Knowledgeable Preference Alignment for LLMs in Domain-specific Question Answering

Anonymous ACL submission

Abstract

Deploying large language models (LLMs) to real scenarios for domain-specific question answering (QA) is a key thrust for LLM applications, which poses numerous challenges, especially in ensuring that responses are both accommodating to user requirements and appropriately leveraging domain-specific knowledge. They are the two major difficulties for LLM application as vanilla fine-tuning falls short of addressing. Combining these requirements, we conceive of them as the requirement for the model's **preference** to be harmoniously aligned with humans'. Thus, we introduce **Know**ledgeable **P**reference **A**lignmen**T** (KnowPAT), which constructs two kinds of preference sets to tackle the two issues. Besides, we design a new alignment objective to align the LLM preference with different human preferences uniformly, aiming to optimize LLM performance in **real-world, domain-specific** QA settings. Adequate experiments and comprehensive comparisons with 15 baseline methods illustrate that our KnowPAT is a superior pipeline for real-scenario domain-specific QA with LLMs. Our code is available at this anonymous github link.

1 Introduction

In contemporary digital commerce platforms, the deployment of automated and intelligent **question-answering** (QA) services is a pivotal task to augment service quality. These services are designed to furnish answers to domain-specific customer queries. Building such a domain-specific QA system, while highly sought after, remains a daunting challenge in practical scenarios.

Domain-specific QA necessitates a comprehensive understanding of a specific domain to answer specialized questions. However, traditional deep learning models (Devlin et al., 2019; Raffel et al., 2020) still have insufficient domain-specific expertise. This makes the domain **knowledge graph**

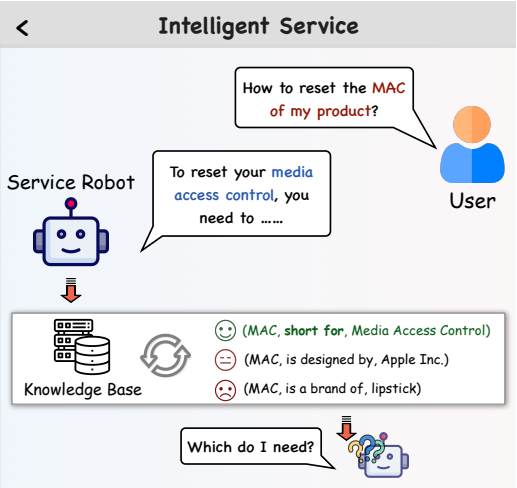

Figure 1: A simple case of intelligent service for cloud products. Such a simple example is meant to illustrate the importance of selective use of retrieved knowledge as MAC is a terminology in computer networking rather than a kind of computer or lipstick in the user context.

(KG) (Liang et al., 2022) a pivotal tool for the storage and querying of domain knowledge. KGs can store human knowledge in the triple form, offering a unified, maintainable, and extensible representation of the knowledge from heterogeneous sources. The utility of KGs has already been demonstrated across various application scenarios such as E-commerce (Zhu et al., 2021), and health care (Li et al., 2020). Within the context of QA, incorporating KGs as an external knowledge source represents a promising approach, which is known as KG-based QA (Jiang et al., 2023b).

Meanwhile, as large language models (LLMs) (West et al., 2023) achieve significant progress and exhibit substantial proficiency within numerous NLP fields (Zhu et al., 2023), applying LLMs into various downstream tasks have been a predominant trend in industry (Zhang et al., 2023a). Contrasting earlier pre-trained language models (Devlin et al., 2019; Raffel et al., 2020), LLMs trained on the massive corpus have outperformed text generation capabilities which perform better when interact-

ing with human (Ouyang et al., 2022). To adapt the LLMs for downstream usage, supervised fine-tuning (SFT) (Zhang et al., 2023e) is applied to fit the model with specific tasks and data. However, the LLM application for real-scenario QA with external KG remains an underexplored domain, with limited work addressing this intersection.

Our goal entails the resolution of a challenge in real-world applications: **how can LLM be used to solve real-scenario QA problems supported by external knowledge graphs?** A generic pipeline for this problem is the retrieve-augmented generation (RAG) (Tian et al., 2023), which first retrieves relative knowledge triples for the question as reference data and subsequently fine-tunes the LLM with knowledge-enhanced prompt. However, this conventional approach often encounters obstacles in practical scenarios. Firstly, the LLM-produced responses must prioritize user-friendliness, avoiding any generation of inappropriate or unfriendly content. Secondly, the retrieved knowledge is not invariably useful, necessitating that LLMs develop the capacity to judiciously exploit knowledge. Figure 1 illustrates a simple case in which retrieved knowledge is not always desperately needed (e.g., MAC is a kind of lipstick), which requires the LLMs to selectively utilize the retrieved knowledge instead of generating answers without thoughtful consideration. These two issues can uniformly collectively constitute the preference problem of LLMs. LLMs have their **style preference** to generate contents and **knowledge preference** to selectively use the retrieved knowledge in the prompt. **As a practical application, the preference of LLMs needs to align with human expectations and requirements for better service**. This refers to preference alignment (PA) (Yuan et al., 2023), a burgeoning topic in the LLMs community, which would incorporate human preference to tune the LLMs during training. PA aims to control the model to generate human-preferred content and avoid unpreferred content. However, the scenarios faced by current PA works tend to be generic. No research has been explicitly directed towards domain-specific applications such as our scenario, providing impetus for further exploration.

In this paper, we propose a novel three-step **Know**ledgeable **P**reference **A**lignmen**T** (KnowPAT) pipeline to address the domain-specific QA task for a real-scenario LLM application. KnowPAT propose **knowledgeable preference set construction** to incorporate domain KGs to construct knowledgeable preference data. Besides, a new alignment objective is designed to optimize the LLM with the knowledge preference. Our contribution can be summarized as three-folded:

(1). We are the first work that introduces preference alignment for domain-specific QA with LLMs and domain KGs, which is an industrial practice with practical applications.

(2). We propose a knowledgeable preference alignment (KnowPAT) framework to incorporate KGs into the preference alignment process of LLMs. We balanced the need for both style and knowledge preference and devised a new training objective to align the LLM with human preference.

(3). We conduct comprehensive experiments to validate the effectiveness of our methods by automatic and human evaluations, which shows that KnowPAT stands as a paramount option for real-world applications, outperforming 15 existing baselines.

