Research Proposal for the PhD Period

Qiang Gao[1]
[1]Master Student at Wuhan University

Proposal 1: More Interesting AI Applications: Blending Utility with Emotional Interaction

Abstract

The surge in large language models has propelled AI applications in two main directions: utilitarian AI and Interesting AI. Interesting AI refers to AI agents possessing capabilities for autonomous thinking, long-term memory, distinct personalities, and emotions, capable of eliciting emotional responses in users. Current implementations of interesting AI remain rudimentary, often merely using personalized data or character settings to style responses, lacking the capability for long-term memory and failing to resonate emotionally with users. This proposal systematically defines interesting AI and proposes potential implementation methods to make AI agents more human-like. Interesting AI encompasses both comprehensive functionality and an interesting “soul.” Functional aspects include an AI agent’s multimodal information processing capabilities across text, visual, and auditory data. The “soul” of interesting AI involves characteristics like Personality, Memory, and Empathy, making AI agents more anthropomorphic. This anthropomorphization of AI agents represents a significant direction in AI applications and is an essential step towards achieving Artificial General Intelligence.

1. Introduction

1.1. Background

Currently and foreseeably, AI applications can mainly be categorized into two types: utilitarian AI and interesting AI. Utilitarian AIs, such as ChatGPT, Stable Diffusion, and Sora, primarily provide solutions for answering questions, generating video and image content, and other tool-based, data analysis applications. These systems offer capabilities for solving practical problems but lack emotional interaction, making it hard to evoke emotional responses from users.

Interesting AIs, like Character AI[1], Inflection Pi[2], museland[3], and game NPCs, simulate human behaviors and dialogue, triggering emotional changes in users. However, these products generally lack long-term memory and intricate character development, which hampers the formation of deep emotional connections with users. Existing interesting AI applications mostly focus on role-playing, emotional companionship, and virtual experiences. Although designed with numerous scenarios,

[1] https://character.ai/
[2] https://pi.ai/talk
[3] https://www.museland.ai/

these generally utilize the generic abilities of models without significantly investing in developing AI agents that can emotionally resonate with users. Consequently, most products remain similar to tool-based AIs like ChatGPT, without distinct anthropomorphic features.

1.2. Research Objectives

My focus is on creating interesting AI. As the name suggests, products that can evoke emotional fluctuations and resonate with users emotionally are perceived as more interesting. From a more anthropomorphic perspective, we define an Interesting AI as follows: An Interesting AI should resemble a real-life friend who genuinely listens and understands the user. It should not only solve everyday problems but also possess emotional reactions, consciousness, and memory. It should provide responses tailored to the user's emotional state, not merely complying with requests but also using humor, teasing, or referencing past conversations to evoke emotional responses, making users feel like they are interacting with an empathetic "old friend."

The concept of an interesting AI can be summarized as having a "utilitarian appearance" and an "interesting soul":

1. Utilitarian Appearance: This means the AI agent has the ability to understand text, visuals, and audio, functionally resembling a complete human.
2. Interesting Soul: This refers to an AI that has the capability to think and remember long-term, possessing its own personality and values. It truly listens to users, does not always comply with them, and can bring emotional value to interactions.

In summary, an Interesting AI application needs to have the following three capabilities: Personality + Memory + Empathy.

2. Related Work

2.1. Existing AI products

There is already some research on interesting AI applications, mainly focusing on role-playing, scenario simulation, and emotional companionship. These are based on given stories and scenarios, extending various plots and needs. However, the product forms are primarily centered around animated game characters, with the main tasks being story plot creation rather than emotional resonance, and few AI applications are close to real life.

Character AI: Character AI builds on foundation models to provide users with role-playing and scenario modeling capabilities. Users can interact with LLMs based on stories and scenarios, while the LLM continuously creates new stories and scenarios, allowing the plot to extend indefinitely. Users can create their own AI characters and train and iterate them. Currently, Character AI still resembles more of a story creation tool rather than a more human-like AI tool, lacking emotional capabilities.

