OnRL-RAG: Real-Time Personalized Mental Health Dialogue System

Abstract. Large language models (LLMs) have been widely used for various tasks and applications. However, LLMs and fine-tuning are limited to the pre-trained data. For example, ChatGPT's world knowledge until 2021 can be outdated or inaccurate. To enhance the capabilities of LLMs, Retrieval-Augmented Generation (RAG), is proposed to augment LLMs with additional, new, latest details and information to LLMs. While RAG offers the correct information, it may not best present it, especially to different population groups with personalizations [1]. Reinforcement Learning from Human Feedback (RLHF) adapts to user needs by aligning model responses with human preference through feedback loops. In real-life applications, such as mental health problems, a dynamic and feedback-based model would continuously adapt to new information and offer personalized assistance due to complex factors fluctuating in a daily environment. Thus, we propose an Online Reinforcement Learning-based Retrieval-Augmented Generation (OnRL-RAG) system to detect and personalize the responding systems to mental health problems, such as stress, anxiety, and depression. We use an open-source dataset collected from 2028 College Students with 28 survey questions for each student to demonstrate the performance of our proposed system with the existing systems. Our system achieves superior performance compared to standard RAG and simple LLM via GPT-4o, GPT-4o-mini, Gemini-1.5, and GPT-3.5. This work would open up the possibilities of real-life applications of LLMs for personalized services in the everyday environment. The results will also help researchers in the fields of sociology, psychology, and neuroscience to align their theories more closely with the actual human daily environment.

1 Introduction

Large Language Models have grown increasingly into promising tools for diverse fields [21]. For example, LLMs play an important role in improving customer support through chatbots, which can understand queries and provide relevant answers in real time [20]. Additionally, LLMs can streamline tasks such as automated dialogue analysis in educational contexts, where they analyze classroom interactions to improve teaching effectiveness [14]. One of the important examples of using dialogue properties of LLMs is to measure how good a conversation is in chat systems. Researchers use these models to evaluate whether responses are relevant, meaningful and fully address questions or topics in the chat [8].

While Generative AI models can offer a great understanding of health problems, such as mental health, due to the fluctuations in a daily environment, offering personalized assistance is sometimes challenging and hard to achieve. For example, people with different demographic information may have various levels of exposure to mental health problems. We want to use LLMs with their dialogue properties to understand the differences among different groups of people. Once differences in patterns are found, they can be used to better understand the impact of personal characteristics, such as age, health conditions, and education, on daily routines. They can also be used to automate diagnoses. The results will also help researchers in the fields of sociology, psychology, and neuroscience to align their theories more closely with actual daily environments.

LLMs and fine-tuning are limited to the pre-trained data, which might be outdated or inaccurate. For example, ChatGPT's knowledge of the world is until 2021, and Gemini models' knowledge is until November 2023 [23]. To enhance the capabilities of LLMs, Retrieval-Augmented Generation (RAG), is proposed to augment LLMs with additional, new, latest details and information [7]. RAG facilitates dialogue between users and LLMs systems by combining information retrieval systems from the knowledge base with LLMs [27]. RAG has achieved excellent performance in many fields [7]. For example, in healthcare, they power smart chatbots that offer personalized medical advice during emergencies like COVID-19. These chatbots quickly pull relevant information from large databases, ensuring accurate answers, and even use voice technology to make interactions more natural and helpful for users [16]. Another example of RAG is in software development, where it enhances code completion [15] by retrieving relevant code snippets from large databases. This approach, inspired by how developers reuse code, improves accuracy and helps predict the next lines of code in languages like Python and Java, boosting productivity. Similarly, RAG is also used in automated form filling [5], where it combines LLMs with relevant data to accurately and efficiently complete tasks like loan applications using natural language. RAG can enhance the performance and accuracy of an open-domain question-answering system significantly [6] by retrieving relevant information from external sources and integrating it with the system's existing knowledge. RAG models excel in finance by enhancing decision-making. For instance, [30] combines data from news platforms (e.g., Bloomberg) and social media (e.g., Twitter) with queries to improve financial sentiment analysis.