2 Problem Setting

In this section, we will first introduce our problem scenario and basic notations.

Our overall target is to fine-tune a LLM $\mathcal{M}$ with our QA datasets $\mathcal{D} = \{(q_i, a_i) \mid i = 1, 2, \ldots, N\}$ where q_i, a_i represent a question and answer pair. The questions in the dataset are all about common usage issues with our cloud products while the questions and answers are manually collected and labeled, which are gloden answers with decent and knowledgeable responses. For vanilla fine-tuning (VFT), we first wrap the QA pair with a prompt template $\mathcal{I}$ and the model $\mathcal{M}$ is autogressively (Brown et al., 2020) optimized as:

$$\mathcal{L}_{ft} = -\frac{1}{|a_i|}\sum_{j=1}^{|a_i|}\log P_{\mathcal{M}}(a_{i,j}|\mathcal{I}, q_i, a_{i,<j}) \quad (1)$$

where $a_{i,j}$ is the j-th token of a_i and $P_{\mathcal{M}}$ denotes the token probability predicted by the model $\mathcal{M}$. With such a training objective, the training QA data serves as the supervision information to tune the model $\mathcal{M}$ to the QA scenario. Besides, as a domain-specific task, we construct a **cloud product knowledge graph** (CPKG) based on the product documents maintained in the real production environment. The CPKG is denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ are the entity set, relation set, and triple set respectively. The knowledge graph will be used as an external knowledge source to support the model for QA. By retrieving top-k knowledge with higher relevance, the input prompt will in-

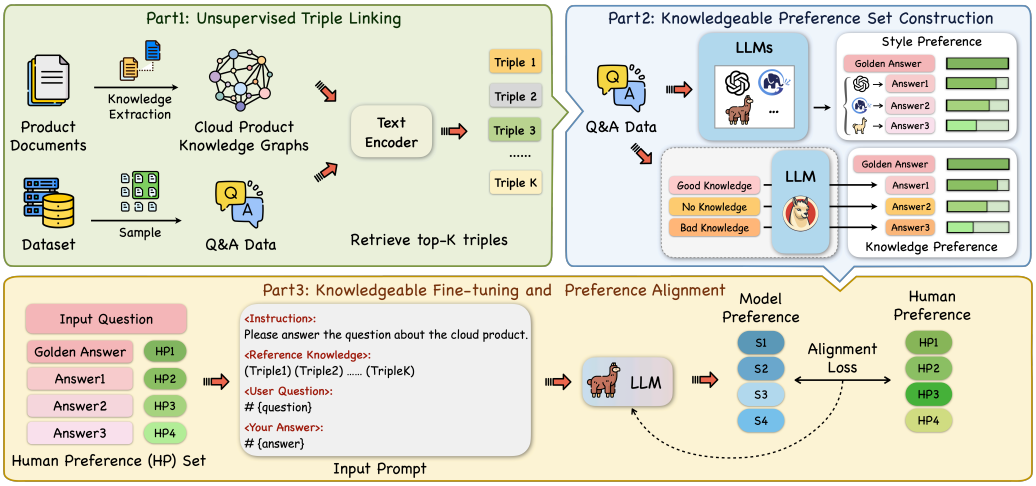

Figure 2: The overall architecture of KnowPAT. We design three important modules in our framework: unsupervised triple linking (part 1), knowledgeable preference set construction (part 2), and knowledgeable preference alignment (part 3). We first retrieve relative knowledge triples for the question in part 1 and apply the retrieved knowledge to construct the knowledgeable preference set in part 2. The preference sets will participate in the fine-tuning and preference alignment process in part 3, which will align the LLM with human preference.

corporate the retrieved knowledge $\mathcal{K}$. Thus, $\mathcal{M}$ can learn the relative knowledge during the VFT process, which is a general pipeline for domain-specific LLM applications.

However, such a VFT approach can not achieve pretty good results for the domain-specific QA. On the one hand, applications in real scenarios should be user-friendly, otherwise, they will not bring commercial value. Thus, the text style of the generated response should be more acceptable for users. On the other hand, the knowledge retrieval process is unsupervised and the effectiveness of the retrieved knowledge is hard to guarantee, which means that the model $\mathcal{M}$ needs to acquire the ability to judge and selectively utilize the knowledge triples. Therefore, we should improve the basic VFT to solve these two problems.

Actually, both of these problems can be summarised as model preference. The LLM $\mathcal{M}$ has its style preference to generate texts and its knowledge preference to selectively utilize the retrieved knowledge. For the model to be practically applicable, the model preference should align with human preference, aiming to generate high-quality answers that humans prefer. Preference alignment (PA) is an important topic for LLMs. To apply PA during LLM fine-tuning, we sample a preference set $\mathcal{P} = \{b_1, b_2, \ldots, b_l\}$ with l different answers for each QA pair (q, a). We denote r_i as the preference score of each answer b_i where higher r_i represents that humans prefer this answer. During training, we will define another objective $\mathcal{L}_{align}$ to align the model $\mathcal{M}$ with the preference set $\mathcal{P}$, aiming to increase the probability of a human preferred answer appearing and simultaneously decrease the probability of an unpreferred answer. The human preference of each answer is the preference score r. The overall training objective then becomes $\mathcal{L} = \mathcal{L}_{ft} + \mathcal{L}_{align}$.With such a multi-task objective, the LLM is fine-tuned to **fit the golden answers while avoiding unpreferred results**. The next question is how to generate a preference set to reflect both the style and knowledge preference.

3 Our KnowPAT Pipeline

In this section, we will present our pipeline of knowledgeable preference alignment (KnowPAT), which consists of three key parts: unsupervised triple linking, knowledgeable preference set construction, fine-tuning, and training. Figure 2 demonstrates an intuitive view of the three parts in our pipeline design.

3.1 Unsupervised Triple Linking

The first key parts is the unsupervised triple linking which aims to link the triples in the CPKG $\mathcal{G}$ to each question q_i. We design a simple semantic similarity-based retriever $\mathcal{H}$ to achieve this goal. The similarity between the i-th question q_i and the j-th triple (h_j, r_j, t_j) is:

$$sim(i, j) = \textbf{Cosine}(\mathcal{H}(q_i), \mathcal{H}(h_j, r_j, t_j)) \quad (2)$$

where the retriever $\mathcal{H}$ serves as a textual encoder and we treat both the question and knowledge triple as a text sequence to get their sentence representations. The similarity is based on the cosine similar-

ity of the two representations. We retrieve the top-k triples with the highest similarities for each question q_i and denote the **retrieved knowledge** (RK) as $\mathcal{K}$. RK will be added into the input prompt as the background knowledge for the current question.