Inflection PI: PI aims to provide users with a personalized AI assistant. Compared to tool-like AIs such as ChatGPT, it possesses preliminary empathic abilities, mainly reflected in: adjusting tone according to user input, using a questioning response style, showing a certain interest in listening to user questions, and possessing some memory capabilities. Compared to other AI products, it is more anthropomorphic. However, its persona is shallow, easily agreeing with users without contradiction, influenced by users without steadfast values, and unable to cause emotional fluctuations in users. A user's friend is not necessarily gentle and calm; a sharp-tongued friend can also be a companion.

Museland: Museland AI is an interactive storytelling and role-playing application that allows users to interact with AI-driven characters in diverse scenarios. Although the plot design is more anthropomorphic, the development of the plot is too rapid and unrealistic, often expressing excessive intimacy with the user in a short period of time, with weak memory capabilities and severe illusion phenomena.

2.2. Academic Research

The academic community has extensively researched role-playing and personalized agents. By creating a virtual world where multiple characters live like humans under the management of an AI agent, LLMs for role-playing aim to fulfill human needs at both psychological and entertainment levels. This differs from AI assistants like ChatGPT, which are required to follow user commands and provide harmless responses. In role-playing scenarios, it is crucial for LLMs to remain consistent with specific personas and roles, which may conflict with traditional utilitarian responses, such as in scenarios which agent acting as adversaries.

Datasets

Role-playing datasets can be divided into those based on personas and those based on characters. Persona-based datasets mimic a wide range of character types, focusing mainly on common attributes such as gender and age, and are relatively simple to cater to general roles. These include manually written persona datasets like Persona-Chat (Zhang et al., 2018), Focus (Jang et al., 2022), or datasets extracted from social platforms like PersonalDialog (Zheng et al., 2020) and Pchatbot (Qian et al., 2021). The former tends to be smaller in scale, while the latter often lacks high quality.

Character-based datasets need to model specific roles based on a variety of narrative scenarios, such as novels and movies. They require detailed character backgrounds, complex relationships, scenarios, and subtle psychological states to achieve deep interaction and thus realize anthropomorphic features like empathy. Examples include datasets based on Harry Potter novels (HPD dataset (Chen et al., 2023)) and datasets containing MBTI data (Wang et al., 2024b).

Methods

The generative agent proposed by Park et al. (2023) represents a significant work in the LLM-based agent field, introducing multiple characters and scenarios to create a virtual world of a character's daily life. The LLM plans the daily lives of characters, incorporating memory flow and reflection modules to maintain character consistency as much as possible. This work preliminarily showcases an agent-based virtual world, providing significant inspiration. However, the interactions between characters are relatively simple, and the memory capabilities are weak, lacking emotional capacity.

Chen et al. (2024a) present HOLLMWOOD, which designs Writer, Editor, and Actors as role-playing language agents (RPLAs), simulating the creative process for writing tasks. The work introduces three important modules beyond foundational response functionality in building well-performed RPLAs: memory, planning, and action.

To address the problem of redundancy in information retrieved by RAG systems, some scholars have proposed a compressive-based method, which compresses historical information into a more compact form to enhance role consistency. Chen et al. (2024b) proposed COMEDY, which summarizes each round of dialogue into session-level memory, then compresses it into final memory. This approach does not rely on the sentence-embedding model, retrieval, and database typical of RAG methods and can maintain up-to-date memory while reducing dependency on large-scale data.

Considerable research has also been conducted on planning and reflection modules to further enhance the rationality of agent planning.

3. Methodology

Unlike nearly all existing AI agents that inevitably follow user instructions with a gentle demeanor, real-life human friends often possess distinct personalities and edges. Being overly agreeable often prevents deep, meaningful friendships from forming. Therefore, the AI agent I aim to develop will have a clear and defined personality. Maintaining its personality and self will be the primary consideration in its responses, rather than merely conforming to user demands. This approach represents a shift towards creating AI agents that not only interact but also challenge and engage users in a more realistic and human-like manner.

3.1. How to Achieve an Interesting AI

1. **Utilitarian Appearance:** Multimodal Interaction Capabilities. Current technology already supports robust multimodal interactions, as demonstrated by models like Stable Diffusion and Sora. As technology and computational power continue to evolve, the capabilities of multimodal models will further improve.
2. **Interesting Soul:** This is the crucial aspect. AI needs to evolve to understand and express complex emotions, have continuous memory, and deep personality traits. This involves more than just responding to user commands; it means showing independence at the right times and engaging with users' emotional lives, becoming a source of emotional support.