However, RAG has limitations, such as content missing (i.e., questions can not be answered from available documents), not extracted here (i.e., due to noises in the context, LLMs fail to extract out the correct answer) [4]. For example, when the LLM-based dialogue tries to help Hispanic women students in STEM graduate programs, it may not find any highly related knowledge in the knowledge base due to a lack of represented information [10].

In this paper, we develop a personalized dialogue system, Online Reinforcement Learning-based Retrieval-Augmented Generation (OnRL-RAG) system, to adapt to user needs. OnRL-RAG would use RL to compare the similarity between the generated response and the ground truth, based on the user's demographic information and query details. The accumulated reward function in RL would also help the system for self-improvement and learning over time. In addition, given the back-and-forth properties in a dialogue, feedback-based systems (such as reinforcement learning) would offer better personalized information. We use an open-source dataset MHP (Mental Health Problems), survey-based responses [24] as shown in Table 1, to demonstrate the performance of the proposed method

2 Related Work

Recent developments in Generative AI and large language models have made the approach of Retrieval-Augmented Generation, introduced by [13], a favored one for answering complex questions with context. RAG combines a retrieval model that finds relevant data with an LLMs that uses this data to generate responses, reducing the risk of generating unsupported or inaccurate information [29]. Despite its strengths, RAG still faces challenges. For example, it can retrieve irrelevant or low-quality information, leading to errors in the final response.

Researchers have developed various methods to offer improvements. For example, researchers propose fine-tuned pre-trained models to focus on specific tasks, making them better at finding precise answers, such as detailed impacts of climate change in a particular region rather than generic answers [12]. Similarly, researchers propose, Iter-RetGen [18], to iterative retrieval and generation by considering all retrieved knowledge as a whole, then refines answers by iterating between the retrieval of better data and the update of the model's response to ensure highly relevant and precise outputs. Moreover, although RAG provides accurate information, it may not present it effectively, particularly when personalization is required for different population groups [1]. Reinforcement Learning from Human Feedback (RLHF) adapts to user needs by aligning model responses with human preference through feedback loops. Researchers also propose Self-RAG with offline Reinforcement Learning to refine results further by integrating feedback based on a trained critic model [3].

However, in real-life applications, such as mental health problems, due to complex factors fluctuating in a daily environment, a dynamic and feedback-based model would continuously adapt to new information and offer personalized assistance. While there are performance gaps between offline RL and online RL [25], especially for real-time problems, we propose an Online Reinforcement Learning-based RAG (OnRL-RAG) designed to retrieve highly relevant and similar information. The goal is to provide real-time personalized dialogues, especially for mental health support, and to enhance the performance of dialogue systems.

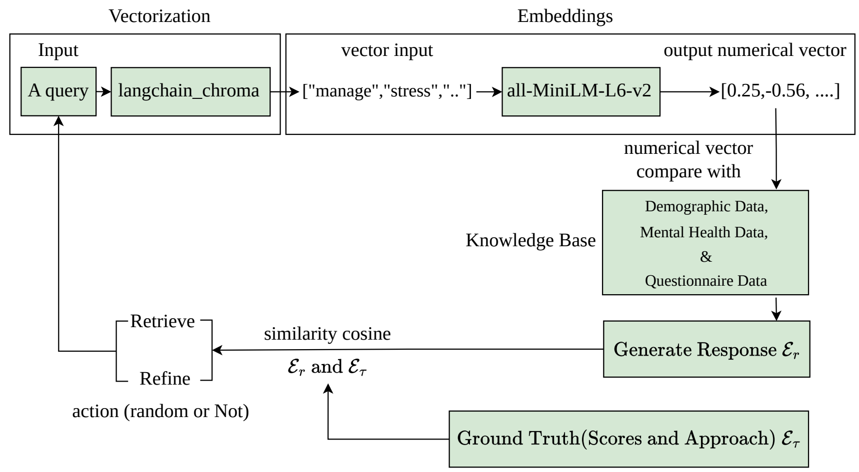

Fig. 1: Overall Flowchart of RAG and RL enhanced RAG

3 Methodology

This section discusses our development of the Online Reinforcement Learning-based Retrieval-Augmented Framework (OnRL-RAG). Retrieval-Augmented Framework(RAG) has been applied in diverse domains ranging from education to healthcare due to its ability to integrate external knowledge sources and enhance decision-making capabilities[26], [22]. We want to further expand the applications of RAG, especially in personalized healthcare, by proposing a novel OnRL-RAG.