This process is unsupervised as we have no manually labeled question-knowledge pairs. Besides, our model will be deployed for real scenario usage, so it also requires strong zero-shot generalization capabilities to new questions. For these two reasons, the retrieved knowledge $\mathcal{K}$ might be noisy and useless to provide background knowledge. We think that the LLM $\mathcal{M}$ should learn the knowledge preference to select helpful information from the retrieved knowledge $\mathcal{K}$.

3.2 Knowledgeable Preference Set Construction

Motivated by such goal, we propose a knowledgeable preference set construction process to enable retrieved knowledge in the preference set construction, which consists of two parts: the style and the knowledge preference set.

For the **style preference set (SPS)** $\mathcal{P}_s$, we select $l-1$ different LLMs denoted $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{l-1}$. These different LLMs $\mathcal{M}_i$ have different textual comprehension and expression skills, which can generate answers with different text styles. The ability and quality of these models to answer domain-specific questions are inferior compared to human-labeled golden answers. The $l-1$ answers generated in this way and golden answers form a style preference set $\mathcal{P}_s = \{b_1, b_2, \ldots, b_l\}$ with length l. For the **knowledge preference set (KPS)**, we assume that the knowledge triples that have high similarity but do not reach the top-k rank are more likely to be knowledge that is not useful for the input question. We can get preference sets with different quality by retrieving some relatively worse knowledge and prompting the model to generate responses with knowledge of different quality. In our design, we retrieve 3 groups of knowledge triples $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ from the CPKG. $\mathcal{K}_1$ represents the retrieved top-k triples, $\mathcal{K}_2 = \emptyset$ is an empty set with no retrieved knowledge. $\mathcal{K}_3$ represents the triples with top $k+1$ to $2k$ similarities which we think are easily misused knowledge with relatively high semantic similarity. Then we wrap the different knowledge $\mathcal{K}_i$ with the input prompt $\mathcal{I}$ into the LLM $\mathcal{M}$ and generate different answers. These generated 3 answers and the golden answer form a knowledge preference set $\mathcal{P}_k = \{c_1, c_2, c_3, c_4\}$.

By doing this, we can get two preference sets for each QA pair. To simplify the setting, we set $l = 4$ to let the two sets be of the same size. Besides, we design a rule-based strategy to decide the preference score r for each answer. For the style preference set $\mathcal{P}_s$, the high-quality golden answer b_1 is assigned with the highest score, and answers from other LLMs were determined by their general capabilities. In practice, we choose three different LLMs ChatGPT (b_2) (Ouyang et al., 2022), ChatGLM-6B (b_3) (Zeng et al., 2023), and Vicuna-7B (b_4). The results of several LLM ranking lists indicate that the three are ranked in order of ability as follows ChatGPT > ChatGLM > Vicuna. Besides, after verification by human experts, we also believe that the quality of the answers generated by these three models in our QA scenarios also conforms to this rule. Thus, the preference scores are assigned in this order: $r_1 > r_2 > r_3 > r_4$.

Meanwhile, for the knowledge preference set $\mathcal{P}_k$, the golden answer c_1 still has the highest preference score r_1. The answer c_2 generated with top-k knowledge $\mathcal{K}_1$ has the second highest preference. The answer c_3 generated with no extra knowledge $\mathcal{K}_2$ has the third highest preference, and the answer c_4 generated with knowledge $\mathcal{K}_3$ is the worst. We found in our actual tests that the mismatch rate between the retrieved knowledge $\mathcal{K}_3$ and the question q is very high and easily misleads the model $\mathcal{M}$, so we set its score to be lower than the case of the empty knowledge $\mathcal{K}_2$. Thus, for the knowledge preference set $\mathcal{P}_k$, the preference scores are still in the order: $r_1 > r_2 > r_3 > r_4$. For each QA pair, we can construct two preference sets and we finally get the whole preference data with $2N$ preference sets. The preference data will participate in the fine-tuning process to control the style preference and knowledge preference for the model $\mathcal{M}$. Note that the size of the two preference sets need not be strictly same, and we have adopted the above formulation for the sake of uniformity of representation in our paper.

3.3 Fine-tuning and Preference Alignment

In addition to the vanilla fine-tuning loss $\mathcal{L}_{ft}$ with the golden answer, the preference data will also participate in the training process. For each preference set, the preference score r_i of the i-th answer represents our degree of preference. We expect the model $\mathcal{M}$ to align with our preference. Thus, we design another score to represent the preference of the model, which is denoted as:

$$\mathcal{S}_i = \frac{1}{|a_i|} \sum_{j=1}^{|a_i|} \log P_{\mathcal{M}}(a_{i,j} | \mathcal{I}, q_i, a_{i,<j}) \quad (3)$$

This score $\mathcal{S}_i$ is the average log-likelihood of each answer token conditioned on the given prompt template $\mathcal{I}$ and question q_i. Higher scores represent a higher probability that the model considers the current answer to occur. To align the model preference with our envision, we designed a new alignment objective for our scenario. The alignment objective is denoted as:

$$\mathcal{L}_{align} = -\sum_{i=1}^{|\mathcal{P}|-1} \left(\log \sigma(\mathcal{S}_i) + \sum_{r_j < r_i} \log \sigma(-\mathcal{S}_j) \right) \quad (4)$$

where σ is the sigmoid function. Such an objective is newly proposed by us to achieve the preference alignment process, which contrasts the preferred answer and the unpreferred answers. It is worth noting that the human preference scores r_i will only determine the ordering corresponding to different answers and will not be directly involved in the computation and gradient accumulation. Existing methods like RRHF (Yuan et al., 2023) and SLiC-HF (Zhao et al., 2023) apply a margin-rank loss in the form $\sum_{r_j<r_i} \max(0, \lambda - \mathcal{S}_i + \mathcal{S}_j)$ to achieve preference alignment. But their design only optimizes the model preference when the model preference score $\mathcal{S}$ of a human preferred answer is lower than an unpreferred answer (a more formalized formulation would be $\mathcal{S}_i < \mathcal{S}_j$ when $r_j < r_i$). However, we think that the preference should still be optimized in this situation and propose such a training objective to continuously decrease the occurrence probability of the unpreferred answers. Meanwhile, as different answers have different text quality and preference degrees, we further design an adaptive weight to control the influence of each preferred answer, which is denoted as:

$$\mu_i = \frac{\mathcal{S}_i - \mathcal{S}_{min}}{\mathcal{S}_{max} - \mathcal{S}_{min}} \quad (5)$$

where $\mathcal{S}_{max}$ and $\mathcal{S}_{min}$ are the max and min model preference scores in a preference set $\mathcal{P}$. With such an adaptive weight, the influence of the answers with different preferences could be dynamically adjusted. The alignment loss then becomes:

$$\mathcal{L}_{align} = \sum_{i=1}^{|\mathcal{P}|-1} \mu_i \left(\log(1 + e^{-\mathcal{S}_i}) + \sum_{r_j < r_i} \log(1 + e^{\mathcal{S}_j}) \right) \quad (6)$$

The final training objective is still in a multi-task manner and we add a hyper-parameter λ as the coefficient of the alignment loss:

$$\mathcal{L} = \mathcal{L}_{ft} + \frac{\lambda}{|\mathcal{P}| - 1} \mathcal{L}_{align} \quad (7)$$

where $\mathcal{P}-1$ represents the count of prefer-unprefer contrast to normalize the alignment loss. For each preference set constructed in the previous section, the model is trained and optimized with such an objective.

4 Experiments and Analysis

In this section, we present the detailed experimental settings and analyze the experiment results to investigate the following four research questions:
(i) RQ1: How does KnowPAT perform compared with the baseline methods?
(ii) RQ2: Do the proposed modules in KnowPAT really benefit the performance of KnowPAT?
(iii) RQ3: Are there some intuitive cases to demonstrate the effectiveness of KnowPAT.
(iv) RQ4: Does the LLM still keep the general ability rather than catastrophic forgetting?

These four questions evaluate our approach on four dimensions: performance, design soundness, intuition, and usability in real scenarios. We will answer the four questions in the following sections.

4.1 Experiment Settings

4.1.1 Dataset Information

The dataset we used in our experiment consists of two parts. The first part is the CPKG with 13995 entities, 463 relations, and 20752 triples. The second part is the QA dataset with 8909 QA pairs. We split the dataset into 7909/500/500 for training/validation/test. For each data instance in the training, we construct two preference sets and get 15818 preference sets with 4 answers in each set.

4.1.2 Baseline Methods

To make a comprehensive study, we select four types of different baseline methods to demonstrate the effectiveness of our preference alignment approach. We not only want to show that alignment is a better framework for LLM application compared to other paradigms (e.g. zero-shot reasoning, in-context learning (Dong et al., 2023), vanilla fine-tuning (Ouyang et al., 2022; Fang et al., 2023)), but also to show that our method is better than other preference alignment methods (Yuan et al., 2023; Zhao et al., 2023; Song et al., 2023; Wang et al.,

Table 1: The experimental results for traditional text generation metrics. We reproduce four types of baseline methods to make a comprehensive comparison. For zero-shot approaches, we select several popular LLMs as the backbone. For other methods, Atom-7B is employed as the backbone. The **red** numbers represent the improvement of KnowPAT. The best baseline performance is underlined.

Type	Setting	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	CIDEr	METEOR
Zero-shot Reasoning	Vicuna	14.18	7.89	5.02	2.69	16.31	6.15	15.69	2.03	17.96
	ChatGLM	14.21	8.36	5.41	2.79	15.38	5.64	14.75	0.95	19.34
	Baichuan	15.51	9.08	5.86	2.87	16.74	6.64	15.81	1.86	19.71
	Atom	10.07	4.11	2.06	8.15	6.24	1.99	6.02	0.87	11.31
	ChatGPT	13.09	7.72	4.93	2.59	16.96	6.68	16.15	2.98	19.52
In-context Learning	Atom(1-shot)	8.97	3.84	1.88	0.53	7.49	1.99	7.31	1.34	10.41
	Atom(2-shot)	9.11	3.84	1.85	0.5	7.34	1.82	7.01	0.99	9.88
	Atom(4-shot)	8.18	3.42	1.65	0.48	7.07	2.04	6.91	1.77	8.83
	Atom(8-shot)	7.79	3.29	1.7	0.79	6.57	1.38	6.41	1.62	8.19
Fine-tuning	w/o KG	14.33	8.85	6.81	5.71	14.33	5.01	14.26	22.92	15.29
	w/ KG (RAG)	14.89	9.35	7.33	6.05	14.77	5.57	14.61	21.34	15.99
Alignment	RRHF	11.99	6.32	4.52	3.47	12.56	4.08	12.29	5.39	12.62
	SLiC	16.55	10.34	7.99	6.53	14.69	5.03	14.48	26.55	16.95
	PRO	18.27	12.36	10.04	8.41	17.07	6.75	16.85	28.46	19.17
	AFT-BC	18.39	12.17	9.86	7.81	18.09	7.14	17.76	33.04	19.48
	AFT-DC	15.34	8.44	5.94	4.35	14.51	5.59	14.15	13.22	16.31
KnowPAT		**22.56** ↑22.67%	**16.66** ↑34.79%	**14.26** ↑42.03%	**12.11** ↑43.99%	**20.28** ↑12.10%	**9.09** ↑27.31%	**19.91** ↑12.11%	**54.86** ↑66.04%	**23.62** ↑21.25%

Table 2: The experimental results of model-based metrics. We report the BERTScore, reward score, and perplexity (PPL) for KnowPAT and the baseline methods. The best result of each metric is bold and the second best is underlined.

	BERTScore↑	Reward↑	PPL↓
VFT	66.24	**-1.64**	31.13
RRHF	64.48	-1.67	31.26
SLiC	66.69	-1.74	32.51
PRO	67.41	-1.78	32.37
AFT	66.16	-2.25	30.11
KnowPAT	**69.34**	-1.69	**29.93**

2023b). The detailed information of the baselines are shown in Appendix B.1.

