KnowTuning: Knowledge-aware Fine-tuning for Large Language Models

Anonymous ACL submission

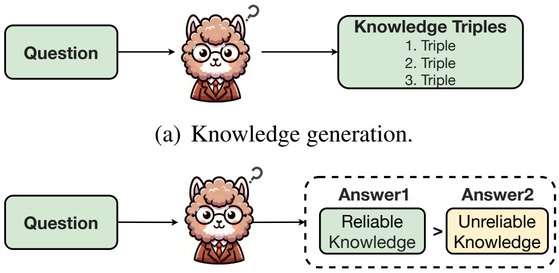
Abstract

Despite their success at many natural language processing (NLP) tasks, large language models (LLMs) still struggle to effectively leverage knowledge for knowledge-intensive tasks, manifesting limitations such as generating incomplete, non-factual, or illogical answers. These limitations stem from inadequate knowledge awareness of LLMs during vanilla fine-tuning. To address these problems, we propose a knowledge-aware fine-tuning (KnowTuning) method to explicitly and implicitly improve the knowledge awareness of LLMs. We devise an explicit knowledge-aware generation stage to train LLMs to explicitly identify knowledge triples in answers. We also propose an implicit knowledge-aware comparison stage to train LLMs to implicitly distinguish between reliable and unreliable knowledge, in three aspects: completeness, factuality, and logicality. Extensive experiments on both generic and medical question answering (QA) datasets confirm the effectiveness of KnowTuning, through automatic and human evaluations, across various sizes of LLMs. Finally, we demonstrate that the improvements of KnowTuning generalize to unseen QA datasets.

1 Introduction

Large language models (LLMs) have become a default solution for many natural language processing (NLP) scenarios, including the question answering (QA) task (Brown et al., 2020; Ouyang et al., 2022; Qin et al., 2023). To achieve strong performance, most LLM first accumulate substantial knowledge by pre-training on extensive datasets (Jiang et al., 2023; Touvron et al., 2023). Then, these LLMs further learn how to exploit the knowledge to answer diverse questions by supervised fine-tuning (SFT) (Wei et al., 2022; Chung et al., 2022; Wang et al., 2023f; Peng et al., 2023; Kang et al., 2023; Wang et al., 2023c).

However, many recent studies indicate that fine-tuned LLMs may struggle to effectively leverage knowledge for question-answering (Yu et al., 2023a; Bai et al., 2023; Chen et al., 2023c; Chang et al., 2023), which aims to answer questions that require in-depth explanations and wide-range domain knowledge. In particular, LLMs are susceptible to generating answers that may be incomplete (Singhal et al., 2022; Bian et al., 2023; Xu et al., 2023b), non-factual (Wang et al., 2023a; Min et al., 2023; Wang et al., 2023b), or illogical (Chen et al., 2023c; Zhong et al., 2023; Kang et al., 2023). Incomplete answers offer incomprehensive and insufficient knowledge, non-factual answers deliver factually incorrect knowledge, and illogical answers provide incoherent and poorly structured knowledge.



(a) Knowledge generation.

(b) Knowledge comparison.

Figure 1: Illustrations of vanilla fine-tuned LLMs lacking knowledge awareness. (a) Vanilla fine-tuned LLMs struggles to identify the necessary knowledge to answer a specific question precisely. (b) Vanilla fine-tuned LLMs cannot effectively distinguish between reliable knowledge and unreliable knowledge in answers.

We hypothesize that these limitations stem from the inadequate knowledge awareness of LLMs during vanilla fine-tuning (Bian et al., 2023; Ji et al., 2023; Dou et al., 2023; Hua et al., 2024). Specifically, as shown in Figure 1, vanilla fine-tuning seldom identifies the necessary knowledge to answer a question. In addition, it usually fails to distinguish between reliable knowledge and unreliable knowledge in answers. Consequently, there is a pressing need for designing knowledge-aware fine-tuning methods. This, then, is the overarching research question that motivates our work: *how can we effectively improve the knowledge awareness of LLMs for solving knowledge-intensive tasks?*

To this end, we propose a novel knowledge-aware fine-tuning method, named KnowTuning, which aims to improve the knowledge awareness of LLMs. KnowTuning consists of two stages: (i) explicit knowledge-aware generation, and (ii) implicit knowledge-aware comparison. In the first stage, we extract knowledge triples from given answers and train LLMs to explicitly generate knowledge triples. In the second stage, we adopt several knowledge-disturbing methods to construct knowledge comparison sets along three dimensions, completeness, factuality, and logicality. Specifically, we generate answers that are worse in terms of completeness, factuality, or logicality, by deleting, revising, and shuffling these knowledge triples. Besides, we rephrase original answers based on the knowledge triples to prevent overfitting. Finally, we combine the rephrased answers and answers with worse completeness, factuality, and logicality as our knowledge comparison sets. We adopt direct preference optimization (DPO) (Rafailov et al., 2023) for optimizing LLMs on our knowledge comparison sets.

We conduct experiments on a generic QA dataset and a medical QA dataset using automatic and human evaluations. Experimental results demonstrate the effectiveness of our proposed method KnowTuning, assessing completeness, factuality, and logicality across various sizes of LLMs. In addition, we demonstrate the improvement that KnowTuning brought can generalize to unseen QA datasets.

In summary, our main contributions are:

- We focus on improving the knowledge awareness of LLMs via fine-tuning for knowledge-intensive tasks.
- We introduce KnowTuning, a novel method that fine-tunes LLMs to leverage explicit knowledge-aware generation and implicit knowledge-aware comparison to improve knowledge awareness of LLMs.
- We demonstrate the effectiveness of KnowTuning in generic and medical domain QA datasets through automatic and human evaluations, across various sizes of LLMs. Furthermore, the improvement of KnowTuning generalizes to unseen QA datasets.