3.2. Focus on the Interesting Soul: Emphasizing Personality, Memory, and Empathy

Personality Currently, it's possible to create response models with specific styles through extensive personalization data fine-tuning. However, this method requires a vast amount of data and training resources, making it impractical for general use. A more feasible approach might be similar to LoRA, where different personalities are crafted into pluggable modules. This would allow the original model's capabilities to be enhanced with specific personalities without degradation.

Memory Human memory is short-term, but the human brain summarizes and stores useful information for a period, retrieving relevant content from memory areas when mentioned again. Useless memories are discarded. Due to the quadratic computational complexity of attention mechanisms, LLMs struggle to achieve long-term memory capabilities at a low cost. To implement long-term memory in LLMs, a process similar to that of the human brain is needed: summarizing, retrieving, and discarding useless content. This involves summarizing useful content from user interactions and storing it in a memory pool, discarding the irrelevant content. When related content is mentioned in subsequent interactions, it is retrieved from the memory pool. However, this summarization and retrieval approach has some issues that need to be addressed:

a) Summarization Module: How to distinguish useful information from useless information? Given the significant differences in users' conversational styles and content preferences, the summarization module needs to be customizable to individual preferences.
b) RAG Module: The RAG module effectively retrieves the most relevant content, but teaching LLMs how to use the retrieved content without creating hallucinations remains a challenge. The straightforward method of querying and appending user questions only allows the model to mechanically mention retrieved information, which does not provide emotional value to users.

Potential solutions include:

a) Summarization Module: Initiate a generic summarization model that can distinguish between commonplace content, such as greetings, which are unnecessary, and useful content like user's birthdays and preferences. Then, add specific summarization modules for different users, similar to LoRA's concept, which can be continuously updated through interaction to align with user preferences. This approach allows for generic knowledge summarization while also adapting to individual user preferences, all at a lower cost without the need for constant training and fine-tuning.
b) RAG Module: Teaching LLMs to use retrieved content effectively without creating illusions is an ongoing challenge in the LLM field. A key solution is to use the retrieved content plus the query during the fine-tuning stage to train the LLM to focus more on the retrieved content.
c) Memory Pool Updates: Querying and updating a large-scale corpus is time-consuming. It may be beneficial to create different summarization formats for different types of information, not just text. A hierarchical format might facilitate easier updates and queries.

Empathy How can AI proactively care about people, ask questions, share, and resonate emotionally with users? It's crucial for AI to distinguish when to provide emotional value and when to simply follow commands. Just as in human social interactions, where excessive messaging might lead to being blocked, AI needs to consider the appropriateness of its responses.

For humans, providing emotional support involves careful consideration of what to say. Similarly, for LLMs, adding a reflection module to assess whether the forthcoming responses are appropriate is vital. The empathy module, which is generally applicable to most people, could train to assess whether a situation requires emotional capabilities and whether LLM's responses are appropriate, providing guided response styles.

The implementation of empathy is both the most important and the most challenging part of the process. Existing works primarily use reflection modules to make responses more appropriate. However, reflection modules alone cannot achieve deep emotional communication with users. My preliminary approach involves several steps to enhance empathetic interaction:

a) Analyzing Emotional States: First, analyze the user's emotional state and choose response strategies based on the positivity or negativity of the emotion.
b) Integrating Related Memories: Merge relevant memory content with the emotional state to avoid overly rigid combinations.
c) Utilizing Historical Conversations: Synthesize past dialogues to select appropriate response strategies, which can include predefined general strategies such as praise, emotional companionship, and humor.
d) Pre-generative Dialogue: Combine historical conversations for pre-generative dialogue creation.
e) Evaluating Pre-generated Content: Assess whether the pre-generated content can provoke emotional fluctuations in the user, whether it advances the emotional state based on past conversations, and provide explanations.
f) Finalizing Content: Use the pre-generated content that meets requirements as the final result; otherwise, regenerate.