4.1.3 Evaluation Metrics

To make a comprehensive evaluation of the experimental results, we employ the different evaluation metrics from three aspects: traditional text generation metrics (BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015), and METEOR (Banerjee and Lavie, 2005)), model-based metrics (BERTScore (Zhang et al., 2020a), PPL), and manual evaluation. The detailed information of the evaluation metrics refers to Appendix B.2.

4.1.4 Implementation Details

In our experiment, we select Atom-7B [1] as the backbone LLM $\mathcal{M}$, which is an open-source version of Llama2 (Touvron et al., 2023b,a) with Chinese vocabulary extension. As our dataset is mainly in Chinese, we choose Atom-7B-chat to be our backbone model for experiments. Another consideration for us is that using the open-source Llama architecture model enhances the generality of our method to maintain the community ecology of LLMs. For unsupervised triple linking, BGE-base-zh-v1.5 (Xiao et al., 2023) is applied as the retriever $\mathcal{H}$ to encode and retrieve relative knowledge triples.

During training, we tune the backbone model with bf16 float precision. The training epoch is set to 3 and the gradient accumulation step is set to 8. We optimize the model using AdamW optimizer (Loshchilov and Hutter, 2019) while the learning rate is fixed to $3e^{-4}$. The coefficient hyperparameter λ is search in $\{1, 0.1, 0.01, 0.001\}$.

4.2 Main Results (Q1)

The main results of the traditional metrics are shown in Table 1. As we mentioned before, the traditional metrics can measure the similarity between the generated answer and the golden answer. From the results, we can observe that KnowPAT achieved obvious improvements compared with the baseline methods. We can conclude that KnowPAT achieves a more significant improvement in the BLEU-3(42.03%)/BLEU-4(43.99%) than BLEU-1(22.67%)/BLEU-2(34.79%), which means that KnowPAT makes more significant progress in capturing some complex phrases and discourse. Corresponding to our cloud product QA scenario, these

[1] https://github.com/FlagAlpha/Llama2-Chinese

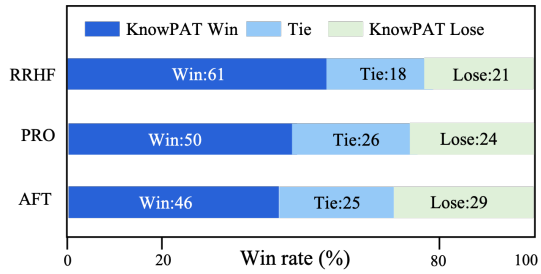

Figure 3: The human evaluation results. For each competition, we randomly select 100 questions and compare the generated results of the two methods.

Table 3: The ablation study results. We evaluate various stripped-down versions of our model to compare the performance gain brought by different components. The full names of these abbreviations are as follows: FT (fine-tuning); AW (adaptive weight); SPS (style preference setting); KPS (knowledge preference setting); RK (retrieved knowledge).

Setting	**BLEU-1**↑	**ROUGE-1**↑	**Reward**↑	**PPL**↓
KnowPAT	22.56	20.28	-1.69	29.93
w/o FT	13.17	12.91	-2.14	31.96
w/o AW	21.87	19.91	-1.71	30.84
w/o SPS	17.57	17.66	-1.75	31.08
w/o KPS	16.12	16.51	-1.79	30.82
w/o RK	17.46	17.56	-1.89	30.85
w/o KG	15.09	16.55	-2.09	33.50

complex phrase usages are usually specialized terms that have a critical impact on the quality of the answer.

Besides, we evaluate our methods with three model-based metrics BERTScore (Zhang et al., 2020a), reward score (Yuan et al., 2023), and PPL (Yuan et al., 2023), which is shown in Table 2. We can observe that KnowPAT still achieves good performance in the model-based metrics such as BERTScore and PPL, which means that the results generated by KnowPAT are more acceptable for the language models. For the reward score, relatively good results have also been achieved by KnowPAT.

Further, we conduct a human evaluation for our method and baseline methods. The two results from the two models are shown to the human evaluator anonymously so that the human evaluator can choose a better result. The model which generates that result will get one point and the competition results are shown in Figure 3. We can observe from the figure that our method generates answers that are more acceptable to humans compared to other baselines, maintaining a relatively high win rate in the competition. Only a small number of times does KnowPAT perform weaker than the baselines, and most of the time KnowPAT is equal or even better. Therefore, combining the above three different perspectives of evaluation, we can conclude that KnowPAT achieves outperforming results in the cloud product QA scenario.

4.3 Ablation Study (Q2)

We conduct Ablation experiments to verify the validity of each module design. We validated the effectiveness of the designed components in our KnowPAT. We can find that the fine-tuning objective $\mathcal{L}_{ft}$ and the alignment objective $\mathcal{L}_{align}$ are both contributing to the model performance. Without fine-tuning (FT), the model performance can take a serious dip, as the LLM is not tuned to fit the golden answer. Besides, both two preference sets (SPS and KPS) in KnowPAT are contributing to the performance. The adaptive weights (AW) can control for the participation of different quality samples in the loss, which is also effective in KnowPAT.

Besides, we demonstrate the necessity of the CPKG with two groups of experiments. w/o RK denotes the experiment that removes the retrieved knowledge in the input prompt during the fine-tuning and preference alignment process. w/o KG denotes the experiment without KG in the whole process, which means the KPS and RK in the input prompt are all removed. For the results of these two groups of experiments, we can observe that the CPKG plays a remarkable role in KnowPAT. In the design of KnowPAT, the CPKG does not only serve as an external knowledge source during training but also participates in the preference set construction process, which is important to the model performance. In summary, each detailed design in our method KnowPAT has its unique role and contributes to the overall performance.

4.4 Case Study (Q3)

To make an intuition for the effectiveness of our method, we conduct a case study as shown in Table 4. We can observe that the answers generated by KnowPAT are more similar to the golden answer while keeping a user-friendly tone and providing sufficient information such as the host parameters in the second case. This suggests that the model learns appropriate style preferences. Besides, the retrieved knowledge in the first case is (EIP, used for, IP Binding), (Select Box, belongs to, Alarm Management Component), etc., which are all helpless to answer this question. However, KnowPAT is

Table 4: The case study results for ground truth (GT), our KnowPAT predictions, and RRHF (Yuan et al., 2023) results. The original Chinese text have been translated into English for clarity.