Given the open-source dataset MHP, we first use the RAG framework to retrieve relevant information from the knowledge base and generate a response based on the retrieved context, focusing on issues such as stress, depression, or anxiety caused by irregular sleep, declining grades, or reduced social engagement. We discuss two steps: Step 1) Dataset Preprocessing and Knowledge Base Creation; Step 2) Vectorization and Query Processing. We then develop the proposed OnRL-RAG for personalized interventions using Online RL, specifically model-free Q-learning, to retrieve personalized information from the knowledge base. We define the state space, the action space, as well as the reward function. The outline of the proposed OnRL-RAG is shown in Figure 1 and Algorithm 1.

3.1 Retrieval-Augmented Framework for MHP Datasets

The development of a general AI model via RAG consists of data preprocessing and knowledge base creation, as well as vectorization and query processing. We are using a simple example to go through these two steps before employing the steps in the MHP dataset. For example, given a set of questions and the collected answers from all participating students, RAG would first create a knowledge base. Then, when a student from the knowledge base asks "How

Algorithm 1: OnRL-RAG for Personalized Responses

Input: knowledge base $\mathcal{K}$ stored in ChromaDB
Output: personalized response
Given an input case of query/question from a student
Input Query: "How can I manage stress during the final week before the exam?"
Input Query $\xrightarrow{\text{Chunking}}$ ["manage", "stress", "final week", "exams"]
Chunked Query $\xrightarrow{\text{Embedding}} \mathrm{I}_i = [0.25, -0.56, 0.78, 0.23]$
for *each embedded query* $\mathcal{I}_i$ **do**
  generate the initial response $\mathcal{R}_{\text{initial}}$ by retrieving information $\mathcal{C} \subset \mathcal{K}$;
  Convert $\mathcal{R}_{\text{initial}} \xrightarrow{\text{all-MiniLM-L6-v2 model}} \mathcal{E}_r$;
  Convert ground truth (assessed mental health scores and corresponding recovery approach) $\xrightarrow{\text{embedding}} \mathrm{E}_\tau$;
  calculate the cosine similarity between $\mathcal{E}_r$ and $\mathcal{E}_\tau$, $S = \cos(\theta) = \frac{\mathcal{E}_r \cdot \mathcal{E}_\tau}{\|\mathcal{E}_r\| \|\mathcal{E}_\tau\|}$;
  initialize Q-value to 0 for the initial state-action pair
  while *$S < 85\%$ (not converged) or less than maximum input queries* **do**
    select an action a_t:

$$a_t = \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \max Q(s_t, a) & \text{with probability } (1-\epsilon) \end{cases}$$

    $\epsilon \to$ small exploration factor (0.1) $\to$ probability of random action selection.
    if $a_t \to$ *"retrieve"* **then**
      $\mathcal{C}' \to$ Retrieve additional context;
      $\mathcal{C} \to \mathcal{C} + \mathcal{C}'$;
      $R^{t+1}_{\text{updated}} \to$ Generate response using $\mathcal{C}$;
    else if $a_t \to$ *"add details"* **then**
      $R_{\text{details}} \to$ Generate additional details;
      $R^{t+1}_{\text{updated}} \to R_{\text{current}} + R_{\text{details}}$;
    update state: updated state s_{t+1} is the updated response R^{t+1}_{updated};
    reward calculation: reward r is calculated as

