Knowledge Augmented Finetuning Matters in Both RAG and Agent Based Dialog Systems

Yuxuan Wu[1,⋆], Yucheng Cai[1,*], Yi Huang[2], Junlan Feng[2], and Zhijian Ou[1,†]

[1] SPMI Lab, EE Department, Tsinghua University, Beijing, China
{wyx23,cyc22}@mails.tsinghua.edu.com, ozj@tsinghua.edu.com
[2] China Mobile Research Institute, Beijing, China
{huangyi,fengjunlan}@chinamobile.com

Abstract. Large language models (LLMs) have recently been applied to dialog systems. Despite making progress, LLMs are prone to errors in knowledge-intensive scenarios. Recently, approaches based on retrieval augmented generation (RAG) and agent have emerged to improve the factual accuracy by enhancing the LLMs with knowledge retrieved from external knowledge bases (KBs). This is mostly implemented by prompting the LLMs with instructions, examples and the retrieved knowledge. However, LLMs may have difficulty using the retrieved knowledge effectively for response generation, because they are not well trained to do such generation for specific domains. To mitigate this problem, we propose to finetune the LLMs in the RAG-based and agent-based systems with domain-specific data, together with domain-specific external knowledge, which is called **k**nowledge **a**ugmented **f**ine**t**uning (KAFT). We base our study on the MobileCS2 dataset, a real-life customer service dialog dataset that features intensive knowledge interactions, to systematically compare the prompting and KAFT techniques in the RAG-based and agent-based systems. Experiment results show that KAFT substantially surpasses prompting in both RAG and agent systems, particularly in terms of factual accuracy. To the best of our knowledge, this paper represents the first solid empirical work to investigate the KAFT idea.

Keywords: Knowledge augmented finetuning · Prompting · Retrieval augmented generation· Agent · Dialog Systems

1 Introduction

Recent progress in large language models (LLMs), such as GPT4 and PALM [1, 8], has shown improved performance in a range of natural language processing (NLP) tasks. These improvements have stimulated researchers and practitioners to integrate LLMs into real-world applications such as dialog systems. For real-life dialog systems, it is crucial for LLMs to respond accurately and reliably, which usually require domain-specific knowledge. Despite their power, in

⋆ Equal contribution, †Corresponding author.
This work is supported by the National Science and Technology Major Project (2023ZD0121401).

knowledge-intensive dialog systems, LLMs often generate outputs that are inaccurate or misleading, a phenomenon known as "hallucination" [18]. This poses a significant challenge to the factual accuracy of the systems.

In order to mitigate the phenomenon of hallucination, several approaches have been proposed. Among them, the RAG approach stands out as a promising solution. By integrating knowledge retrieval into the generative system, RAG significantly enhances factual accuracy and reduces hallucination [21, 17, 6]. In addition, there are growing interests in the agent-based approach, which exploits the tool-calling capability of LLM [25, 27]. By employing API calls, the agent approach aims to improve factual accuracy within question-answering (QA and dialogue systems, as demonstrated by the study in [31]. The knowledge obtained by the API calls in the agent-based system is similar to the retrieved knowledge in the RAG-based system.

For both the RAG and agent based dialog systems, the commonly adopted implementation is to prompt the LLM to directly utilize the external knowledge obtained by the system as in Figure 1(b). Instructions and examples are added to the prompts, along with the retrieved knowledge, to improve the system performance [23]. However, even with the instructions and examples, the LLMs may still have difficulty in effectively using the retrieved knowledge, because they are not well trained to do such generation for specific domains [7, 11]. For example, in the case shown in Figure 1(b), the LLM does not understand the term "directional flow" which is specific to the mobile service domain. Therefore, the LLM cannot deduce that the reason for the user's overage flow is the use of the flow in other apps. To mitigate this problem, we propose to finetune the LLMs in the RAG and agent based systems with domain-specific data, together with the domain-specific external knowledge, which is called knowledge augmented finetuning (KAFT) in this work, as shown in Figure 1(c). Conventionally, to adapt LLMs to a specific domain, the LLMs can also been directly finetuned, without using the RAG or agent-based systems, as shown in Figure 1(a). However, the performance of this direct finetuning of LLMs is often even inferior to the un-finetuned RAG-based systems [13, 28]. *In this paper, we focus on the RAG and agent based systems, investigate the method of finetuning the LLMs with retrieved knowledge in those systems, and compare it to the method of prompting the LLMs in those systems.*

