhancing the model's ability to understand and process complex medical questions, ultimately leading to more accurate and reliable results.

We extend our experimentation with the MME module to the MedMCQA dataset. In this study, we leverage explanations from the training set of MedMCQA and text from the PubMedQA dataset as the corpus. Before the large language model generates an answer, we first retrieve the most relevant text from the corpus as a one-shot prompt to help the large language model better answer the question. The results are shown in Table IV.

TABLE IV: Accuracy (%) of LLMs on the MedMCQA Dataset using different models

Corpus	MedMCQA		PubMedQA	
Setting/LLM	Llama3	Deepseek-V2	Llama3	Deepseek-V2
Zero-shot	49.87	53.64	49.87	53.64
E5-base	51.28	61.77	50.25	60.02
E5-MME	**52.57**	**62.73**	**52.11**	**60.46**
GTE-base	51.20	61.53	49.57	60.55
GTE-MME	**54.77**	**66.29**	**53.58**	**62.26**
PubMedBert	**57.96**	68.35	50.15	61.72
PubMedBert-MME	57.87	**69.72**	**52.96**	**63.79**

Our method can improve the quality of retrieved text, which can better enhance the reasoning accuracy of large language models. As shown in the Table IV, we used the training sets of MedMCQA and PubMedQA as the corpus to retrieve relevant texts. We tested the performance on the MedMCQA test set using two language models, Llama3-8B-Instruct [1] and DeepSeek-V2 [3]. The results indicate that the MME method improved retrieval-augmented generation.

Corpus relevance and model capability significantly impact the accuracy of question-answering systems. Using MedMCQA as the corpus results in substantial accuracy improvements, highlighting the advantages of a domain-specific dataset. In contrast, the PubMedQA dataset yields smaller gains, with some models showing slight performance decreases. Notably, Deepseek-V2 maintains stable performance on PubMedQA, likely due to its prior knowledge, which enhances answer accuracy.

F. Ablation

TABLE V: Perfomance of MME with multiple datasets

Setting	Recall	Precision	F1 Score	mAP	Accuracy
E5-MME-Title	0.346	0.915	**0.502**	**0.917**	**0.680**
E5-MME-RP	0.352	0.869	**0.501**	**0.902**	**0.676**
E5-MME(Title)	0.315	0.830	0.457	0.850	0.641
E5-MME(RP)	0.336	0.836	0.479	0.840	0.660
GTE-MME-Title	0.363	0.978	**0.529**	**0.984**	**0.703**
GTE-MME-RP	0.359	0.905	**0.514**	**0.917**	**0.686**
GTE-MME(Title)	0.343	0.936	0.502	0.950	0.678
GTE-MME(RP)	0.354	0.905	0.509	0.889	0.683

Multiple datasets We also evaluated the performance of our MoE method across multiple datasets. In Table V, E5-MME

and GTE-MME represent models that were jointly trained on the MPD-RP and MPD-Title datasets, while E5-MME-RP and E5-MME-Title indicate models trained individually on each of these datasets. In the table, we can observe that using the MME method enables the model to perform well across all datasets, achieving a balanced state between the two datasets.

TABLE VI: Performance evaluation of different Top-k Value

Top-k	Recall	Precision	F1 Score	mAP	Accuracy
10	0.367	**0.922**	0.525	**0.921**	0.686
15	0.494	0.915	0.642	0.917	0.729
20	0.601	0.877	0.713	0.912	**0.772**
30	**0.755**	0.706	**0.730**	0.897	0.701

Effect of different Top-k. We investigate the effect of varying Top-k values on different evaluation metrics on Table VI. This experiment was conducted on the MPD-RP dataset using the GTE-MME-RP setting. As the Top-k value increases from 10 to 30, Recall improves steadily, while Precision decreases. The F1 Score increases, and mAP slightly declines. Accuracy first rises, peaking at 77.2% for k=20, but drops to 70.1% at k=30. This indicates a trade-off between recall and precision as Top-k increases, with overall performance remaining balanced across different values.

V. Conclusions

We introduce an efficient medical document retrieval module Med-MoE-Embed comprised of a fixed pre-trained embedding model and a trainable MoE network. This enables pretrained embedding models to quickly adapt to private data and specific downstream tasks. The gating network is trained to select the most suitable expert for each task. We evaluate the performance of the Med-MoE-Embed module in RAG tasks, demonstrating its effectiveness. Future work can explore using larger embedding backbones and developing deeper, more sophisticated MoE networks, to achieve better results.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (Project 62106156), and the GuangDong Basic and Applied Basic Research Foundation (Project 2024A1515011650).