$$R^{t+1}_{\text{updated}} \xrightarrow{\text{embed}} \mathcal{E}_r^{t+1},$$

$$S_{\text{next}} = \text{cosine}(\mathcal{E}_r^{t+1}, \mathcal{E}_\tau),$$

$$S_{\text{curr}} = \text{cosine}(\mathcal{E}_r^{t}, \mathcal{E}_\tau),$$

$$r = S_{\text{next}} - S_{\text{curr}} + \begin{cases} 0.5 & \text{if } S_{\text{next}} > S_{\text{curr}} \text{ (improved alignment)}, \\ -0.5 & \text{if } S_{\text{next}} < S_{\text{curr}} \text{ (reduced alignment)}, \\ 10 & \text{if } S_{\text{next}} = 1 \text{ (perfect similarity)}. \end{cases}$$

    Update Q-value:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

    Where:
    $\mathrm{Q}(\mathrm{s}_t, a_t)$: Current Q-value for state-action pair (initialized to 0 if not present)
    $\max_{a \in A} Q(s_{t+1}, a)$: Maximum Q-value for the next state s_{t+1} over all actions in action space.
    α is 0.1 (learning rate), and γ is 0.95 (discount factor).
return personalized response;

can I manage stress during the final week before the exam?", this question as a query would be converted into a vector with keywords, such as ["manage", "stress", "final week", "exams"]. After that, we follow two steps: first, we retrieve relevant information from the knowledge base; second, we generate an exact response using the retrieved information as context. Finally, we perform a similarity analysis by comparing the generated response with the ground truth response, which is based on the mental health assessment score and supporting recovery recommendations, focusing on mental health and academic success. We conduct a similarity analysis based on these two parts. Below we will talk about how we process the MHP dataset and create a knowledge base from the dataset.

Step 1: Dataset Preprocessing and Knowledge Base Creation. We use the newly collected open-source mental health dataset MHP [24]. MHP is a Google Forms-based questionnaire that collects responses from college students about their levels of anxiety, depression, and stress. For example, one of the questions in the questionnaire is, "Over the past two weeks, how often have you been bothered by feeling nervous, anxious, or on edge?". We denote the answers to those questions as *questionnaire data*. In this dataset, psychologists use collected information from questionnaires to calculate scores for different mental health problems. For example, General Anxiety Disorder-7 (GAD-7) is used for anxiety detection; Perceived Stress Scale (PSS-10) is used for stress detection, and Patient Health Questionnaire (PHQ-9) is used for depression detection [2]. We denote those calculated scores for different mental health problems/concerns as *mental health data*.

In addition, MHP also includes sociodemographic information about each participant, such as age, gender, and scholarship waiver status. We denote this information as *sociodemographic data*. A simple example of those three parts of data (questionnaire, mental health data, and sociodemographic data) is listed in Table 1. All three parts together of the entire training data would be defined as a ***knowledge base***.

Table 1: A Sample of MHP Dataset [24] for University Students

Survey Questionnaire	
Felt Nervous	Several Days
Upset Feeling	Nearly every day
Mental Health Data	
Anxiety Score (GAD-7)	12 (Moderate Anxiety)
Stress Score (PSS-10)	29 (High Perceived Stress)
Depression Score (PHQ-9)	20 (Severe Depression)
Sociodemographic Data	
ID	1
Age	18-22
Gender	Male
Current CGPA	3.00 - 3.39
Waiver or Scholarship	No

Step 2: Vectorization and Query Processing. We used ChromaDB to construct the ***knowledge base*** for the dataset, as outlined in Table 1. Since ChromaDB can store different types of data formats, including text and numerical values.

We use a simple example to go through Step 2. Given a query (such as a question from a student), we clean and simplify the text by breaking it down into key terms through chunking via `langchain_chroma` Python library. The `langchain_chroma` library is fast and easy to manage for tasks like searching and finding relevant information. For example, when a student asks, "How can I manage stress during the final week before the exam?", chuking would convert this question as a query into a vector with keywords and/or terms, ["manage", "stress", "final week", "exams"].