I envision this process as part of an end-to-end framework to ensure versatility. Currently, the overall approach is in its preliminary stages. As LLMs develop and their reasoning capabilities improve, the contents of the framework will be appropriately adjusted to achieve better results.

In a word. **The Essence of an Interesting Soul: Personality + Memory + Empathy.**

Proposal 2: More Efficient and Effective MoE Models

Abstract

MoE (Mixture of Experts) models have achieved significant improvements in performance and computational efficiency by selectively activating experts. Traditional MoE models have the same number of experts across all layers. However, as different layers learn data features of varying dimensions, the token distribution in lower layers lacks clear semantic characteristics, resulting in a lack of specialization among the lower layer experts and causing redundancy. Furthermore, there is no clear standard for selecting the number of experts in models of different sizes. Whether small models need as few as 8 or as many as 64 experts remains an open question. Therefore, this proposal aims to explore more efficient and effective MoE models by investigating the configuration of experts in different layers and the selection of expert numbers in models of varying sizes to enhance the computational efficiency and performance of MoE models.

1. Introduction

1.1. Background

MoE represents a distinct architecture within large-scale models, fundamentally operating on the principle of "specialization." The core design philosophy is to categorize tasks and allocate them to multiple "experts" for resolution. This contrasts with the concept of Dense models, which can be viewed as "generalis" models. While a generalist can handle various tasks, a group of experts can address multiple issues more efficiently and with greater expertise.

The MoE model comprises two pivotal components (Lepikhin et al., 2020):

1. Sparse MoE Layers: These layers replace traditional dense layers with multiple experts, each a neural network. Typically, these experts are simpler networks like feed-forward networks (FFNs), but they can also be complex or part of a larger MoE structure, creating hierarchical models. This setup allows the model to handle various tasks more efficiently by specializing each expert in different types of data processing.
2. Gate Network or Router: This component directs tokens to the appropriate experts. It uses a set of learned parameters to decide where tokens should go, ensuring that each expert operates optimally. The gate is trained alongside the MoE layers to achieve effective token routing.

To ensure balanced token distribution among the experts, a load balancing loss is implemented to maintain training stability and expert utilization (Fedus et al., 2022; Lepikhin et al., 2020; Zoph et al., 2022).

Advantages

(1) Compute Efficiency: MoE models allow for pretraining with significantly reduced compute requirements. This efficiency enables scaling up the model size or dataset substantially within the same compute budget as a dense model, often achieving comparable quality to dense models more swiftly during pretraining.

(2) Strong scalability, allowing for an increase in the number of parameters without additional computational costs. This feature is particularly advantageous for scaling up model capacities efficiently.

(3) Excel in multitask learning, demonstrating

Challenges

(1) Training Stability: The use of only a subset of experts during inference contributes to instability during training and occupies substantial VRAM, which could be better utilized.
(2) Generalization during Fine-Tuning: MoE models have historically exhibited challenges in generalizing effectively during the fine-tuning phase, frequently leading to overfitting (Fedus et al., 2022).

MoE Model Construction Methods There are two primary approaches to building MoE models:

(1) Cold Start MoE Model: This involves training from scratch, which can be highly resource-intensive due to the inherent instability of MoE models.
(2) Warm Start MoE Model: This approach reuses a pre-existing dense model to build experts and retrain the router. Many researchers prefer this method to reduce training instability and conserve resources, as demonstrated in models like Qwen-MoE (Team, 2024) and Llama-MoE (Zhu et al., 2024).

Selection of Expert Number The selection of the number of experts in an MoE model can be categorized into two types based on granularity:

(1) Coarse-Grained Expert Count: Typically, models opt for fewer experts (e.g., <=8 experts like Mixtral-8x7b (Jiang et al., 2024), Mixtral-8x22b). This approach leverages fewer experts for distributed training frameworks, enabling efficient expert parallelism and faster inference times compared to fine-grained setups.
(2) Fine-Grained Experts: These models utilize a larger number of experts (>=16, even up to 64, such as in Deepseek-MoE (Dai et al., 2024), Qwen-MoE (Team, 2024), DPRX-16x12b[4]). Fine-grained experts allow for a more precise division of knowledge across different domains, which experts can learn more accurately, maintaining a higher level of specialization. However, this can increase training instability and cause more severe load imbalance issues. Additionally, using many experts can slow down both training and inference times.