Question	Please provide the steps for handling IOPS detection errors.
GT	It is recommended to replace the disk with one that meets the IOPS specification.
Ours	It is recommended to replace the server with device that meets the IOPS specifications.
RRHF	After ADAC troubleshooting, restart the business and check whether it is valid.
Question	What is the explanation for the hwFlowRestoreFailed alarm in CloudEngine 1800V product?
GT	The switch flow table restore failed (host_ip= [host_ip], host_name=[host_name])
Ours	The switch flow table restore failed. (host_ip= [host_ip], host_name=[host_name])
RRHF	Flow table restore failed

not misled by this useless knowledge and generates the correct answer while RRHF falls into the trap.

4.5 Knowledge Retention Analysis (Q4)

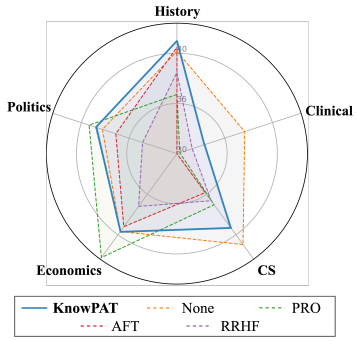

Figure 4: The commonsense ability on five domains.

As a project that needs to get off the ground in real-world scenarios, the general ability of the trained model should also be carefully evaluated, because the user may ask various kinds of questions if they like the model. We expect the model to keep their existing knowledge learned during pre-training and obtain new knowledge about our domain. Thus, we also conduct a commonsense evaluation on the trained models with the CMMLU (Li et al., 2023) dataset, which is a benchmark for LLM's Chinese ability evaluation. The evaluation result is shown in 4. We demonstrate the general ability on five distinctive commonsense regions (history, clinical, politics, computer science, and economics) for KnowPAT, vanilla Atom-7B (none), and other PA methods. As can be seen from the radargram, there is a relatively significant decline in the KnowPAT's ability in medicine, but in the areas of politics, history, and economics it still maintains the ability of the original backbone model and even grows slightly. PRO, while unexpectedly showing a significant improvement in the economics problem, shows a more pronounced performance degradation than KnowPAT in several other areas. Taken together, such variations of KnowPAT in generalized ability are acceptable for our cloud product QA scenario.

5 Related Works

Preference alignment (PA) (Wang et al., 2023d; Cheng et al., 2023) seeks to tailor pre-trained LLMs to align with human preferences (feedbacks) (Ouyang et al., 2022). RLHF is a landmark work for PA, which leverages reinforcement learning (RL) (Schulman et al., 2017) to align human preference with LLMs. Due to the sensitivity of RL parameters and the intricate three-stage processes of RLHF, many PA approaches have been proposed to address these challenges. For example, RRHF (Yuan et al., 2023) propose a margin-rank loss to optimize the LLMs without the need for extra reward models. PRO (Song et al., 2023) optimizes complex preference data with a list-wise contrastive loss. DPO (Rafailov et al., 2023) propose a direct preference optimization method by treating the LLM itself as a reward model. AFT (Wang et al., 2023b) propose a ranking-feedback boundary-constrained alignment loss to optimize the preference data. Besides, our work also focuses on the large language model application and knowledge-enhanced QA. We give a brief introduction of these fields in Appendix A.1 and A.2.

6 Conlusion

In this paper, we introduce a novel framework, knowledgeable preference alignment (KnowPAT), for domain-specific QA tasks in cloud product services, leveraging LLMs and KGs in a practical application setting. Our approach constructs a knowledgeable preference set by retrieving and utilizing knowledge triples to generate answers with different quantities. A new alignment objective is designed to unleash the power of the preference set. Comprehensive experiments demonstrate that our method surpasses existing solutions for this real-world challenge. Looking ahead, we aim to apply KnowPAT to more real scenarios such as enterprise-class services and further investigate the potential of KG-enhanced LLM application in the future.

Limitations

In this paper, we mainly focuses on a real-world application problem to align LLMs with knowledge preference for better domain-specific QA. There are still some limitations in our work.

Domain-specific scenario. Our approach is designed for specific domain (cloud product QA in our paper), and its effectiveness on general domains and open-source datasets is still subject to further validation. This will be the goal of our future endeavours.

Forms of external knowledge. In our paper, we apply knowledge graphs (KGs) to store the external background knowledge for the QA tasks. This is a convenient and efficient way of storing knowledge for our scenario, but in more other scenarios, knowledge may be stored in other forms (e.g. unstructured text). Therefore, a more general framework to process the external knowledge with any format (KGs, unstructured text, documents) should be considered for better usage, which is also our future plan.

Ethical Considerations

In this paper, we employ the open-source LLM to validate the effectiveness of our approach. Besides, the dataset we used is manually labeled with golden answer from domain experts engaged legally with suitable work intensity and well above aversage wages. Their rights are well protected at work. The content of the dataset is mainly questions about our cloud product usage, which do **not involve private information and sensitive data** of the target users. We promise that the content and collection steps of our dataset that are not against scientific ethics.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. In *ICCV*, pages 2425–2433. IEEE Computer Society.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *CoRR*, abs/2306.04136.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, pages 65–72. Association for Computational Linguistics.

Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangnan He, and Qi Tian. 2023a. A bi-step grounding paradigm for large language models in recommendation systems. *CoRR*, abs/2308.08434.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023b. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *RecSys*, pages 1007–1014. ACM.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Hao Chen, Runfeng Xie, Xiangyang Cui, Zhou Yan, Xin Wang, Zhanwei Xuan, and Kai Zhang. 2023a. LKPNR: LLM and KG for personalized news recommendation framework. *CoRR*, abs/2308.12028.

Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Z. Pan, Zonggang Yuan, and Huajun Chen. 2021. Zero-shot visual question answering using knowledge graph. In *ISWC*, volume 12922 of *Lecture Notes in Computer Science*, pages 146–162. Springer.

Zhuo Chen, Yufeng Huang, Jiaoyan Chen, Yuxia Geng, Yin Fang, Jeff Z. Pan, Ningyu Zhang, and Wen Zhang. 2022. Lako: Knowledge-driven visual question answering via late knowledge-to-text injection. In *IJCKG*, pages 20–29. ACM.