Then, we performed embeddings, converting the text vector into numerical vectors, all-MiniLM-L6-v2 model, due to its fast and accurate text understanding, balancing performance, and semantic clarity [28]. For example, an embedding result might look like $[0.25, -0.56, 0.78, 0.23]$. The all-MiniLM-L6-v2 model transforms text into dense vector embeddings, allowing for a nuanced comparison of semantic meaning rather than relying on simple token matches. The similarity score ranges from -1 (completely dissimilar) to 1 (highly similar).

We use a cosine similarity analysis to compare the numerical vector with the *knowledge base*. This is done using `utils` from `sentence-transformers` library in Python. For example, we use the function`util.pytorch_cos_sim(input1, input2)` to calculate the similarity between the input vector (denoted as input1) and stored embeddings (denoted as input2). To efficiently retrieve information from *knowledge base*, vectors with similar values are placed close to each other. The similarity is calculated by finding the dot product of the vectors and dividing it by the product of their magnitudes as shown in Equation 1. For example, if two vectors are $[0.25, -0.56, 0.78, 0.23]$ and $[0.24, -0.55, 0.77, 0.22]$, the cosine similarity would be 0.99 because their values are close, indicating that both vectors point in a similar direction in the vector space. On the other hand, if the input vector is $[0.25, -0.56, 0.78, 0.23]$ and $[0.95, 0.14, -0.89, 0.12]$, the cosine similarity would be -0.37, reflecting a larger angle between the vectors, and indicating they are not similar. ***Knowledge base*** consists of a high-dimensional vector space, with each entry represented as an embedding of a specific dimension (e.g., 768-dimensional for the all-MiniLM-L6-v2 model [17]).

For example, if a student from a knowledge base asks for mental health advice, the system can tailor the response according to the person's age, gender, or specific situation. For instance, it might recommend stress management techniques like prioritizing tasks and deep breathing exercises for a college student feeling overwhelmed with assignments, or relaxation methods such as mindfulness. Because similar queries, such as "feeling stressed about assignments" or "struggling to cope with academic pressure," are stored close together in the vector space, the system can quickly identify related contexts [19]. If relevant information is not available in the ***knowledge base***, the system can either generate an appropriate response based on available data or request additional input from the user to refine the search and improve response relevance.

3.2 OnRL-RAG

We start with background information on RL and then describe our development of OnRL-RAG (summarized in Figure 1 as well). Traditional RL is based on Markov Decision Process (MDP) with a tuple (S, A, T, γ, D, R), where S is a set of states; A is as a set of actions; $T = \{P(s_j|s_i, a_i)\}$ is a set of state transition probabilities (e.g. the probability of the next state s_j for executing an action a_i in the current state s_i, $\forall s_j, s_i \in S, a_i \in A$); $\gamma \in [0, 1)$ is a discount factor; D is a distribution over possible start states s_0; and $r(s_i, a_i)$ is a function specifying the reward that is received for executing an action a_i in the state s_i. Traditional RL is to maximize the expected sum of rewards with the discounted rate in order to find an optimized policy.

Existing RL can be broadly categorized into model-based approaches and model-free approaches. The former assumes that prior knowledge of system dynamics is provided, while the latter works on the assumption that such prior information is unknown. Reliable priors are not provided for many real-world problems, such as real-time dialogue systems [11]. Therefore, we adopted a model-free approach, online Q-learning, an optimal action-selection policy given the Markov decision process without the prior information of the transit matrix [9].

We employ Q-learning to select actions, prioritizing those with higher values, and employ an epsilon-greedy strategy for exploration. This process iterates to refine the RAG responses with real-time streaming queries continuously. OnRL-RAG enhances the response by choosing the best action, based on Q-values in the Q-table, to optimize the explanation. The agent can either retrieve additional relevant information from ***knowledge base*** or ask RAG to add more details to the response, ensuring high similarity.