2 Related work

2.1 LLMs for knowledge-intensive Tasks

Large language models (LLMs) have been applied to various knowledge-intensive tasks (Moiseev et al., 2022; Yu et al., 2023b; Khattab et al., 2022; Tian et al., 2023; Zhang et al., 2023a; Xu et al., 2023c; Mishra et al., 2023; Nguyen et al., 2023). Liu et al. (2022b) use few-shot demonstrations to elicit relevant knowledge statements from LLMs for QA tasks. Liu et al. (2022a) train a neural model to generate relevant knowledge through reinforcement learning for QA tasks. Liu et al. (2023) propose a unified model for generating relevant knowledge and solving QA tasks.

However, these approaches mainly focus on multiple-choice QA instead of complex knowledge-intensive QA tasks (Krishna et al., 2021; Kadavath et al., 2022; Liu et al., 2022a, 2023; Kang et al., 2023), which aim to solve questions that require in-depth explanations and wide-range domain knowledge. Recent research indicates that LLMs face challenges in tackling complex knowledge-intensive QA tasks (Yu et al., 2023a; Bai et al., 2023; Chen et al., 2023c; Chang et al., 2023). In particular, they are prone to generating responses that are non-factual (Lee et al., 2022; Sun et al., 2023; Su et al., 2022; Wang et al., 2023b), incomplete (Singhal et al., 2022; Bian et al., 2023), or illogical (Chen et al., 2023c; Zhong et al., 2023; Kang et al., 2023). These limitations stem from the inadequate knowledge awareness of LLMs, hindering their ability to effectively utilize knowledge for solving complex knowledge-intensive QA tasks.

Consequently, there is a need for designing methods to improve the knowledge awareness of LLMs for solving knowledge-intensive tasks.

2.2 Fine-tuning for LLMs

Fine-tuning is a kind of methods to optimize pre-trained LLMs for better understanding and answering to natural language questions (Brown et al., 2020; Ouyang et al., 2022). Previously, fine-tuning is mainly focused on enhancing general-purpose QA abilities of LLMs (Wang et al., 2022; Wei et al., 2022; Longpre et al., 2023). These approaches mainly adopt human-annotated datasets to build the QA dataset. Recently, an alternative strategy involves generating QA datasets through the utilization of advanced LLMs to create answers to a variety of questions (Wang et al., 2023f; Shumailov et al., 2023).

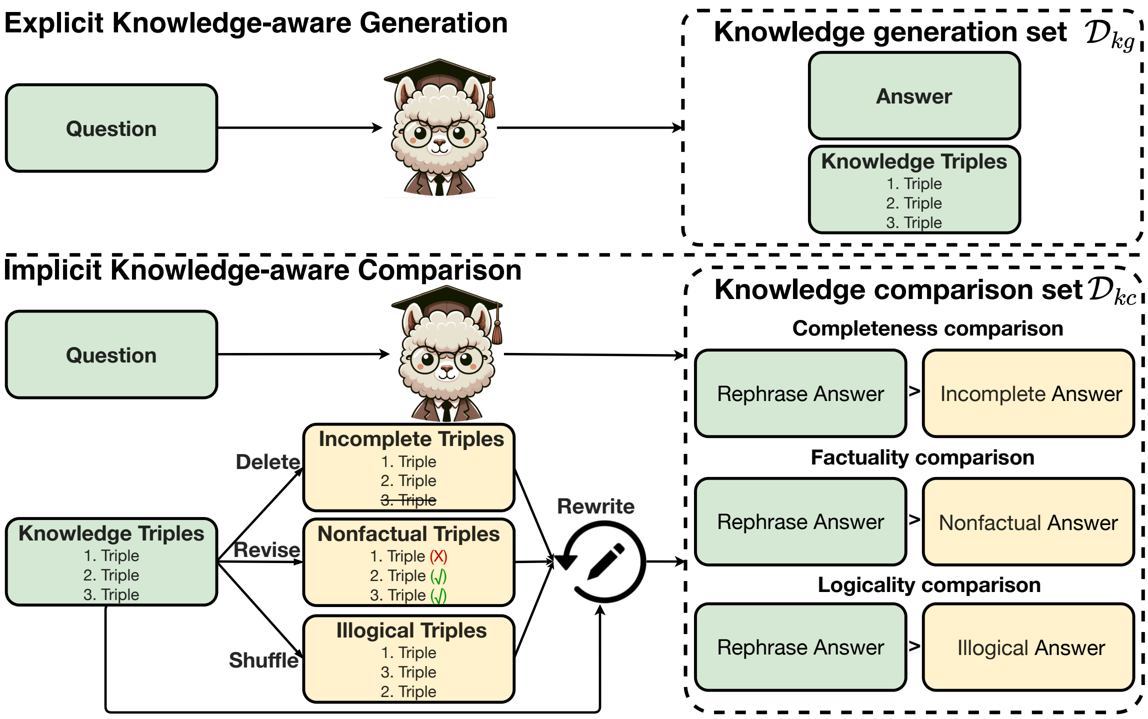


Figure 2: Overview of KnowTuning. KnowTuning leverages explicit knowledge generation and implicit knowledge comparison to improve the knowledge awareness of LLMs.

Recent studies on fine-tuning fuse information about the quality of the generated answers into the supervision signals (Zhao et al., 2023; Guo et al., 2023; Wang et al., 2023d; Dong et al., 2023; Chen et al., 2024). Rafailov et al. (2023) propose direct preference optimization (DPO) to directly optimize LLMs on the pair-wise comparison set. Song et al. (2023) propose Preference Ranking Optimizatio (PRO) to fine-tune LLMs on list-wise comparison sets. Yuan et al. (2023) propose a margin-rank loss to optimize the LLMs on comparison sets.

However, these methods are not designed to improve knowledge awareness of LLMs. In this paper, we aim to leverage explicit knowledge-aware generation and implicit knowledge-aware comparison to improve knowledge awareness of LLMs for solving knowledge-intensive QA tasks.