The idea of finetuning LLMs is not new, but prior works mostly study direct inference tasks with LLMs (such as close-book QA, text completion, machine translation, and so on). As far as we know, there is no solid work to investigate the KAFT idea in the knowledge-intensive tasks where RAG or agent based methods are built to retrieve knowledges from external KBs. This paper represents the first solid empirical work to investigate the KAFT idea. Futuremore, unlike other fine-tuning techniques such as the Lora technique [15], which aims to improve the effectiveness and efficiency of fine-tuning, the proposed KAFT method aims to teach LLMs a new skill, i.e., the ability to make use of domain-specific external knowledge. KAFT teaches LLMs to take advantage of retrieved knowledge by constructing corresponding training data to finetune the model,

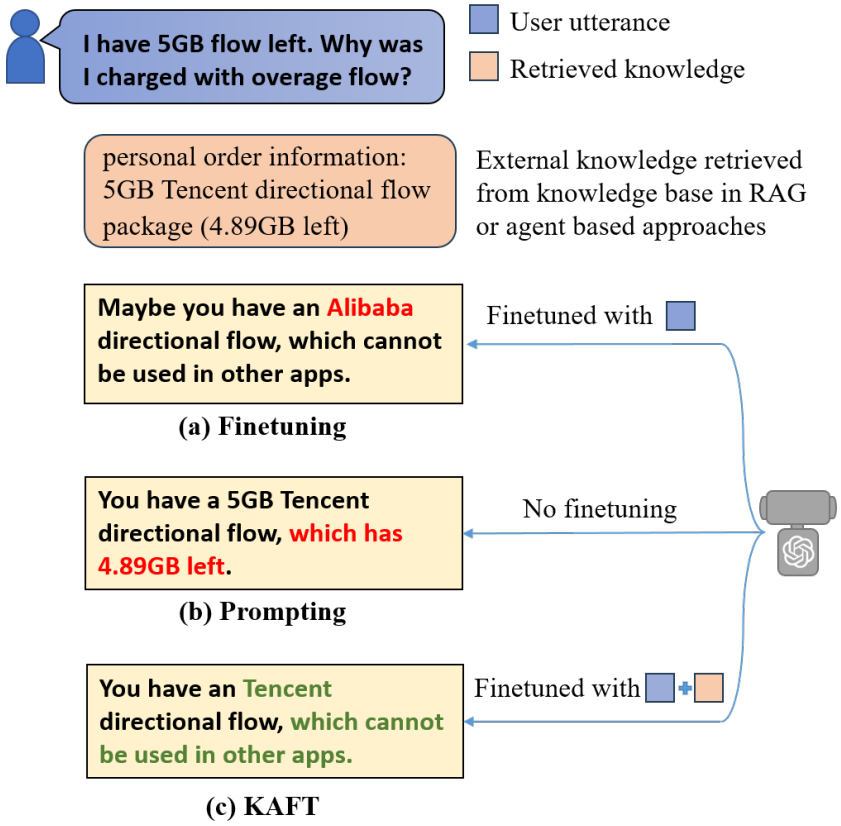

Fig. 1: Overview of the three methods to improve factual accuracy for the LLM based dialog systems. Direct finetuning without knowledge still often leads to serious hallucination, as the LLM may be unaware of the information gap during the training and testing situations. Meanwhile, for the prompting method, the LLM is not trained on domain specific data, which cannot fully leverage the domain-specific external knowledge (such as the "directional flow" in the example). The proposed KAFT method finetunes the LLM on domain specific data with external knowledge, which overcomes the drawback of the direct finetuning method and the prompting method.

similar to the instruction-tuning technique [9] which teaches LLMs the ability to follow instructions.

To demonstrate the efficacy of the proposed KAFT method, we build both RAG-based and agent-based dialog systems upon the MobileCS2 dataset [4]. The mobileCS2 is a real-life human to human knowledge-grounded dialog dataset, released from the SLT 2024 FutureDial Challenge [4]. It consists of real-world dialog transcripts between real users and customer service staff, along with annotated knowledge pieces necessary for staff to respond properly. Extensive experiments are conducted on the dataset to compare the KAFT method and the prompting method. The experiment results demonstrate that the KAFT technique can substantially outperform the prompting technique in both the RAG and the agent based systems, thereby showing the efficacy of the proposed KAFT method in knowledge-intensive dialog systems.

In summary, the main contributions of this work are:

- This paper proposes to teach the LLMs to make use of external knowledge by finetuning the LLMs with domain-specific data, together with the domain-specific external knowledge, which is called knowledge augmented finetuning (KAFT).
- To validate the efficacy of the proposed KAFT method, RAG-based and agent-based dialog systems are built on the real-life customer service dataset MobileCS2 dataset to compare the proposed KAFT method with the prompting method.
- Extensive experiments on the MobileCS2 dataset show that the proposed KAFT method can improve the ability of LLMs to make use of knowledge and substantially surpass the prompting method in both RAG-based and agent-based dialog systems.