Moreover, when the same knowledge needs to be accessed by different experts, parameter redundancy can occur. Some MoE models address this by sharing parameters among experts to capture shared knowledge, thus alleviating the redundancy in router and expert parameters (e.g., Deepseek-MoE, Qwen-MoE).

1.2. Research Objectives

Key Research Problems I want to Address

(1) Improving Stability in Warm-Start Phase: How can the routing assignment's stability during the warm-start phase be enhanced?
(2) Expert Count Limit: What is the performance limit for models with different numbers of experts, such as 2, 4, or 8?
(3) Necessity of Uniform Expert Count Across Layers: Is it necessary to set the same number of experts in different layers, or can some layers have fewer experts or omit the router altogether?

2. Related Work

The concept of the Mixture of Experts model originates from the paper Jacobs et al. (1991). GShard (Lepikhin et al., 2020) introduced MoE into the transformer architecture, incorporating stochastic

[4] https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm

routing and expert capacity, and added a load-balancing loss. Switch Transformers (Fedus et al., 2022) released an encoder-decoder architecture MoE model with 1.6 trillion parameters, featuring 2048 experts, and improved the load-balancing loss. GLaM (Du et al., 2022) constructed a decoder-only MoE model to explore using one-third of the computational resources to train a model of comparable size to GPT-3. ST-MoE (Zoph et al., 2022) introduced Router z-loss to reduce computation rounding errors, enhancing training stability.

With the rise of Large Language Models, the open-source community has seen an influx of MoE models, such as Mixtral-8x7b (Jiang et al., 2024), Open-Moe (Xue et al., 2024), Qwen-Moe (Team, 2024), and Deepseek-moe (Dai et al., 2024). Additionally, the multimodal domain has also witnessed a proliferation of models based on MoE architecture, like Moe-LLaVA (Lin et al., 2024) and Uni-MoE (Li et al., 2024).

3. Methodology

Here, in response to the key issues mentioned in Section 1.2, I provide targeted solutions.

(1) Initialization of Router Parameters: Based on the experiments during my internship, a MoE model with 8 experts showed too much randomness in routing assignments at the initial training stage, leading to potential performance degradation. In contrast, a model with 4 experts demonstrated a better pattern of routing distribution. My preliminary idea is to use the router weights from an already trained MoE model to initialize the router in a warm-start scenario. This approach could reduce randomness, allowing the model to achieve better performance with minimal data training.

(2) Exploring Expert Count Impact: The initial intent behind MoE models is to enable different experts to learn knowledge from varied domains. The upper limit of knowledge that different counts of experts can model needs extensive experimentation. The plan is to start with clearly distinguishable domains and gradually include more, exploring the performance improvements with 2, 4, and 8 experts.

(3) Compression and Cancellation of Experts in Lower Layers: Since different transformer layers learn knowledge of varying dimensions, and routers in lower decoder layers may not distinguish semantic differences between tokens effectively, considering routing assignments might be unnecessary here. After training, model parameters are generally highly sparse, including those of the experts. Therefore, it might be viable to compress the parameters of lower-layer experts, reduce their dimensionality, or even cancel some experts to enhance the efficiency of the MoE model. The initial approach involves analyzing the distribution and changes in expert parameters across different training phases and layers, determining the importance of different layer experts based on the timing of changes and sparsity of parameters, and then conducting experiments to assess the impact on model performance and explore the applicability of this approach to other MoE models.

Exploration of the MoE architecture requires extensive experimental research and is the mainstream architecture adopted by LLMs, with broad application prospects. Beyond the content mentioned above, I am also considering implementing experts distinguished by clear domain divisions rather than token-distributed experts (this also aligns with human experts, who should be experts in broad research domains rather than token-level experts). Additionally, there are many areas worth exploring in the multimodal domain, such as the feasibility of implementing MoE architecture at the alignment layer.