3 Method

In this section we detail the KnowTuning method. First, we introduce the preliminaries. Then, we introduce the explicit knowledge-aware generation. Next, we introduce implicit knowledge-aware comparison in detail. Finally, a training process for KnowTuning is explained.

3.1 Preliminaries

Supervised fine-tuning. supervised fine-tuning (SFT) aims to train pre-trained LLMs to understand and answer natural language questions. Formally, given a QA dataset $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{N}$, where q_i and a_i denotes a question and a corresponding answer. The training objective of SFT is to minimize the following loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{j=1}^{|a_i|} \log P_{\pi_{SFT}}(a_{i,j}|a_{i,<j}, q_i), \quad (1)$$

where $a_{i,j}$ denotes the j-th token of a_i.

Knowledge triples. Since subject-predicate-object knowledge triples can well cover the necessary knowledge for QA (Yahya et al., 2016; ElSahar et al., 2018; Ouyang et al., 2021), we denote the knowledge in the answer as subject-predicate-object knowledge triples set $\mathcal{K}_i = \{\mathcal{S}_i, \mathcal{P}_i, \mathcal{O}_i\}$, where $\mathcal{S}_i$, $\mathcal{P}_i$ and $\mathcal{O}_i$ refer to subject set, predicate set and object set of answer a_i.

3.2 Explicit Knowledge-aware Generation

To improve the explicit knowledge awareness of LLMs, we fine-tune LLMs to explicitly generate knowledge triples relevant to the question, as illustrated in Figure 2. Specifically, we extract knowledge triples set $\mathcal{K}$ from the original answers a as follows:

$$\mathcal{K}_i = \{\mathcal{S}_i, \mathcal{P}_i, \mathcal{O}_i\} = \text{Extract}(a_i), \quad (2)$$

where $\text{Extract}(\cdot)$ is implemented by prompting OpenAI models to extract knowledge triples, following Bai et al. (2023). Then, we construct the knowledge triples generation dataset $\mathcal{D}_{tk}$ as follows:

$$\mathcal{D}_k = \{q_i, a_i^k\}_{i=1}^N, \tag{3}$$

where a_i^k denotes the text of knowledge triples set $\mathcal{K}_i$. Finally, we combine the original QA dataset $\mathcal{D}$ and the knowledge triples generation dataset $\mathcal{D}_k$ as the explicit knowledge-aware generation dataset $\mathcal{D}_{kg}$ as:

$$\mathcal{D}_{kg} = \mathcal{D} \cup \mathcal{D}_k. \tag{4}$$

3.3 Implicit Knowledge-aware Comparison

To improve implicit knowledge awareness of LLMs in terms of completeness, factuality and logicality, we construct three comparison sets by deleting, revising, and shuffling knowledge triples.

Knowledge completeness comparison. To improve knowledge completeness awareness of LLMs, we construct the knowledge completeness comparison set by randomly deleting the knowledge triples and rewriting the answers. Specifically, we first randomly delete the subject, predicate and object in the knowledge triples set $\mathcal{K}_i$ as follows:

$$\mathcal{K}_i^{ic} = \{\mathcal{S}_i^{ic}, \mathcal{P}_i^{ic}, \mathcal{O}_i^{ic}\}, \tag{5}$$

where $\mathcal{S}_i^{ic}$, $\mathcal{P}_i^{ic}$ and $\mathcal{O}_i^{ic}$ refer to the incomplete sets after randomly deleting α percent of $\mathcal{S}_i$, $\mathcal{P}_i$ and $\mathcal{O}_i$, respectively. Then, we rewrite the answer based on the incomplete knowledge triples set as:

$$a_i^{ic} = \mathrm{Rewrite}(\mathcal{K}_i^{ic}), \tag{6}$$

where $\mathrm{Rewrite}(\cdot)$ is implemented by prompting OpenAI models. In addition, to avoid overfitting on the original answers (Jain et al., 2023), we rephrase the original answers based on knowledge triples.

$$a_i^{rep} = \mathrm{Rewrite}(\mathcal{K}_i). \tag{7}$$

Finally, we combine the rephrase answer a_i^{rep} and the incomplete answer a_i^{ic} into knowledge completeness comparison set as follows:

$$\mathcal{D}_{kcc} = \{(q_i, (a_i^{rep}, a_i^{ic}))\}_{i=1}^N, \tag{8}$$

Knowledge factuality comparison. To improve the knowledge factuality awareness of LLMs, we construct the knowledge factuality comparison set by randomly revising the knowledge triples as nonfactual knowledge triples and rewriting the answers. Specifically, we first randomly revise the knowledge triples set $\mathcal{K}_i$ as follows:

$$\mathcal{K}_i^{nf} = \mathrm{Revise}(\mathcal{K}_i), \tag{9}$$

where $\mathrm{Revise}(\cdot)$ is implemented by prompting OpenAI models to revise the knowledge triples to the wrong knowledge triples. Then, we rewrite the answer based on the nonfactual knowledge triples set as:

$$a_i^{nf} = \mathrm{Rewrite}(\mathcal{K}_i^{nf}). \tag{10}$$

Finally, we combine the rephrased answer a_i^{rep} and the nonfactual answer a_i^{nf} into knowledge factuality comparison set as follows:

$$\mathcal{D}_{kfc} = \{(q_i, (a_i^{rep}, a_i^{nf}))\}_{i=1}^N. \tag{11}$$

Knowledge logicality comparison. To improve the knowledge logicality awareness of LLMs, we construct the knowledge logicality comparison set by randomly shuffling the knowledge triples and rewriting the answers. Specifically, we first randomly shuffle the subject, predicate and object in the knowledge triples set $\mathcal{K}$ as follows:

$$\mathcal{K}_i^{il} = \{\mathcal{S}_i^{il}, \mathcal{P}_i^{il}, \mathcal{O}_i^{il}\}, \tag{12}$$