2 Related Work

2.1 Large Language Models (LLMs)

LLMs are large foundation models pretrained with corpus of trillions of tokens. The emergence of LLMs [1, 8] has greatly improved the performance in various NLP tasks. Previous studies have discovered the strong in-context learning [3] and reasoning [30] ability of LLMs, which inspired the researchers to explore LLMs in more complicated tasks like question-answering and dialog systems [34, 24, 29, 22]. Despite their success in open-domain dialogs, the absence of specific domain knowledge and up-to-date facts in the data can pose limitations for those systems in vertical domains. The RAG-based approach [21, 17, 6] and the agent-based approach [25, 27] are used to mitigate this issue. The most commonly adopted implementation of the RAG-based system and agent-based system is to prompt the LLM with instructions and examples, along with the retrieved knowledge [31]. However the LLMs still struggles to effectively utilize the knowledge in the RAG-based and agent-based systems as they lack the background information related to the specific areas [7, 11]. In this work, we propose to use

the KAFT method to finetune the LLMs to gain the ability to make full use of the external knowledge.

2.2 Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) [21] is a technique that enhances the performance of LLMs by utilizing external pre-stored data, such as texts, dialogues, and knowledge bases. Specifically, when the model needs to generate text or provide an answer, it can first retrieve relevant information from external sources and then generate more accurate and enriched outputs by integrating the information retrieved. Therefore, a retrieval augmented generation system typically contains two components, a retriever and an LLM (also called a generator). There are several works that improve the original RAG work, mainly focused on improving the retriever [19, 12, 16, 5] and the generator [14, 32, 2, 20]. Unlike the previous works that focus on general domain question answering, this paper represents the first solid empirical work to investigate the idea that the ability of LLMs to make use of domain-specific knowledge can be improved by KAFT.

Recently, there are some studies to compare RAG with the method of directly finetuning LLMs without using RAG for vertical domains [28, 13, 35]. Unlike those studies, this paper aims to compare the proposed KAFT method with the method of prompting the LLMs in both RAG-based and agent-based dialog systems to show that the ability of LLMs to make use of domain-specific knowledge can be enhanced by post-training.

2.3 Large Language Model based Agents

With the development of LLMs, recent researches have explored the potential of building agents upon LLMs, leveraging their strong generation and understanding abilities. The introduction of LLM based agents has significantly enhanced the ability of machines to interact with the world [25]. Using the superior ability of the LLMs, those agents can plan their actions and interact with tools as human [27, 29]. For knowledge-intensive tasks, the ability to interact with tools is important, as the agent can actively get the information necessary for accomplishing the task like human does. In this work, we explore the possibility of building a customer service agent that can act like real-life customer service staffs.

3 Method

3.1 Task and Definition

In a customer service dialog system, assume we have a dialog X with T turns of user utterances and system responses, denoted by $u_1, r_1, \cdots, u_T, r_T$ respectively. At turn t, based on the dialog context $c_t \triangleq u_1 \oplus r_1 \oplus \cdots \oplus u_{t-1} \oplus r_{t-1} \oplus u_t$ ($\oplus$

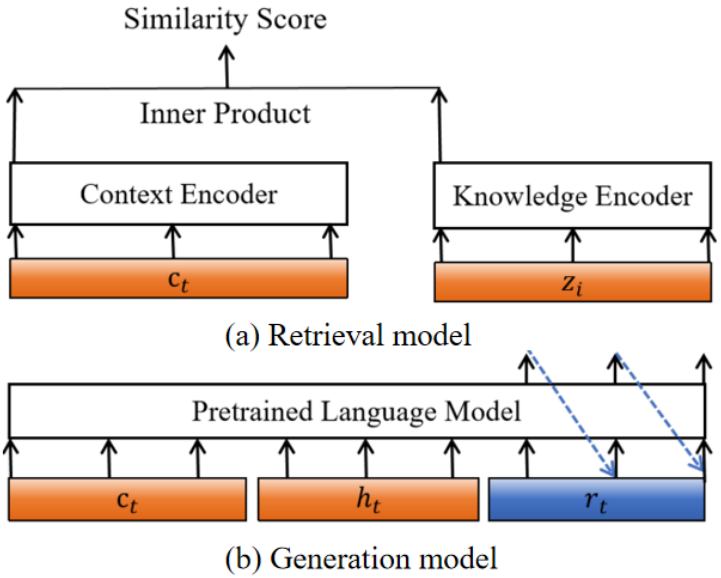

Fig. 2: Overview of the RAG-based dialog systems: (a) the retrieval model, (b) the generation model.

means sequence concatenation), the system needs to generate an appropriate response leveraging the knowledge base (KB). For the MobileCS2 dataset, the knowledge base is made up of the user information (KB_{user}), which is unique for each dialog, the product information list ($KB_{product}$), and the FAQ list for commonly asked questions (KB_{FAQ}).