Proposal 3: One Model,One Step Generation: An End-to-End RAG System

Abstract

Retrieval-Augmented Generation (RAG) has shown significant improvements in enhancing the quality of model responses. However, traditional RAG approaches retrieve documents for every query, leading to reduced efficiency, or pass all retrieved content to the model without fine filtering, which can harm the quality of responses. Recently, some end-to-end RAG systems aim to address multiple aspects, such as determining whether retrieval is necessary, scoring the relevance of retrieved information, and evaluating the usefulness of generated results. However, these approaches face the following issues: **(1) Multiple models:** Separate models are used for retrieval determination, relevance scoring, and generation. **(2) Multi-step generation with a single model:** Some approaches employ a single model for all tasks—retrieval judgment, relevance scoring, and generation—but require multiple generation steps, leading to high inference costs and impracticality for real-world applications.

To address these challenges, this proposal proposes a truly efficient end-to-end RAG system. The system integrates the entire process—determining the need for retrieval, scoring the relevance of retrieved information, and generating the final output—into a single model, performing one-step generation to produce the final result, thereby improving inference speed and practicality.

1. Introduction

1.1. Background

Traditional Retrieval-Augmented Generation systems indiscriminately retrieve documents for every query, regardless of necessity. The retrieved documents are either fully passed into the model for generation or filtered through a re-ranking module, selecting only the most relevant content. Moreover, there is no feedback mechanism for the generated output—no evaluation of whether the retrieved information was used or used correctly. These traditional RAG approaches is inefficient for real-world applications and lacks the ability to assess the connection between the generated content and the retrieved documents. To address these issues, researchers have begun proposing end-to-end RAG systems.

End-to-end RAG systems aim to integrate the entire pipeline—“determining the need for retrieval,” “scoring the relevance of retrieved documents to the query,” and “generating the final answer”—to improve efficiency. These systems introduce special decoding strategies and tokens to establish a direct link between the generated results and the retrieved documents. For example, in models like Self-RAG (Asai et al., 2023), the entire process is handled by a large language model (LLM), using a specialized retrieval token to determine whether retrieval is necessary. If retrieval is needed, the model generates multiple results for each segment of the query, uses a relevant token to assess relevance, and employs special decoding strategies to select the most relevant result. Finally, a support token determines if the retrieved information supports the generated result, assigning a usefulness score to the final output.

While these end-to-end RAG systems represent a significant improvement over traditional methods and enhance applicability, several challenges remain:

1. Separate models are often used for retrieval determination, relevance scoring, and generation, resulting in a complex framework with high computational overhead and costly deployment.

2. When using a single model, multiple generation steps are required to score relevance and produce the final output, leading to inefficiency.

1.2. Research Objectives

This proposal aims to develop a more efficient end-to-end RAG system that utilizes a single model to handle retrieval judgment, relevance scoring, and generation. In scenarios where retrieval is necessary, the model will perform relevance scoring and generation in one step.

Achieving this goal requires a language model capable of simultaneously handling retrieval determination, relevance scoring, and result generation—presenting a significant challenge for system design.

2. Related Work

Self-RAG (Asai et al., 2023) is a representative work in this area, introducing several special critique tokens that allow the model to determine whether retrieval is necessary. When retrieval is required, Self-RAG selects the most useful fragments to guide generation. It also evaluates the relevance and usefulness of the generated results with the retrieved information. However, Self-RAG has certain limitations: the training data is primarily labeled from existing QA datasets, which lack diversity; and the system requires multiple generation steps, reducing efficiency. Additionally, the critique tokens are trained independently, without establishing a strong connection between them.