where $\mathcal{S}_i^{il}$, $\mathcal{P}_i^{il}$ and $\mathcal{O}_i^{il}$ refers to the illogical sets after random shuffling β percent of $\mathcal{S}_i$, $\mathcal{P}_i$ and $\mathcal{O}_i$, respectively. Then, we rewrite the answer based on the illogical knowledge triples set as:

$$a_i^{il} = \mathrm{Rewrite}(\mathcal{K}_i^{il}), \tag{13}$$

We combine the rephrased answer a_i^{rep} and the illogical answer a_i^{il} into knowledge logicality comparison set as follows:

$$\mathcal{D}_{klc} = \{(q_i, (a_i^{rep}, a_i^{il}))\}_{i=1}^N. \tag{14}$$

Finally, we combine the knowledge completeness comparison set, the knowledge factuality comparison set, and the knowledge logicality comparison set as the implicit knowledge-aware comparison set:

$$\mathcal{D}_{kc} = \mathcal{D}_{kcc} \cup \mathcal{D}_{kfc} \cup \mathcal{D}_{klc}. \tag{15}$$

3.4 Training

To improve the knowledge awareness of LLMs for solving complex knowledge-intensive tasks, KnowTuning includes explicit knowledge-aware generation training and implicit knowledge-aware comparison training. Specifically, we first train LLMs on explicit knowledge-aware generation dataset $\mathcal{D}_{kg}$, resulting in a model denoted as π_{kg}. Then, KnowTuning aims to further improve the implicit knowledge awareness of the model π_{kg}

in completeness, factuality, and logicality. To accomplish this, we rewrite the DPO (Rafailov et al., 2023) loss to obtain the implicit knowledge-aware comparison loss as follows:

$$\mathcal{L}_{kc} = \mathbb{E}_{(q,(a_w,a_l))\sim\mathcal{D}_{kc}}\left[\log\sigma\left(\beta\log\frac{\pi_{kc}(a_w|q)}{\pi_{kg}(a_w|q)} - \beta\log\frac{\pi_{kc}(a_l|q)}{\pi_{kg}(a_l|q)}\right)\right], \tag{16}$$

where (a_w, a_l) denotes the answer pair of the question $q \in D_{kc}$, and a_w is the better answer.

4 Experiments

4.1 Research questions

We aim to answer the following research questions in our experiments: **RQ1**: How does KnowTuning perform on generic and medical domain QA under automatic evaluation? **RQ2**: What is the performance of KnowTuning on generic and medical domain QA under human evaluation? **RQ3**: How do explicit knowledge-aware generation and implicit knowledge-aware comparison affect the performance of KnowTuning? **RQ4**: How effective is KnowTuning at generalizing to unseen QA datasets?

4.2 Datasets

We divide the datasets in our experiments into two groups: generic domain and domain-specific. We conduct experiments on generic domain and domain-specific knowledge-intensive question-answering datasets:

- **LIMA** (Zhou et al., 2023) is a carefully curated generic domain QA dataset. The dataset is collected from three community QA websites: Stack Exchange, wikiHow, and the Pushshift Reddit Dataset (Baumgartner et al., 2020). The dataset includes 1000 QA pairs for training and 300 questions for testing.
- **MedQuAD** (Abacha and Demner-Fushman, 2019) is a medical domain QA dataset, which is collected from 12 National Institutes of Health websites. The dataset covers 37 different question types. In this paper, following (August et al., 2022), we filter the questions of the category "Information" for giving definitions and information about medical terms. Specifically, we filter 1000 QA pairs for training and 100 questions for testing.

In addition, to evaluate the ability of methods to generalize to unseen questions, we employed two diverse test sets: Vicuna (Chiang et al., 2023) and WizardLM (Xu et al., 2023a). These test sets totally contain 298 real-world human questions from diverse sources and diverse difficulties.

4.3 Baselines

We compare our model with the following baselines:

- **Base** denotes that testing the Llama2-base model (Touvron et al., 2023) under zero-shot setting.
- **SFT** (Ouyang et al., 2022) represents vanilla fine-tuning backbone LLMs on QA datasets according to Eq. 1.
- **DPO** (Rafailov et al., 2023) fine-tunes LLMs on comparison sets by increasing the likelihood of generating good answers while decreasing the likelihood of bad ones. Following Cui et al. (2023), we first collect candidate answers from different sizes of vanilla fine-tuned LLMs and golden answers, and then use GPT-4 scoring to construct comparison sets with the same size as the knowledge comparison set.

4.4 Evaluation Metrics

We present our experimental results using two evaluation metrics: automatic evaluation and human-based evaluation. Since ROUGE (ROUGE, 2004) and BLEU (Papineni et al., 2002) can not effectively evaluate the quality of answers for complex questions (Krishna et al., 2021; Xu et al., 2023b; Chen et al., 2023a), recent studies propose to use GPT-4 for evaluating the quality of LLMs answers (Zheng et al., 2023; Dubois et al., 2023; Fu et al., 2023). Consequently, we employ GPT-4 to rate generated answers on three aspects: completeness, factuality, and logicality, on a range of 1 to 10. Following Singhal et al. (2022); Zheng et al. (2023); Zhang et al. (2023b), we define completeness, factuality and logicality as: (i) **Completeness**: it examines whether the answers provide comprehensive and sufficient knowledge to the questions. (ii) **Factuality**: it examines whether the knowledge in the answers is factually correct. (iii) **Logicality**: it examines whether the knowledge in the answers is logically rigorous and structured. To avoid positional bias (Ko et al., 2020; Wang et al., 2023e), we evaluate each answer in both positions during two separate runs. Following Li et al. (2023); Chen et al. (2023b), we define "Win-Tie-Lose" as: (i) **Win**: KnowTuning wins twice, or wins once and ties once. (ii) **Tie**: KnowTuning ties twice, or wins