3.2 Retrieval Augmentation Generation (RAG) based Dialog System

Knowledge Augmented Finetuning (KAFT) of LLMs The RAG based dialog system employs a retriever to retrieve the relevant knowledge pieces from the knowledge base and uses the retrieved knowledge to help the generator (i.e. the LLM) generate the response. In this work, the RAG-based dialog system is similar to the system in [4]. For a dialog X in the MobileCS2 dataset, the knowledge base KB_X can be denoted as: $KB_X \triangleq KB_{user} \cup KB_{FAQ} \cup KB_{product}$. Given the knowledge base KB_X, at turn t of a dialog X, the system uses a retriever $p_\eta(z_i \mid c_t)$, which is shown in Figure 2(a), to obtain the relevant knowledge h_t from the knowledge base and generate appropriate responses with the generator $p_\theta(r_t \mid c_t, h_t)$, which is shown in Figure 2(b).

The retriever is implemented with the dual-encoder architecture including the knowledge piece encoder Encoder_p and the context encoder Encoder_c, as shown in Figure 2(a). To train the retrieval model, for each knowledge piece

z_i $(i = 1, 2, \cdots, K)$ in KB_X, the models fit the retrieval distribution of $p_\eta(z_i \mid c_t)$ as in [21]:

$$p_\eta(z_i \mid c_t) \propto \exp\left(\text{Encoder}_p(z_i)^\top \text{Encoder}_c(c_t)\right) \tag{1}$$

Encoder_p and Encoder_c are both initialized with a BERT-based pretrained model [10]. The log probabilities of the positive pieces $z \in Z_+$ (labeled in the dataset) are optimized:

$$\mathcal{L}_{ret} = -\frac{1}{\mid Z_+ \mid} \sum_{z \in Z_+} \log p_\eta(z \mid c_t) \tag{2}$$

The Encoder_p is fixed during the training, while the Encoder_c is trained with the loss in Eq. 2, following the setting in [19].

KAFT for RAG refers to finetune the LLM with the dialog context and knowledge piece, i.e., to finetune the generation model $p_\theta(r_t \mid c_t, h_t)$. We use the auto-regressive loss to optimize the generation probability:

$$p_\theta(r_t \mid c_t, h_t) = \prod_{l=1}^{|r_t|} p_\theta(y^l \mid c_t, h_t, y^1, \ldots, y^{l-1}) \tag{3}$$

where $|\cdot|$ denotes the length in tokens, and y^l the l-th token of r_t. This is similar to [4]. The baseline system in [4] used oracle knowledge in training the LLM. However, we use the generated h_t from the retriever rather than the annotated h_t so that the training procedure of the generation model is aligned to the test setting where the annotated h_t is not available. This adaption, similar to the noise adding technique in [32, 35], brings some improvement to the system in our experiments. The generation model in Eq. 3 is initialized with a GPT2-based pretrained language model [26].

Prompting of LLMs In the method of *prompting LLMs for RAG*, we use the prompted LLM as the generator, while using the same retriever as in KAFT. The in-context learning (ICL) [3] method is used to prompt the LLM to generate appropriate responses given the examples randomly selected from the dataset. The generation probability of the LLM can be written as $p_\theta(r_t \mid prompt_t, c_t, h_t)$. The prompt $prompt_t$ contains the instruction for the LLM and the example dialogs, which clearly instructs the LLM to generate the response leveraging the retrieved knowledge h_t. The prompt for the LLM, as well as an example of a turn in a dialog using the RAG-based system is shown in Figure 4(a).