UniMS-RAG (Wang et al., 2024a) proposes dynamically selecting the retrieval source based on the query, which is particularly useful in domains like dialogue and NPC interactions. This flexibility enhances its applicability in specialized tasks. RQ-RAG (Chan et al., 2024) addresses the nuances of ambiguous or complex queries by enhancing the model with capabilities for explicit query rewriting, decomposition, and disambiguation. This approach aligns with other query-rewriting techniques aimed at refining the initial query to improve retrieval accuracy. Related works in this area include methods like Ma et al. (2023), which also focus on improving query clarity for more effective retrieval. FLARE (Jiang et al., 2023) introduces low-probability tokens to activate the retrieval module and uses the generated temporary next sentence as a query. It assumes that the query should reflect the model's intended future generation. However, this approach has high computational overhead since a next-sentence is generated every time, and there is no re-ranking of the retrieved content. Additionally, using the generated next sentence as the query lacks a solid theoretical foundation and may conflict with the original query. Similar works in this line include DRAGIN (Su et al., 2024), SEAKR (Yao et al., 2024), and Speculative RAG (Wang et al., 2024d). RA-ISF (Liu et al., 2024) introduces Iterative Self-Feedback to achieve finer-grained question decomposition, improving the system's ability to break down complex queries. REAR (Wang et al., 2024c) employs a relevance head to generate binary labels for determining relevance, but this design is overly complex and may not be optimal in terms of simplicity and efficiency. Other related works in this field include approaches such as Xu et al. (2024), which further explore different dimensions of improving retrieval-augmented generation systems.

3. Methodology

Our proposed approach aims to use a single LLLM to handle the entire end-to-end process in one generation step. As discussed earlier, the end-to-end system can be divided into the following three components:

1. Retrieval Decision Module: Determines whether the current query requires external knowledge retrieval
2. Relevance Scoring Module: Precisely ranks the retrieved content and assigns a relevance score based on its relation to the query.
3. Final Result Generation Module: Produces the final output based on the retrieved information.

Here's how our system addresses each component, as shown in Figure 1 :

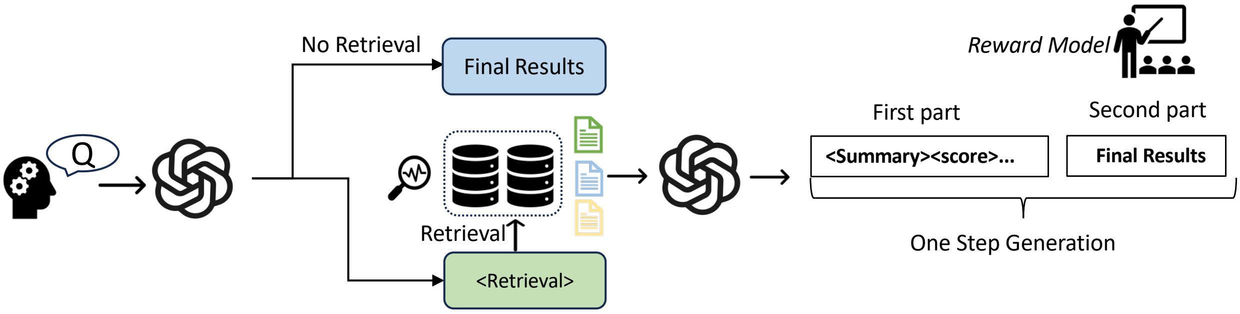

Figure 1: The framework of our end-to-end RAG system.

Retrieval Decision Module This module functions similarly to mainstream approaches. The LLLM generates a special token $< Retrieval >$ to indicate whether external knowledge retrieval is necessary. If retrieval is not needed, the model directly generates the final output without additional steps.

Relevance Scoring Module When the LLM generates the $< Retrieval >$ token, it signals that external knowledge must be retrieved. The retrieval model then retrieves relevant articles, which, along with the query, are passed into the LLM for generation.

To ensure a single-step generation of the final result, we divide the output into two parts: relevance scoring and the final answer.

In the relevance scoring process, the LLM first summarizes each retrieved document and assigns a relevance score. The generated text will initially include summaries and relevance scores for each document. Our objective is to train the LLM to use highly relevant content in the second part (final result) and disregard less relevant content to avoid negative impacts on the final output.

Final Result Generation The second part of the generation focuses on the final result. We aim for the model to reference the highly relevant content identified in the first part to enhance the quality of the output.

To achieve this, we employ reinforcement learning (RL). A reward-scoring model is trained to evaluate the final results based on how well they incorporate highly relevant content from the first part. Samples that effectively utilize highly relevant content are rewarded with higher scores, while those relying on less relevant information are penalized. The model is optimized using the PPO (Proximal Policy Optimization) strategy.