		Completeness			Factuality			Logicality			
Model	Dataset	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Avg. gap
Backbone Language Model: Llama2-7b-base											
KnowTuning vs Base	LIMA	95.00*	3.67	1.33	88.33*	10.34	1.33	92.00*	6.67	1.33	+90.45
KnowTuning vs SFT	LIMA	72.67*	17.66	9.67	48.33*	43.67	8.00	61.33*	29.67	9.00	+51.89
KnowTuning vs DPO	LIMA	68.67*	22.66	8.67	41.00*	51.00	8.00	61.67*	29.66	8.67	+48.67
KnowTuning vs Base	MedQuAD	87.00*	11.00	2.00	70.00*	20.00	10.00	73.00*	20.00	7.00	+70.33
KnowTuning vs SFT	MedQuAD	56.00*	28.00	16.00	49.00*	32.00	19.00	52.00*	30.00	18.00	+34.67
KnowTuning vs DPO	MedQuAD	43.00*	32.00	25.00	48.00*	29.00	23.00	45.00*	34.00	21.00	+22.33
Backbone Language Model: Llama2-13b-base											
KnowTuning vs Base	LIMA	90.67*	8.33	1.00	68.00*	28.00	4.00	74.00*	23.00	3.00	+74.89
KnowTuning vs SFT	LIMA	66.67*	19.67	13.66	48.67*	40.67	10.66	60.67*	29.00	10.33	+47.12
KnowTuning vs DPO	LIMA	60.33*	22.00	17.67	37.00*	49.00	14.00	49.67*	36.67	13.67	+33.89
KnowTuning vs Base	MedQuAD	94.00*	4.00	2.00	70.00*	25.00	5.00	72.00*	23.00	5.00	+74.67
KnowTuning vs SFT	MedQuAD	51.00*	26.00	23.00	37.00*	45.00	18.00	40.00*	46.00	14.00	+24.33
KnowTuning vs DPO	MedQuAD	51.00*	27.00	22.00	35.00*	44.00	21.00	39.00*	44.00	17.00	+21.67

Table 1: Main results on generic QA and medical QA datasets evaluated by GPT-4. The scores marked with $*$ mean KnowTuning outperforms the baseline significantly with p-value< 0.05 (sign. test), following Guan et al. (2021).

	LIMA	MedQuAD
Model	Avg. length	Avg. length
Backbone Language Model: Llama2-7b-base		
Base	377.84	328.43
SFT	387.66	287.88
DPO	405.47	432.15
KnowTuning	426.13	367.21
Backbone Language Model: Llama2-13b-base		
Base	255.01	223.52
SFT	369.96	325.31
DPO	391.12	368.58
KnowTuning	444.57	392.62

Table 2: Average length of generated answers.

once and loses once. (iii) **Lose**: KnowTuning loses twice, or loses once and ties once.

In addition, we employ human judgments as the gold standard for assessing the quality of answers. Specifically, human evaluators perform pair-wise comparisons of the top-performing models identified in automatic evaluations. They are presented with a question and two answers and asked to judge on three aspects: completeness, factuality, and logicality. More details of the evaluation are in Appendix A.

4.5 Implementation details

We employ Llama2-base models of different sizes (7b and 13b) as our backbone models for training. We adopt the Alpaca template (Taori et al., 2023) for training and inference. The OpenAI model used for $\mathrm{Extract}(\cdot)$, $\mathrm{Rewrite}(\cdot)$ and $\mathrm{Revise}(\cdot)$ is *gpt-3.5-turbo-16k*. More details of the implementation are in Appendix B.

5 Experimental results and analysis

To answer our research questions, we conduct generic domain and medical domain QA experiments, ablation studies, and unseen QA experiments. In addition, we conducted a case study to gain further understanding of the effectiveness of KnowTuning.

5.1 Main results (RQ1)

Table 1 presents the GPT-4 evaluation results for both generic and medical domain QA datasets. Across all metrics, KnowTuning outperforms the baseline models in these domains. Based on the results, we have three main observations:

- **KnowTuning consistently surpasses baselines in terms of completeness, factuality and logicality.** Compared with Base and SFT, KnowTuning focuses on explicitly and implicitly improving knowledge awareness of LLMs, which significantly improves the performance of LLMs on knowledge-intensive QA tasks. Compared with DPO, KnowTuning is more effective in improving the performance of LLMs on complex knowledge-intensive QA in multiple aspects. Although DPO improves the performance of vanilla fine-tuned LLMs by distinguishing between generally good and bad answers, it ignores improving the knowledge awareness of LLMs in multiple essential aspects. In contrast, KnowTuning improves knowledge awareness of LLMs in terms of completeness, factuality and logicality, simultaneously. These improvements of KnowTuning are observed across generic and

Model	Dataset	Completeness			Factuality			Logicality			Avg. gap
		Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	
Backbone Language Model: Llama2-7b-base											
KnowTuning vs DPO	LIMA	62.33	27.00	10.67	34.67	58.33	7.00	54.00	37.67	8.33	+41.67
KnowTuning vs DPO	MedQuAD	54.00	19.00	27.00	46.00	36.00	18.00	47.00	36.00	17.00	+28.33
Backbone Language Model: Llama2-13b-base											
KnowTuning vs DPO	LIMA	55.33	28.34	16.33	31.00	58.33	10.67	42.67	45.66	11.67	+30.11
KnowTuning vs DPO	MedQuAD	47.00	31.00	22.00	33.00	55.00	12.00	29.00	63.00	8.00	+22.33

Table 3: Human evaluation results on generic domain and medical domain QA datasets.

Model	Completeness			Factuality			Logicality			Avg. gap
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	
-KG vs KnowTuning	19.33	32.67	48.00	14.67	58.00	27.33	15.67	46.00	38.33	-21.33
-KCC vs KnowTuning	25.67	32.33	42.00	18.67	59.33	22.00	18.00	50.00	32.00	-11.22
-KFC vs KnowTuning	27.67	30.33	42.00	16.33	59.00	24.67	22.33	48.34	29.33	-9.89
-KLC vs KnowTuning	25.33	33.67	41.00	14.00	63.67	22.33	19.33	44.00	36.67	-13.78
-KC vs KnowTuning	14.00	16.67	69.33	12.67	40.66	46.67	13.00	23.33	63.67	-46.67

Table 4: Ablation study evaluated by GPT-4.

medical domain QA datasets, which indicate the importance of improving explicit and implicit knowledge awareness of LLMs.