References

[1] AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
[2] Cai, L., Li, J., Lv, H., Liu, W., Niu, H., Wang, Z.: Integrating domain knowledge for biomedical text analysis into deep learning: A survey. Journal of Biomedical Informatics **143**, 104418 (2023)
[3] DeepSeek-AI, Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., et al., C.Z.: Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model (2024)
[4] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)
[5] Du, N., Huang, Y., Dai, A.M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A.W., Firat, O., et al.: Glam: Efficient scaling of language models with mixture-of-experts. In: International Conference on Machine Learning. pp. 5547–5569. PMLR (2022)
[6] Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., Poon, H.: Domain-specific language model pretraining for biomedical natural language processing (2020)
[7] Hossain, E., Rana, R., Higgins, N., Soar, J., Barua, P.D., Pisani, A.R., Turner, K.: Natural language processing in electronic health records in relation to healthcare decision-making: a systematic review. Computers in biology and medicine **155**, 106649 (2023)
[8] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Unsupervised dense information retrieval with contrastive learning. Trans. Mach. Learn. Res. **2022** (2022)
[9] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)
[10] Janowski, A.: Natural language processing techniques for clinical text analysis in healthcare. Journal of Advanced Analytics in Healthcare Management **7**(1), 51–76 (2023)
[11] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Computing Surveys **55**(12), 1–38 (2023)
[12] Jin, Q., Dhingra, B., Liu, Z., Cohen, W., Lu, X.: Pubmedqa: A dataset for biomedical research question answering. In: EMNLP. pp. 2567–2577 (2019)
[13] Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data **7**(3), 535–547 (2019)
[14] Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the em algorithm. Neural computation **6**(2), 181–214 (1994)
[15] Lee, C., Roy, R., Xu, M., Raiman, J., et al.: Nv-embed: Improved techniques for training llms as generalist embedding models (2024)
[16] Lee, J., Dai, Z., Ren, X., Chen, B., Cer, D., Cole, J.R., Hui, K., Boratko, M., Kapadia, R., et al.: Gecko: Versatile text embeddings distilled from large language models. arXiv preprint arXiv:2403.20327 (2024)
[17] Lepikhin, D., Lee, H., Xu, Y., Chen, D., et al.: Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668 (2020)
[18] Lewis, P., Perez, E., Piktus, A., Petroni, F., et al., K.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
[19] Li, I., Pan, J., et al.: Neural natural language processing for unstructured data in electronic health records: a review. Computer Science Review **46**, 100511 (2022)
[20] Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M.: Towards general text embeddings with multi-stage contrastive learning (2023)
[21] Liu, X., Pang, T., Fan, C.: Federated prompting and chain-of-thought reasoning for improving llms answering. In: International Conference on Knowledge Science, Engineering and Management (2023)
[22] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., et al.: Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024)
[23] Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: Mteb: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316 (2022)
[24] Ni, J., Abrego, G.H., Constant, N., Ma, J., Hall, K.B., Cer, D., Yang, Y.: Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. arXiv preprint arXiv:2108.08877 (2021)
[25] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
[26] OpenAI: Gpt-4 technical report. ArXiv **abs/2303.08774** (2023)
[27] Pal, A., Umapathi, L.K., Sankarasubbu, M.: Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In: Conference on health, inference, and learning. pp. 248–260. PMLR (2022)
[28] Pang, T., Tan, K., Yao, Y., Liu, X., Meng, F., Fan, C., Zhang, X.: Remed: Retrieval-augmented medical document query responding with embedding fine-tuning. IJCNN (2024)
[29] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
[30] Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. (2009)
[31] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q.V., Hinton, G.E., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. CoRR **abs/1701.06538** (2017)
[32] Siriwardhana, S., Weerasekera, R., et al.: Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. Trans. Assoc. Comput. Linguistics (Oct 2022)
[33] Sivarajkumar, S., Mohammad, H.A., Oniani, D., et al.: Clinical information retrieval: A literature review. Journal of Healthcare Informatics Research pp. 1–40 (2024)
[34] Tamine, L., Goeuriot, L.: Semantic information retrieval on medical texts: Research challenges, survey, and open issues. ACM Computing Surveys (CSUR) **54**(7), 1–38 (2021)
[35] Wang, L., Yang, N., et al.: Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533 (2022)
[36] Wang, Y., Sun, Q., He, S.: M3e: Moka massive mixed embedding model (2023), https://github.com/wangyuxinwhy/uniem
[37] Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H., Jiang, M.: A survey of knowledge-enhanced text generation. ACM Computing Surveys p. 1–38 (Jan 2022). https://doi.org/10.1145/3512467