Zhuo Chen, Wen Zhang, Yufeng Huang, Mingyang Chen, Yuxia Geng, Hongtao Yu, Zhen Bi, Yichi Zhang, Zhen Yao, Wenting Song, Xinliang Wu, Yi Yang, Mingyi Chen, Zhaoyang Lian, Yingying Li, Lei Cheng, and Huajun Chen. 2023b. Teleknowledge pre-training for fault analysis. In *ICDE*, pages 3453–3466. IEEE.

Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone deserves A reward: Learning customized human preferences. *CoRR*, abs/2309.03126.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. KBQA: learning question answering over QA corpora and knowledge bases. *Proc. VLDB Endow.*, 10(5):565–576.

Xuan-Quy Dao, Ngoc-Bich Le, Xuan-Dung Phan, and Bac-Bien Ngo. 2023. An evaluation of chatgpt's proficiency in english language testing of the vietnamese national high school graduation examination. *Available at SSRN 4473369*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *ACL (1)*, pages 320–335. Association for Computational Linguistics.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2023. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *CoRR*, abs/2306.08018.

Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2021. Meaningful answer generation of e-commerce question-answering. *ACM Trans. Inf. Syst.*, 39(2):18:1–18:26.

Shen Gao, Zhaochun Ren, Yihong Eric Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *WSDM*, pages 429–437. ACM.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. *CoRR*, abs/2305.09645.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *ICLR*. OpenReview.net.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781. Association for Computational Linguistics.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese.

Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, and Yuting Liu. 2020. Real-world data medical knowledge graph: construction and applications. *Artif. Intell. Medicine*, 103:101817.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, X Liu, and F Sun. 2022. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multimodal.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: enabling language representation with knowledge graph. In *AAAI*, pages 2901–2908. AAAI Press.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR (Poster)*. OpenReview.net.

Haoran Luo, Haihong E, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, and Wei Lin. 2023. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *CoRR*, abs/2310.08975.

Duc-Vu Nguyen and Quoc-Nam Nguyen. 2023. Evaluating the symbol binding ability of large language models for multiple-choice questions in vietnamese general education. *arXiv preprint arXiv:2310.12059*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *CoRR*, abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *CoRR*, abs/2306.17492.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 2019. Generalizing question answering system with pre-trained language model fine-tuning. In *MRQA@EMNLP*, pages 203–211. Association for Computational Linguistics.

Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. 2020. Multi-modal knowledge graphs for recommender systems. In *CIKM*, pages 1405–1414. ACM.

Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. 2023. Self-influence guided data reweighting for language model pre-training.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V. Chawla, and Panpan Xu. 2023. Graph neural prompting with large language models. *CoRR*, abs/2309.15427.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575. IEEE Computer Society.

Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. 2023a. Huatuo: Tuning llama model with chinese medical knowledge.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023b. Making large language models better reasoners with alignment. *CoRR*, abs/2309.02144.

Peng Wang, Qi Wu, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. 2018. FVQA: fact-based visual question answering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(10):2413–2427.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: knowledge graph attention network for recommendation. In *KDD*, pages 950–958. ACM.

Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. 2023c. Vqa-gnn: Reasoning with multimodal knowledge via graph neural networks for visual question answering.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023d. Aligning large language models with human: A survey. *CoRR*, abs/2307.12966.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Peter West, Ximing Lu, Nouha Dziri, Faeze Brahman, Linjie Li, Jena D. Hwang, Liwei Jiang, Jillian Fisher, Abhilasha Ravichander, Khyathi Chandu, Benjamin Newman, Pang Wei Koh, Allyson Ettinger, and Yejin Choi. 2023. The generative ai paradox: "what it can create, it may not understand".

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *NAACL-HLT*, pages 535–546. Association for Computational Linguistics.

Wonjin Yoon, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong, and Jaewoo Kang. 2019. Pre-trained language model for biomedical question answering. In *PKDD/ECML Workshops (2)*, volume 1168 of *Communications in Computer and Information Science*, pages 727–740. Springer.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: rank responses to align language models with human feedback without tears. *CoRR*, abs/2304.05302.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *ICLR*. OpenReview.net.

Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023a. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In *RecSys*, pages 993–999. ACM.

Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023b. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In *RecSys*, pages 993–999. ACM.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023c. Recommendation as instruction following: A large language model empowered recommendation approach. *CoRR*, abs/2305.07001.

Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. 2023d. FC-KBQA: A fine-to-coarse composition framework for knowledge base question answering. In *ACL (1)*, pages 1002–1017. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023e. Instruction tuning for large language models: A survey. *CoRR*, abs/2308.10792.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with BERT. In *ICLR*. OpenReview.net.

Wen Zhang, Chi Man Wong, Ganqiang Ye, Bo Wen, Wei Zhang, and Huajun Chen. 2021. Billion-scale pre-trained e-commerce product knowledge graph model. In *ICDE*, pages 2476–2487. IEEE.

Yong Zhang, Ming Sheng, Rui Zhou, Ye Wang, Guangjie Han, Han Zhang, Chunxiao Xing, and Jing Dong. 2020b. HKGB: an inclusive, extensible, intelligent, semi-auto-constructed knowledge graph framework for healthcare with clinicians' expertise incorporated. *Inf. Process. Manag.*, 57(6):102324.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *CoRR*, abs/2305.10425.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *CoRR*, abs/2305.13168.

Yushan Zhu, Huaixiao Zhao, Wen Zhang, Ganqiang Ye, Hui Chen, Ningyu Zhang, and Huajun Chen. 2021. Knowledge perceived multi-modal pretraining in e-commerce. In *ACM Multimedia*, pages 2744–2752. ACM.