- **KnowTuning demonstrates effectiveness on LLMs across different sizes.** We observe that KnowTuning consistently improves the performance of QA tasks on different scales (7b and 13B) LLMs. This finding aligns with Bian et al. (2023): LLMs learn a lot of knowledge during the pre-training stage but still need to learn how to effectively leverage knowledge for solving knowledge-intensive QA tasks.
- **Knowtuning tends to generate longer answers with better completeness, factuality, and logicality.** As shown in Table 2, KnowTuning mostly generates longer answers than the baselines and achieves better completeness, factuality and logicality. An exception is observed in the medical QA domain, where DPO based on llama7b-base generates longer answers than KnowTuning. Nonetheless, these answers from DPO are worse in completeness, factuality and logicality. It further demonstrates the importance of improving knowledge awareness of LLMs, as opposed to more surface-level aspects.

5.2 Human evaluation (RQ2)

Human evaluations are crucial for accurately assessing the quality of answers. As shown in Table 3, to facilitate human annotation processes, we focus on comparing KnowTuning with the key baseline DPO:

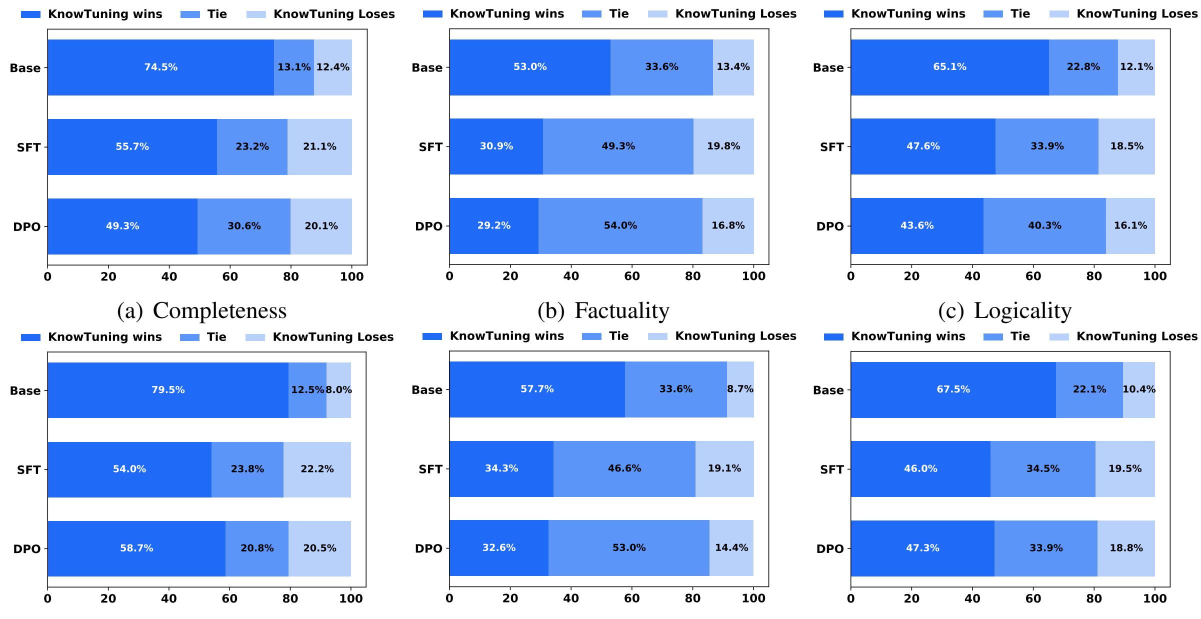
- Our findings indicate that KnowTuning consistently surpasses DPO in terms of completeness, factuality, and logicality performance across various sizes of LLMs under human evaluation.
- KnowTuning demonstrates superior performance over QA in both generic and medical domain QA evaluated by human, in terms of completeness, factuality, and logicality.

5.3 Ablation studies (RQ3)

To analyze the effect of the different knowledge-aware stages in KnowTuning, we conduct an ablation study. Table 4 shows the results on KnowTuning with five settings: (i) **-KG**: KnowTuning without explicit knowledge generation. (ii) **-KCC**: KnowTuning without the implicit knowledge completeness comparison set. (iii) **-KFC**: KnowTuning without the implicit knowledge factuality comparison set. (iv) **-KLC**: KnowTuning without the implicit knowledge logicality comparison set. (v) **-KC**: KnowTuning without any implicit knowledge comparison sets.

Table 4 shows that all knowledge-aware stages help KnowTuning as removing any of them decreases performance:

- **Removing the explicit knowledge-aware generation**. We observe that removing explicit knowledge-aware generation (-KG) decreases the performance of KnowTuning, especially in terms of completeness and logicality. This indicates that explicit knowledge-aware generation helps LLMs to be aware of complete knowledge information and the logical structure of knowledge.
- **Removing the implicit knowledge-aware com-**



(a) Completeness (b) Factuality (c) Logicality

(d) Completeness (e) Factuality (f) Logicality

Figure 3: Results on unseen QA datasets evaluated by GPT-4, including completeness, factuality, and logicality. The backbone model of (a), (b) and (c) is Llama2-7b-base. The backbone model of (d), (e) and (f) is Llama2-13b-base.

parison. We observe that the model without the implicit knowledge-aware comparison faces a huge performance degradation in knowledge-intensive QA. Specifically, removing knowledge completeness comparison (-KCC) negatively impacts completeness, removing knowledge factuality comparison (-KFC) negatively impacts factuality, and removing knowledge logicality comparison (-KLC) negatively impacts logicality. In addition, when removing all implicit knowledge-aware comparison sets (-KC), there is a substantial drop in the performance on the knowledge-intensive QA task on all three aspects. As a result, although the model still explicitly generates knowledge, the absence of distinguishing reliable and unreliable knowledge leads to poor knowledge-intensive QA performance.