3.3 Agent based Dialog System

Knowledge Augmented Finetuning (KAFT) of LLMs The agent based dialog system leverages the planning and search ability of the agent to accomplish the dialog, as shown in Figure 3. The agent consists of a decision maker, API calling, and an LLM as the generator. At turn t in a dialog, the agent first makes the search decision a_t of what database the system needs to search based on the

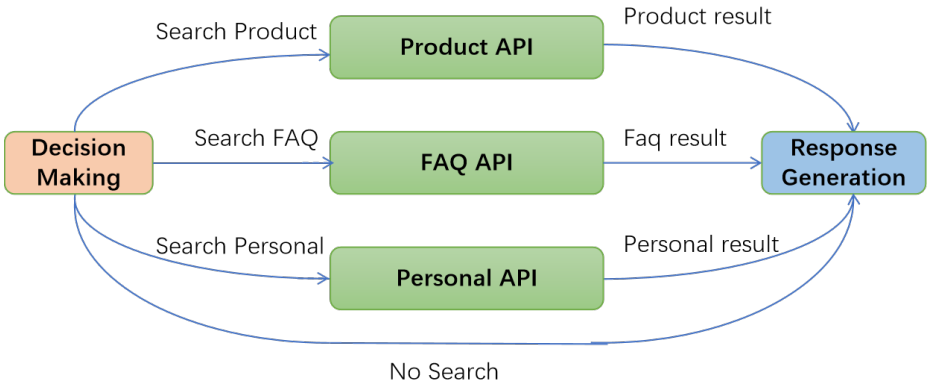

Fig. 3: Overview of the agent-based dialog systems. The system first decides the search intention, then calls the corresponding API to perform the search operation. The system then generates the response based on the search results.

input context c_t. Based on the decision a_t, the agent conducts the corresponding search. If a_t is 'No Search', then the agent responds directly to the context; otherwise, the agent queries the corresponding API for the product, FAQ, and personal information.

To simulate the Product API, the FAQ API, and the Personal APIs, we use the annotated data of the corresponding search decisions in the MobileCS2 dataset. For example, the turns annotated with the search decision of "Search Product" are used to train the Product API. Notably, the purposes of all these search APIs are to retrieve some knowledge pieces from the knowledge bases, which is similar to the retrieval in RAG. Thus, for building the Product API and the FAQ API, we separately train two dual-encoder based retrievers, using the architecture in Figure 2(a) and the loss function in Eq. 2. For the personal API, we return all the personal information to ensure the recall of the information, as the knowledge base for the user information (KB_{user}) is relatively smaller than other knowledge bases.

The search result can be viewed as h_t, denoting a kind of knowledge piece. *KAFT for agent* consists of finetuning LLMs for response generation given h_t and decision making given a_t, respectively. We use the similar auto-regressive loss as in KAFT for RAG to optimize the generation probability $p_\theta(r_t \mid c_t, h_t)$ (Eq. 3). The difference is that h_t in the agent is the result given by the API search rather than from the retriever in RAG. The decision maker of the agent is finetuned based on the probability $p_\theta(a_t \mid c_t)$, also using auto-regressive loss, where the decision a_t, as a token sequence, can be viewed as another kind of knowledge.

Prompting of LLMs In the method of *prompting LLMs for agent*, we employ the prompted LLM to implement decision making and response generation,

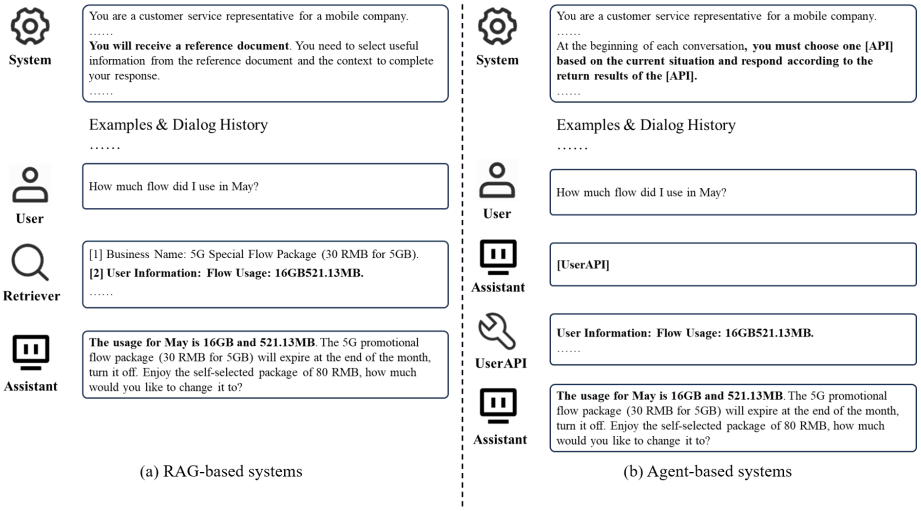

Fig. 4: An illustration of the prompts for the LLMs, as well as an example turn in the dialog in the RAG-based and agent-based systems.

while using the same search API as in KAFT. We use the ICL) method [3] to prompt the LLM to generate appropriate search decisions and responses, given the contexts and the prompts. We give corresponding examples and instructions, which clearly instruct the LLM which task, decision making or response generation, to perform. As there are only four possible search decisions, we enumerate them in the prompts for the decision making and ask the LLM to choose one of the 4 search decisions in the decision making task. The prompt for the LLM, as well as an example of a turn in a dialog using the agent-based system is shown in Figure 4(b).