In our OnRL-RAG, we define **state space** based on time. The current state s_t is the response from RAG at the current time t. The previous state s_{t-1} is the response from RAG at the previous time $t-1$. The next state is the response from RAG given the current state at time t and the action at the current state s_t. We define the **action space** with two actions: retrieve information from the *knowledge base* or refine the response by adding personalized details. The **accumulated reward function** is defined based on the similarity difference calculated by the cosine similarity. We first denote the cosine similarity as S, the RAG generated response as $\mathcal{E}_r$ and the ground truth data as $\mathcal{E}_\tau$. The cosine similarity S is shown in Equation 1. Then, we first calculate the similarity, denoted as S_{curr}, between the ground truth and the current response $\mathcal{E}_r^t$ at the current state s_t. We also calculate the similarity, denoted as S_{next}, between the ground truth and the next response $\mathcal{E}_r^{t+1}$ at the next state s_{t+1}, given state s_t and the action a_t). The reward function is defined by the difference between S_{next} and, S_{curr} shown in Equation 4. The agent updates its Q-values over time to refine its decisions. The Q-value is defined in Equation 5.

$$S = \cos(\theta) = \frac{\mathcal{E}_r \cdot \mathcal{E}_\tau}{\|\mathcal{E}_r\| \|\mathcal{E}_\tau\|}. \tag{1}$$

$$S_{\text{next}} = \text{cosine}(\mathcal{E}_r^{t+1}, \mathcal{E}_\tau), \tag{2}$$

$$S_{\text{curr}} = \text{cosine}(\mathcal{E}_r^{t}, \mathcal{E}_\tau), \tag{3}$$

$$r = S_{\text{next}} - S_{\text{curr}} + \begin{cases} 0.5 & \text{if } S_{\text{next}} > S_{\text{curr}}, \\ -0.5 & \text{if } S_{\text{next}} < S_{\text{curr}}, \\ 10 & \text{if } S_{\text{next}} = 1. \end{cases} \tag{4}$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \right) \tag{5}$$

where α is the learning rate, and γ is the discount factor. We set the learning rate as 0.1 and the discount factor as 0.95 for this specific dataset.

4 Experiment and Results

Given the MHP student mental health dataset, we utilize OpenAI's GPT models, GPT-4o, GPT-4o-mini, Gemini-1.5, and GPT-3.5, to demonstrate the performance of our proposed OnRL-RAG compared with existing methods, Standard RAG, and Simple LLMs. We define simple LLMs as using the response generation API of chatgpt without *knowledge base* from RAG. The MHP dataset collected responses from 2,028 college students and each student responded to 28 survey questions.

Our experiments were conducted using the MHP student mental health dataset [24], which contains 2,028 data points. We use 80% of the data to build up a basic model, then use each data point in 20% of the data as a new input, which would be used in the RL to improve the model performance. We list the results of similarity scores for the experiment in Table 2 and Figure 2.

From Table 2 and Figure 2, we observe that the OnRL-RAG framework consistently outperforms both the standard RAG and simple LLMs across the 3 GPT and 1 Gemini models in terms of similarity scores. For instance, in the GPT-4o model, the OnRL-RAG framework achieved a similarity score of 0.7901, higher than the standard RAG (0.7800) and much higher than the simple LLM (0.3837). Similar trends are observed in the other models as well: OnRL-RAG outperforms standard RAG and simple LLMs in GPT-4o-mini (0.7868 vs. 0.7434 and 0.3837), Gemini-1.5 (0.7320 vs. 0.7290 and 0.2041), and GPT-3.5 (0.7145 vs. 0.6455 and 0.3806). The higher similarity scores of OnRL-RAG indicate that it generates more accurate responses (closer to the ground truth), relevant (directly addressing the user's query), and contextually aligned (considering the specific context of the user's situation), making it a more effective approach for generating personalized responses compared to standard RAG and simple LLMs.