5.4 Unseen QA datasets results (RQ4)

To evaluate the ability of methods to generalize to unseen questions, we conduct experiments on LLMs trained on the generic domain QA dataset. Figure 3 demonstrates that KnowTuning can effectively generalize to unseen questions:

- Compared to baselines, KnowTuning can generalize the improvement to unseen questions across different sizes of LLMs.
- We observe that the factuality improvement of KnowTuning is harder to generalize to unseen questions than completeness and logicality. This difficulty arises because factuality requires specific and detailed knowledge that might not be covered during the training phase (Wang et al., 2023b; Xu et al., 2024).

5.5 Case study

We also conduct a detailed case study to intuitively show how KnowTuning improves knowledge awareness of LLMs for solving knowledge-intensive tasks, compared to SFT and DPO. In the case study, KnowTuning answers the question logically in multiple aspects, while SFT and DPO answer with incomplete knowledge and lack of logicality. In addition, SFT and DPO both introduce incorrect knowledge in answers. More details of our case study results are in Appendix C.

6 Conclusions

In this paper, we focus on improving the knowledge awareness of LLMs via fine-tuning for knowledge-intensive tasks. We have proposed KnowTuning to fine-tune LLMs through explicit knowledge-aware generation and implicit knowledge-aware comparison stages. We have conducted comprehensive experiments on generic and medical domain QA datasets, demonstrating the effectiveness of KnowTuning through automatic and human evaluations, across various sizes of LLMs. Moreover, we have shown that the improvements achieved with KnowTuning can generalize to unseen QA datasets. Our code and dataset are available at `https://anonymous.4open.science/r/ACL_KnowTuning-FBA0`.

Limitations

In this study, KnowTuning is mainly aimed at knowledge-intensive tasks, leaving its applicability to other tasks for future research (Burns et al., 2023). Moreover, our efforts have been concentrated on enhancing the knowledge awareness of LLMs during the fine-tuning stage. Future studies will aim to explore improving knowledge awareness of LLMs in the pre-training stage (Rosset et al., 2020).

Ethics Statement

KnowTuning mainly focuses on completeness, factuality, and logicality, but not social bias or the potential for generating harmful or toxic content (Hewitt et al., 2024). It is imperative to exercise caution when implementing our model in real-world applications, particularly in scenarios involving critical decision-making or direct interactions with users.

References

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23.

Tal August, Katharina Reinecke, and Noah A. Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of ACL*, pages 8298–8317.

Yuyang Bai, Shangbin Feng, Vidhisha Balachandran, Zhaoxuan Tan, Shiqi Lou, Tianxing He, and Yulia Tsvetkov. 2023. Kgquiz: Evaluating the generalization of encoded knowledge in large language models. *CoRR*, abs/2310.09725.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of AAAI*, pages 830–839.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, and Ben He. 2023. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. *CoRR*, abs/2303.16421.

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 *Proceedings of NeurIPS*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *CoRR*, abs/2312.09390.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models. *CoRR*, abs/2307.03109.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023a. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings of EMNLP*, pages 6325–6341.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023b. Alpagasus: Training a better Alpaca with fewer data. *CoRR*, abs/2307.08701.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023c. FELM: benchmarking factuality evaluation of large language models. *CoRR*, abs/2310.00741.

Zhipeng Chen, Kun Zhou, Wayne Xin Zhao, Junchen Wan, Fuzheng Zhang, Di Zhang, and Ji-Rong Wen. 2024. Improving large language models via fine-grained reinforcement learning with minimum editing constraint. *CoRR*, abs/2401.06081.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*, abs/2310.01377.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. RAFT: reward ranked finetuning for generative foundation model alignment. *CoRR*, abs/2304.06767.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *CoRR*, abs/2312.09979.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *CoRR*, abs/2305.14387.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of LREC*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as you desire. *CoRR*, abs/2302.04166.

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long text generation by modeling sentence-level and discourse-level coherence. In *Proceedings of ACL*, pages 6379–6393.

Geyang Guo, Ranchi Zhao, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Beyond imitation: Leveraging fine-grained quality signals for alignment. *CoRR*, abs/2311.04072.

John Hewitt, Sarah Chen, Lanruo Lora Xie, Edward Adams, Percy Liang, and Christopher D Manning. 2024. Model editing with canonical examples. *arXiv preprint arXiv:2402.06155*.

Hiyouga. 2023. Llama factory. `https://github.com/hiyouga/LLaMA-Factory`.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Wenyue Hua, Jiang Guo, Mingwen Dong, Henghui Zhu, Patrick Ng, and Zhiguo Wang. 2024. Propagation and pitfalls: Reasoning-based assessment of knowledge editing through counterfactual tasks. *CoRR*, abs/2401.17585.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Neftune: Noisy embeddings improve instruction finetuning. *CoRR*, abs/2310.05914.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating hallucination in large language models via self-reflection. *CoRR*, abs/2310.06271.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2023. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. *CoRR*, abs/2305.18395.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *CoRR*, abs/2212.14024.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of EMNLP*, pages 1109–1121.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of NAACL-HLT*, pages 4940–4957.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. In *Proceedings of NeurIPS*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. *CoRR*, abs/2308.12032.

Jiacheng Liu, Skyler Hallinan, Ximing Lu, Pengfei He, Sean Welleck, Hannaneh Hajishirzi, and Yejin Choi. 2022a. Rainier: Reinforced knowledge introspector for commonsense question answering. In *Proceedings of EMNLP*, pages 8938–8958.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022b. Generated knowledge prompting for commonsense reasoning. In *Proceedings of ACL*, pages 3154–3169.