4 Experiment

4.1 Experiment Settings

The experiments are carried out on a real-life human-human dialog dataset, called MobileCS2, released from the SLT 2024 FutureDial-RAG Challenge [4]. The MobileCS2 dataset is derived from the real-world mobile conversational scenarios and comprises around 3000 carefully annotated dialog logs between customers and customer service staffs. The dataset aims to promote the study of training and testing dialog systems for knowledge-intensive customer service. The dataset was officially split into train, development and test sets, which consist of 1,926, 412 and 413 dialog samples, respectively.

For evaluation, we follow the official scripts in [4]. To evaluate the retriever in RAG and the search APIs in agent, we use the recall metrics and report recall@1, recall@5 and recall@20. To evaluate the whole dialog system, we use three metrics. The generated response is evaluated by measuring the similarity

Table 1: Comparison between the dialog systems built with different methods and settings on the MobileCS2 dataset. "Direct respond" means that we do not use the knowledge base and let the system to directly respond given the context.

Method	Setting	BLEU	BERTScore	Inform	Combined Score
Direct Respond	prompt + 0-shot (GPT3.5)	4.81	0.601	0.003	0.328
	prompt + 5-shot (GPT3.5)	11.4	0.646	0.002	0.382
	finetuning (GPT2)	**17.3**	**0.652**	**0.011**	**0.424**
RAG	prompt + 0-shot (GPT3.5)	17.2	0.657	0.063	0.478
	prompt + 5-shot (GPT3.5)	20.6	0.663	0.059	0.493
	KAFT (GPT2)	**22.2**	**0.668**	**0.145**	**0.590**
Agent	prompt + 0-shot (GPT3.5)	10.4	0.620	0.033	0.395
	prompt + 5-shot (GPT3.5)	18.0	0.645	0.082	0.495
	KAFT (GPT2)	**23.6**	**0.656**	**0.147**	**0.594**

score with the ground truth response (*BLEU* and *BERTScore*) and whether the system correctly provides the requested information by the user (*Inform Rate*). *BLEU* measures the fluency of the generated responses by analyzing the amount of n-gram overlap between the real responses and the generated responses. *BERTScore* [33] measures the semantic similarity of the generated responses with the oracle responses by using a pretrained BERT model. *Inform Rate* refers to how often the system response is able to cover the information requested by the user. The final score is calculated as $score = 0.5*(BLEU/100+ BERTScore) + Inform$, as in the original scripts in [4].

For the KAFT method, we finetune the GPT2 [26] in this study, while for the prompting method, we use the GPT3.5 [24]. In the experiments, hyperparameters are chosen based on the development set and evaluated on the test set.

4.2 Main Results

In the experiments, we examine the efficacy of the KAFT method compared to the prompting method. Based on the results in Table 1, we find that KAFT can greatly boost the performance over prompting in both RAG-based and agent-based systems. While prompting the LLM with 5 examples can improve the performance, the dialog systems built with the prompting method still lag behind the systems built with the KAFT method on the BLEU, BERTScore, Inform and the Score metrics, especially on the Inform metric that requires accurate understanding and utilization of the domain-specific knowledge. The results demonstrate that the proposed KAFT method can substantially improve the ability of LLMs to make use of knowledge.

Moreover, according to the results in Table 2 for agent-based systems, prompting the LLM with examples and instructions cannot perform well in the decision-

Table 2: The decision making accuracy in the agent-based system for the Personal, Product and FAQ search, using the prompting method and the KAFT method respectively.

Setting	Personal	Product	FAQ
0-shot (+prompt)	0.183	0.357	0.005
5-shot (+prompt)	0.290	0.468	0.355
KAFT	**0.381**	**0.580**	**0.475**

Table 3: Comparison of the retrieval performance between the search APIs in the agent system and the retriever in the RAG system, for the Product search and FAQ search tasks.

Task	Model	Recall@1	Recall@5	Recall@20
Product Search	Retriever	0.049	0.132	0.398
	Product API	**0.075**	**0.199**	**0.451**
FAQ Search	Retriever	0.395	0.649	0.782
	FAQ API	**0.546**	**0.782**	**0.872**

making task, which shows a large performance gap behind the KAFT method. Presumably, this is because the complex contexts in real-life customer service dialogs make it difficult for the LLM to accurately predict the search decision given only instructions and examples.