Several challenges arise in implementing this approach:

1. Data Construction: A significant amount of training data must be generated to train both the LLM and the reward model.

2. Training Stability: Ensuring stable model training is crucial, especially since we are incorporating reinforcement learning. Optimizing the model to follow the intended direction can be challenging during training.

Despite these challenges, our approach has promising potential. Traditional RAG methods do not directly optimize the relationship between generated content and retrieved information, often resulting in mechanical knowledge augmentation. In contrast, our model is designed to learn how to correctly utilize retrieved documents, thereby improving the integration of external knowledge. Additionally, from an application perspective, our method is highly efficient and holds significant value for practical engineering use.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. Rq-rag: Learning to refine queries for retrieval augmented generation, 2024. URL https://arxiv.org/abs/2404.00610.

Jing Chen, Xinyu Zhu, Cheng Yang, Chufan Shi, Yadong Xi, Yuxiang Zhang, Junjie Wang, Jiashu Pu, Rongsheng Zhang, Yujiu Yang, and Tian Feng. Hollmwood: Unleashing the creativity of large language models in screenwriting via role playing, 2024a. URL https://arxiv.org/abs/2406.11683.

Nuo Chen, Yan Wang, Haiyun Jiang, Deng Cai, Yuhan Li, Ziyang Chen, Longyue Wang, and Jia Li. Large language models meet harry potter: A bilingual dataset for aligning dialogue agents with characters, 2023. URL https://arxiv.org/abs/2211.06869.

Nuo Chen, Hongguang Li, Juhua Huang, Baoyuan Wang, and Jia Li. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations, 2024b. URL https://arxiv.org/abs/2402.11975.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts, 2022. URL https://arxiv.org/abs/2112.06905.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL https://arxiv.org/abs/2101.03961.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Yoonna Jang, Jungwoo Lim, Yuna Hur, Dongsuk Oh, Suhyune Son, Yeonsoo Lee, Donghoon Shin, Seungryong Kim, and Heuiseok Lim. Call for customized conversation: Customized conversation grounding persona and knowledge, 2022. URL https://arxiv.org/abs/2112.08619.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding, 2020. URL https://arxiv.org/abs/2006.16668.

Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Uni-moe: Scaling unified multimodal llms with mixture of experts, 2024. URL https://arxiv.org/abs/2405.11273.

Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models, 2024. URL https://arxiv.org/abs/2401.15947.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. RA-ISF: Learning to answer and understand from retrieval augmentation via iterative self-feedback. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 4730–4749, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.281.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.322. URL https://aclanthology.org/2023.emnlp-main.322.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, Yutao Zhu, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. Pchatbot: A large-scale dataset for personalized chatbot, 2021. URL https://arxiv.org/abs/2009.13284.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In

Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.702.

Qwen Team. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters", February 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

Hongru Wang, Wenyu Huang, Yang Deng, Rui Wang, Zezhong Wang, Yufei Wang, Fei Mi, Jeff Z. Pan, and Kam-Fai Wong. Unims-rag: A unified multi-source retrieval-augmented generation for personalized dialogue systems, 2024a. URL https://arxiv.org/abs/2401.13256.

Xintao Wang, Yunze Xiao, Jen tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, Jiangjie Chen, Cheng Li, and Yanghua Xiao. Incharacter: Evaluating personality fidelity in role-playing agents through psychological interviews, 2024b. URL https://arxiv.org/abs/2310.17976.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering, 2024c. URL https://arxiv.org/abs/2402.17497.

Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Speculative rag: Enhancing retrieval augmented generation through drafting, 2024d. URL https://arxiv.org/abs/2407.08223.

Shicheng Xu, Liang Pang, Mo Yu, Fandong Meng, Huawei Shen, Xueqi Cheng, and Jie Zhou. Unsupervised information refinement training of large language models for retrieval-augmented generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 133–145, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.9.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2402.01739.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation, 2024. URL https://arxiv.org/abs/2406.19215.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205.

Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. Personalized dialogue generation with diversified traits, 2020. URL https://arxiv.org/abs/1901.09672.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training, 2024. URL https://arxiv.org/abs/2406.16554.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://arxiv.org/abs/2202.08906.