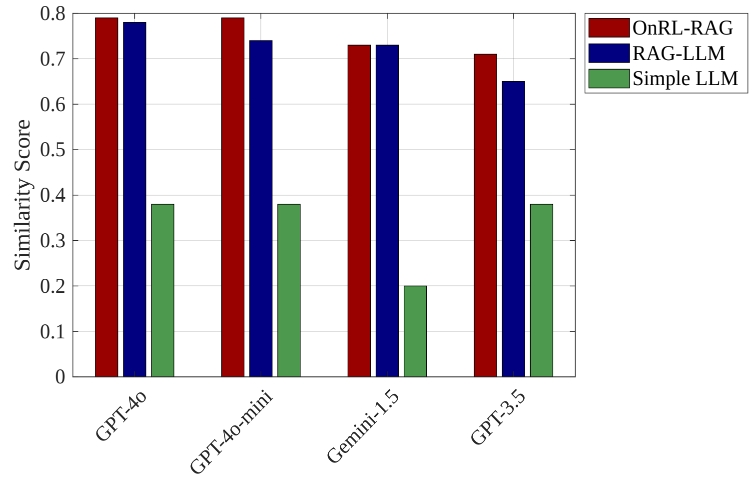

Fig. 2: Similarity Scores: Ground Truth vs. Model Responses

Table 2: Similarity scores between ground truth and generated responses across models

Method	**GPT-4o**	**GPT-4o-mini**	**Gemini-1.5**	**GPT-3.5**
OnRL-RAG	0.79 (+**0.01**)	0.79 (+**0.04**)	0.73 (+**0.00**)	0.71 (+**0.07**)
standard RAG	0.78	0.74	0.73	0.65
simple LLM	0.38	0.38	0.20	0.38

5 Discussions

In this work, we propose a novel OnRL-RAG for a real-life dataset collected from college students' survey questionnaires regarding their mental health stats. Our proposed method offers personalization for different population groups and achieves superior performance compared with the existing method. In this work, we only implement one dataset collected from 2028 college students and each one answered 28 questions. It would be better if the proposed method could be further demonstrated on other datasets in different fields. However, given the performance gaps between online and offline reinforcement learning, this dataset demonstration should prove the superior performance of the proposed OnRL-RAG. The results would open the possibility of incorporating LLMs into data science for diverse fields ranging from psychology to education.

References

1. Ahmad, S.R.: Enhancing multilingual information retrieval in mixed human resources environments: A rag model implementation for multicultural enterprise. arXiv preprint arXiv:2401.01511 (2024)

2. Ali, A.M., Alkhamees, A.A., Hori, H., Kim, Y., Kunugi, H.: The depression anxiety stress scale 21: development and validation of the depression anxiety stress scale 8-item in psychiatric patients and the general public for easier mental health measurement in a post covid-19 world. International journal of environmental research and public health **18**(19), 10142 (2021)
3. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-rag: Learning to retrieve, generate, and critique through self-reflection. arXiv preprint arXiv:2310.11511 (2023)
4. Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., Abdelrazek, M.: Seven failure points when engineering a retrieval augmented generation system. In: Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI. pp. 194–199 (2024)
5. Bucur, M.: Exploring Large Language Models and Retrieval Augmented Generation for Automated Form Filling. B.S. thesis, University of Twente (2023)
6. Chen, Z., Wang, X., Jiang, Y., Xie, P., Huang, F., Tu, K.: Improving retrieval augmented open-domain question-answering with vectorized contexts. arXiv preprint arXiv:2404.02022 (2024)
7. Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.S., Li, Q.: A survey on rag meeting llms: Towards retrieval-augmented large language models. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 6491–6501 (2024)
8. Finch, S.E., Paek, E.S., Choi, J.D.: Leveraging large language models for automated dialogue analysis. arXiv preprint arXiv:2309.06490 (2023)
9. Foster, D.J., Golowich, N., Qian, J., Rakhlin, A., Sekhari, A.: Model-free reinforcement learning with the decision-estimation coefficient. Advances in Neural Information Processing Systems **36** (2024)
10. Frederick, A., Daniels, H.A., Grineski, S.E., Collins, T.W.: 'i've never felt like that inhibits anything': the gendered frameworks of hispanic women college students in a stem program. Gender and Education **32**(5), 646–663 (2020)
11. Ji, T., Graham, Y., Jones, G.J., Lyu, C., Liu, Q.: Achieving reliable human assessment of open-domain dialogue systems. arXiv preprint arXiv:2203.05899 (2022)
12. Khatry, A., Bajpai, Y., Gupta, P., Gulwani, S., Tiwari, A.: Augmented embeddings for custom retrievals. arXiv preprint arXiv:2310.05380 (2023)
13. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
14. Long, Y., Luo, H., Zhang, Y.: Evaluating large language models in analysing classroom dialogue. npj Science of Learning **9**(1), 60 (2024)
15. Lu, S., Duan, N., Han, H., Guo, D., Hwang, S.w., Svyatkovskiy, A.: Reacc: A retrieval-augmented code completion framework. arXiv preprint arXiv:2203.07722 (2022)
16. Raja, M., Yuvaraajan, E., et al.: A rag-based medical assistant especially for infectious diseases. In: 2024 International Conference on Inventive Computation Technologies (ICICT). pp. 1128–1133. IEEE (2024)
17. Reimers, N.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
18. Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., Chen, W.: Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. arXiv preprint arXiv:2305.15294 (2023)