Appendix

A Related Works

A.1 KG-enhanced Question Answering

Knowledge graphs (KGs) (Wang et al., 2017; Liang et al., 2022) is a kind of complex semantic web that models world knowledge in terms of structural triples as (*head entity, relation, tail entity*). KGs serve as external knowledge source and benefit many AI tasks like language model pre-training (Liu et al., 2020), question answering (Yasunaga et al., 2021; Wang et al., 2023c), and recommendation systems (Wang et al., 2019; Sun et al., 2020). Besides, domain-specific KGs are the important infrastructure of internet industry to provide exact factual knowledge, which is widely leveraged in E-commerce (Zhu et al., 2021; Zhang et al., 2021), telecom fault analysis (Chen et al., 2023b), health care (Li et al., 2020; Zhang et al., 2020b) and so on. It is a popular topic to utilize KGs in real industry applications. In our scenario, we construct a domain-specific KG for cloud service products to benefit our Question Answering (QA) task. QA stands as a cornerstone in NLP, aiming at equipping machines with the capability to autonomously respond to human queries (Su et al., 2019; Yoon et al., 2019). QA tasks can take on various forms. Some require the selection from multiple choices, as seen in certain knowledge base QA (KBQA) (Cui et al., 2017; Tian et al., 2023; Baek et al., 2023) and visual question answering (VQA) (Antol et al., 2015; Chen et al., 2021; Wang et al., 2018). Conversely, tasks like open-domain QA often challenge systems to directly produce textual responses without a set answer pool (Gao et al., 2021; Karpukhin et al., 2020). In the last few years, fine-tuning pre-trained language models has been a leading approach for QA tasks. Models like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) have previously achieved notable performance when adapted with question-answer pairs.

We hold that QA doesn't just remain an academic pursuit; it acts as a bridge, facilitating the adoption of AI technologies in real-world applications. Numerous industrial efforts have been directed toward developing domain-specific QA systems to meet the needs of their users (Gao et al., 2021, 2019). Such systems often rely on domain-specific knowledge bases, like Knowledge Graphs (KGs), to provide relevant information for the posed questions. Our current investigation aligns with this trend, focusing on a domain-specific QA scenario for cloud service products. Moreover, our approach diverges from these recent KG-based QA systems (Jiang et al., 2023b,a; Luo et al., 2023; Zhang et al., 2023d; Chen et al., 2022) that utilize prompts for dialog with (large) language models to facilitate path reasoning and refine the scope of KG retrieval. We propose an innovative knowledgeable preference alignment framework that enhances KG-aware QA with the knowledge preference.

A.2 Large Language Model Application

Prominent large language models (LLMs) like GPT (OpenAI, 2023; Brown et al., 2020; Ouyang et al., 2022) and GLM (Zeng et al., 2023; Du et al., 2022) are sparking a wave of research in the community due to their generalization ability in many NLP tasks such as relation extraction (Zhu et al., 2023), algebraic reasoning (Wei et al., 2022), and question answering (Dao et al., 2023; Nguyen and Nguyen, 2023). Most LLMs leverage the transformer (Vaswani et al., 2017) architecture, benefiting from training on vast corpora (Thakkar et al., 2023) through autoregressive tasks. Deploying and applying LLMs in real-life scenarios is also a major topic in industry today and several efforts have been made. For example, many works (Zhang et al., 2023a; Bao et al., 2023b; Zhang et al., 2023c; Bao et al., 2023a; Zhang et al., 2023b; Chen et al., 2023a) attempt to build recommendation systems with LLMs. Some work like Huatuo (Wang et al., 2023a) and LawyerLlama (Huang et al., 2023) have developed LLMs for domain-specific usage.

Our work proposes a knowledgeable preference alignment framework to incorporate the domain-specific KG into the preference alignment pipeline for the LLM application. By constructing a knowledgeable preference set, the LLMs are trained to align the knowledge preference with humans and select better factual knowledge in the input prompt to solve the QA task.

B Experiment Details

B.1 Baseline Details

(i) Zero-shot approach, which directly prompts the LLM with the input question to get the answer without training.

(ii) In-context learning (Dong et al., 2023) approach, which would sample a few (k-shot) QA pairs as demonstrations from the training dataset as examples and get the answers from the LLM without training.

(iii) Vanilla fine-tuning approach, which fine-tunes the LLM using the QA pairs w/ or w/o retrieved knowledge as Equation 1. The fine-tuning baseline with retrieved knowledge is also known as retrieve-augmented generation (RAG) method.

(iv) Preference alignment approaches, which introduce additional preference alignment objectives during training to align with human preference. We select five existing state-of-the-art (SOTA) PA methods including RRHF (Yuan et al., 2023), SLiC-HF (Zhao et al., 2023), PRO (Song et al., 2023), AFT (both AFT-BC and AFT-DC) (Wang et al., 2023b) as our baselines.

B.2 Evaluation Details

We select three types of metrics to evaluate our method against baselines. The detailed information on the metrics is listed in the following:

(i) Traditional text generation metrics. We select several traditional text generation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015), and METEOR (Banerjee and Lavie, 2005) to evaluate the generated answers. However, these evaluation metrics are mainly used to measure the text-level similarity between generated answers and real answers, which means they can not fully reflect the semantic relevance or depth of understanding of the text.

(ii) Model-based metrics. To evaluate the semantic similarity of the generated answers and the golden answers, we employ several model-based metrics such as BERTScore (Zhang et al., 2020a), perplexity (PPL), and preference score. These metrics evaluate the generated answers using various language models. BERTScore employs BERT (Devlin et al., 2019) to calculate the semantic similarity between two sentences. PPL measures the ability of the LLM to understand and predict the entire sentence. The preference score is $\mathcal{S}$ mentioned in

Equation 3 to reflect the model's preference degree of the current answer.

(iii) Manual evaluation metrics. We employ human labelers to evaluate the results from different methods. The labeler makes a judgment on two answers from unknown sources in a single-blind situation, chooses the better one, and counts the results. The comparison result in each turn is recorded as win/tie/lose.

The three main categories of metrics respond to a certain part of the result's characteristics at three levels: similarity at the textual level, similarity at the semantic level, and human preference.

B.3 Implemention Details

For baselines, we select several different LLMs (ChatGPT (Ouyang et al., 2022), ChatGLM-6B (Zeng et al., 2023), Baichuan-7B [2], Vicuna-7B [3], and Atom-7B-CP) for the zeros-shot approach. For in-context learning, we sample 1,2,4,8-shot QA pairs as demonstrations to support the input question. For the PA methods, we leverage the official code of RRHF (Yuan et al., 2023) and implement other PA methods (SLiC-HF (Zhao et al., 2023), PRO (Song et al., 2023), AFT (Wang et al., 2023b)) based on the code to reproduce the results on our preference dataset. The selection of hyper-parameters is based on the original paper. Atom-7B-CP is employed as the backbone model for all the baseline methods such as in-context learning, vanilla fine-tuning, and PA methods.

[2] https://github.com/baichuan-inc/Baichuan-7B

[3] https://github.com/lm-sys/FastChat