Overall, these results show that by using the KAFT method, the system can be greatly improved on both the response quality and the factual accuracy, mainly because the system is trained to adapt to speaking tunes and thinking manner for the vertical domain. This finding reflects the importance of the proposed KAFT method for building dialog systems for vertical domains.

Note that in the experiments, the KAFT method is implemented with GPT2, which is small. Also note that the main research question investigated in this paper is to systematically compare the prompting and KAFT techniques for the RAG-based and agent-based systems. It is found in our experiments that a small model like GPT2 with KAFT can beat GPT3.5 with prompting in the knowledge-intensive vertical domain, which clearly shows the advantage of the KAFT method. Using GPT-2 suffices to investigate the research question.

4.3 Analysis and Ablation

As shown in Table 1, both the RAG based and the agent based systems show great improvements over the systems that directly respond to the user given the

Table 4: Ablation study about using the retrieved knowledge pieces (denoted by "Retrieve") versus using the annotated knowledge pieces (denoted by "Oracle") in **testing** in the RAG-based system (using retrieved knowlede in training).

Test setting	BLEU	BERTScore	Inform	Score
Retrieve	22.23	0.668	0.145	0.590
Oracle	**48.03**	**0.720**	**0.392**	**0.992**

Table 5: Ablation study about using the retrieved knowledge pieces (denoted by "Retrieve") versus using the annotated knowledge pieces (denoted by "Oracle") in **training** in the RAG-based system (using retrieved knowlede in testing).

Test setting	BLEU	BERTScore	Inform	Score
Oracle	14.09	0.640	0.127	0.517
Retrieve	**22.23**	**0.668**	**0.145**	**0.590**

context, in terms of all the BLEU, BERTScore, Inform and the Score metrics. This finding shows that both the RAG and agent based systems can augment pure generative language models with knowledge. This is crucial in building knowledge-intensive dialog systems.

According to the results in Table 1, the RAG based systems and the agent based systems perform on par with each other. As both the RAG based and agent based systems are competitive in building knowledge-intensive dialog systems, it is interesting to compare the two systems and discuss how to improve these systems in future work. First, on the one hand, from Table 3, we can see that the search APIs in the agent system perform better than the RAG system in the retrieval task. On the other hand, from Table 2, we can observe that the agent system suffers from low decision making accuracy, whether by prompting or by KAFT, while the RAG system has no such limitation. The combined effect is that the agent systems perform close to the RAG system. The performance difference between the RAG systems and the agent systems on the knowledge search task and the decision making task may vary under different domains. Therefore, it is suggested to explore both agent systems and RAG systems for a certain real-world application in order to achieve the best performance.

Second, we examine whether the knowledge pieces provided by the RAG retriever or the agent search API is accurate enough for the language model to generate the ideal responses. The results in Table 3 show that the recall@1 is relatively low for the product and FAQ search, especially the product search, indicating that more efforts should be put into increasing the knowledge retrieval

accuracy for both RAG and agent systems. Moreover, as shown in Table 4, we can find out that using the annotated knowledge instead of the retrieved knowledge in testing can greatly improve the RAG performance, which also emphasizes the importance of accurate knowledge retrieval.

Finally, we examine whether using the annotated knowledge or using the knowledge retrieved by the retriever in the training process in the RAG-based system will yield better performance. The results in Table 5 show that using the retrieved knowledge in the training stage will greatly improve the performance, as the generator needs to discern whether the retrieved knowledge are correct or not. In testing, the oracle knowledge is not provided, and therefore the ability to discern whether the knowledge provided by the retriever is correct is an important skill for a good generator.

5 Conclusion

In this paper, we propose to finetune the LLMs in the dialog systems with domain-specific data, together with the domain-specific external knowledge, which is called knowledge augmented finetuning (KAFT). The proposed KAFT method aims to teach LLMs how to make use of external knowledge. To test the efficacy of the KAFT method, we build RAG-based and agent-based dialog systems with the KAFT method, leveraging the real-life customer service dataset MobileCS2. In our experiments, systems using the KAFT method achieve substantial performance gains over those using the prompting method, particularly in terms of factual accuracy, which shows the efficacy of KAFT in building knowledge-intensive dialog systems. With the KAFT method, the model gains improved capability of making use of external knowledge in both RAG-based and agent-based dialog systems. Furthermore, we conduct ablation studies on the knowledge usage and accuracy in the systems, which shed light on future work on building dialog systems that can provide more accurate responses.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In: ICLR (2024)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. NeurIPS (2020)
4. Cai, Y., Chen, S., Wu, Y., Huang, Y., Feng, J., Ou, Z.: The 2nd FutureDial challenge: Dialog systems with retrieval augmented generation (FutureDial-RAG). In: 2024 IEEE Spoken Language Technology Workshop (SLT). pp. 1091–1098. IEEE (2024)
5. Cai, Y., Li, K., Huang, Y., Feng, J., Ou, Z.: Entriever: Energy-based retriever for knowledge-grounded dialog systems. arXiv preprint arXiv:2506.00585 (2025)