19. Shi, L., Zhang, K., Rong, W.: Query-response interactions by multi-tasks in semantic search for chatbot candidate retrieval. arXiv preprint arXiv:2208.11018 (2022)
20. Shi, X., Liu, J., Liu, Y., Cheng, Q., Lu, W.: Know where to go: Make llm a relevant, responsible, and trustworthy searchers. Decision Support Systems **188**, 114354 (2025)
21. Stade, E.C., Stirman, S.W., Ungar, L.H., Boland, C.L., Schwartz, H.A., Yaden, D.B., Sedoc, J., DeRubeis, R.J., Willer, R., Eichstaedt, J.C.: Large language models could change the future of behavioral healthcare: a proposal for responsible development and evaluation. NPJ Mental Health Research **3**(1), 12 (2024)
22. Su, C., Wen, J., Kang, J., Wang, Y., Pan, H., Hossain, M.S.: Hybrid rag-empowered multi-modal llm for secure healthcare data management: A diffusion-based contract theory approach. arXiv preprint arXiv:2407.00978 (2024)
23. Sun, G.H., Hoelscher, S.H.: The chatgpt storm and what faculty can do. Nurse Educator **48**(3), 119–124 (2023)
24. Syeed, M., Rahman, A., Akter, L., Fatema, K., Khan, R.H., Karim, M.R., Hossain, M.S., Uddin, M.F.: MHP (Anxiety, Stress, Depression) Dataset of University Students (5 2024). https://doi.org/10.6084/m9.figshare.25771164.v1
25. Tang, Y., Guo, D.Z., Zheng, Z., Calandriello, D., Cao, Y., Tarassov, E., Munos, R., Pires, B.Á., Valko, M., Cheng, Y., et al.: Understanding the performance gap between online and offline alignment algorithms. arXiv preprint arXiv:2405.08448 (2024)
26. Thüs, D., Malone, S., Brünken, R.: Exploring generative ai in higher education: a rag system to enhance student engagement with scientific literature. Frontiers in Psychology **15**, 1474892 (2024)
27. Wang, H., Huang, W., Deng, Y., Wang, R., Wang, Z., Wang, Y., Mi, F., Pan, J.Z., Wong, K.F.: Unims-rag: A unified multi-source retrieval-augmented generation for personalized dialogue systems. arXiv preprint arXiv:2401.13256 (2024)
28. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. vol. 33, pp. 5776–5788 (2020)
29. Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., Li, M., Lin, J.: End-to-end open-domain question answering with bertserini (2019)
30. Zhang, B., Yang, H., Zhou, T., Ali Babar, M., Liu, X.Y.: Enhancing financial sentiment analysis via retrieval augmented large language models. In: Proceedings of the fourth ACM international conference on AI in finance. pp. 349–356 (2023)