6. Cai, Y., Liu, H., Ou, Z., Huang, Y., Feng, J.: Knowledge-retrieval task-oriented dialog systems with semi-supervision. In: INTERSPEECH (2023)
7. Chen, B., Shu, C., Shareghi, E., Collier, N., Narasimhan, K., Yao, S.: Fireact: Toward language agent fine-tuning. arXiv preprint arXiv:2310.05915 (2023)
8. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research (2023)
9. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al.: Scaling instruction-finetuned language models. Journal of Machine Learning Research **25**(70), 1–53 (2024)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
11. Gao, S., Dwivedi-Yu, J., Yu, P., Tan, X.E., Pasunuru, R., Golovneva, O., Sinha, K., Celikyilmaz, A., Bosselut, A., Wang, T.: Efficient tool use with chain-of-abstraction reasoning. arXiv preprint arXiv:2401.17464 (2024)
12. Glass, M., Rossiello, G., Chowdhury, M.F.M., Naik, A., Cai, P., Gliozzo, A.: Re2G: Retrieve, rerank, generate. In: NAACL-HLT (2022)
13. Gupta, A., Shirgaonkar, A., Balaguer, A.d.L., Silva, B., Holstein, D., Li, D., Marsman, J., Nunes, L.O., Rouzbahman, M., Sharp, M., et al.: Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. arXiv preprint arXiv:2401.08406 (2024)
14. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.W.: REALM: retrieval-augmented language model pre-training. In: ICML (2020)
15. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
16. Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Unsupervised dense information retrieval with contrastive learning. Transactions on Machine Learning Research (2022)
17. Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Few-shot learning with retrieval augmented language models. arXiv preprint arXiv:2208.03299 (2022)
18. 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 (2023)
19. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: EMNLP (2020)
20. Khattab, O., Santhanam, K., Li, X.L., Hall, D., Liang, P., Potts, C., Zaharia, M.: Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. arXiv preprint arXiv:2212.14024 (2022)
21. 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. NeurIPS (2020)
22. Liu, H., Cai, Y., Lin, Z., Ou, Z., Huang, Y., Feng, J.: Variational latent-state gpt for semi-supervised task-oriented dialog systems. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31**, 970–984 (2023)
23. Nori, H., Lee, Y.T., Zhang, S., Carignan, D., Edgar, R., Fusi, N., King, N., Larson, J., Li, Y., Liu, W., Luo, R., McKinney, S.M., Ness, R.O., Poon, H., Qin, T., Usuyama, N., White, C., Horvitz, E.: Can generalist foundation models outcompete special-purpose tuning? Case study in medicine. arXiv preprint arXiv:2311.16452 (2023)

24. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. NeurIPS (2022)
25. Park, J.S., O'Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative agents: Interactive simulacra of human behavior. In: Proceedings of the 36th annual acm symposium on user interface software and technology. pp. 1–22 (2023)
26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog (2019)
27. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language models can teach themselves to use tools. NeurIPS (2024)
28. Soudani, H., Kanoulas, E., Hasibi, F.: Fine tuning vs. retrieval augmented generation for less popular knowledge. In: Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region. pp. 12–22 (2024)
29. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
30. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. NeurIPS (2022)
31. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: The Eleventh International Conference on Learning Representations (2023)
32. Zhang, T., Patil, S.G., Jain, N., Shen, S., Zaharia, M., Stoica, I., Gonzalez, J.E.: Raft: Adapting language model to domain specific rag. arXiv preprint arXiv:2403.10131 (2024)
33. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. In: ICLR (2019)
34. Zhang, Y., Ou, Z., Yu, Z.: Task-oriented dialog systems that consider multiple appropriate responses under the same context. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 9604–9611 (2020)
35. Zhao, Y., Cao, H., Zhao, X., Ou, Z.: An empirical study of retrieval augmented generation with chain-of-thought. In: 2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP). pp. 436–440. IEEE (2024)