Jiacheng Liu, Ramakanth Pasunuru, Hannaneh Hajishirzi, Yejin Choi, and Asli Celikyilmaz. 2023. Crystal: Introspective reasoners reinforced with self-feedback. In *Proceedings of EMNLP*, pages 11557–11572.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of ICML*, volume 202, pages 22631–22648.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of EMNLP*, pages 12076–12100.

Aditi Mishra, Sajjadur Rahman, Hannah Kim, Kushan Mitra, and Estevam Hruschka. 2023. Characterizing large language models as rationalizers of knowledge-intensive tasks. *CoRR*, abs/2311.05085.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of NAACL*, pages 1581–1588.

Minh Nguyen, Kishan K. C., Toan Nguyen, Ankit Chadha, and Thuy Vu. 2023. Efficient fine-tuning large language models for knowledge-aware response planning. In *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part II*, volume 14170 of *Lecture Notes in Computer Science*, pages 593–611.

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 *Proceedings of NeurIPS*.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Fact-driven logical reasoning. *CoRR*, abs/2105.10334.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of EMNLP*, pages 1339–1384.

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.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul N. Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. *CoRR*, abs/2007.00655.

Lin CY ROUGE. 2004. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL*, volume 5.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross J. Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *CoRR*, abs/2305.17493.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Schärli, Aakanksha Chowdhery, Philip Andrew Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Gregory S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge. *CoRR*, abs/2212.13138.

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, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! Faithful long form question answering with machine reading. In *Findings of ACL*, pages 744–756.

Weiwei Sun, Zhengliang Shi, Shen Gao, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2023. Contrastive learning reduces hallucination in conversations. In *Proceedings of AAAI*, pages 13618–13626.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *CoRR*, abs/2311.08401.

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. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Cunxiang Wang, Sirui Cheng, Zhikun Xu, Bowen Ding, Yidong Wang, and Yue Zhang. 2023a. Evaluating open question answering evaluation. *CoRR*, abs/2305.12421.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Jiayang Cheng, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. 2023b. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *CoRR*, abs/2310.07521.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023c. Knowledge-driven CoT: Exploring faithful reasoning in LLMs for knowledge-intensive question answering. *CoRR*, abs/2308.13259.

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

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023e. Large language models are not fair evaluators. *CoRR*, abs/2305.17926.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023f. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of ACL*, pages 13484–13508.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of EMNLP*, pages 5085–5109.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of ICLR*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *CoRR*, abs/2304.12244.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023b. A critical evaluation of evaluations for long-form question answering. In *Proceedings of ACL*, pages 3225–3245.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023c. Search-in-the-chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. *CoRR*, abs/2304.14732.

Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *CoRR*, abs/2401.11817.

Mohamed Yahya, Denilson Barbosa, Klaus Berberich, Qiuyue Wang, and Gerhard Weikum. 2016. Relationship queries on extended knowledge graphs. In *Proceedings of WSDM*, pages 605–614.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Nianyi Lin, Kaifeng Yun, Linlu Gong, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Bin Xu, Jie Tang, and Juanzi Li. 2023a. Kola: Carefully benchmarking world knowledge of large language models. *CoRR*, abs/2306.09296.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023b. Generate rather than retrieve: Large language models are strong context generators. In *Proceedings of ICLR*.

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.

Yichi Zhang, Zhuo Chen, Yin Fang, Lei Cheng, Yanxi Lu, Fangming Li, Wen Zhang, and Huajun Chen. 2023a. Knowledgeable preference alignment for llms in domain-specific question answering. *CoRR*, abs/2311.06503.

Yue Zhang, Ming Zhang, Haipeng Yuan, Shichun Liu, Yongyao Shi, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023b. Llmeval: A preliminary study on how to evaluate large language models. *CoRR*, abs/2312.07398.

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? A comparative study on chatgpt and fine-tuned BERT. *CoRR*, abs/2302.10198.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. *CoRR*, abs/2305.11206.

Appendix

A Details of Evaluation

A.1 GPT-4 Evaluation

This section provides specifics of the GPT-4 prompt utilized for evaluation, employing *gpt4-turbo*. Figure 4 illustrates the adapted prompt from Zheng et al. (2023), aimed at assessing the completeness, factuality, and logicality of answers.

A.2 Human Evaluation

Instructions for human evaluation are depicted in Figure 5.

B Details of Implementation

B.1 Prompts for Extracting, Rewriting, and Revising

Details for the prompts used in $\mathrm{Extract}(\cdot)$, $\mathrm{Rewrite}(\cdot)$, and $\mathrm{Revise}(\cdot)$ are provided. Figures 6, 7, and 8 display the prompts for $\mathrm{Extract}(\cdot)$, $\mathrm{Rewrite}(\cdot)$, and $\mathrm{Revise}(\cdot)$, respectively.

B.2 Training

During the training phase, the AdamW optimizer (Loshchilov and Hutter, 2019) is utilized with initial learning rates of $5 \cdot 10^{-5}$ for SFT and $1 \cdot 10^{-5}$ for DPO. The batch sizes for SFT and DPO are set to 16 and 8, respectively, with SFT undergoing 3 epochs of training and DPO 1 epoch. The deletion and shuffling percentages, α and β, are both fixed at 0.5. Training leverages PEFT (Mangrulkar et al., 2022), LLaMA-Factory (Hiyouga, 2023) and LoRA (Hu et al., 2022). All training hyperparameters for SFT and DPO are recommended by LLaMA-Factory (Hiyouga, 2023).

C Details of Case Study

As shown in Figure 9, this case study presents answers provided by three methods: SFT, DPO, and KnowTuning. Generally, the observations are as follows:

- The answer of KnowTuning is the most complete, providing detailed information on ingredients, texture, taste, and how dosa and poori masalas are served differently. The answer of SFT describes only one type of potato masala and does not compare the differences between the two types of potato masala. And the answer of DPO does not describe poori masala comprehensively, making it bad completeness.
- KnowTuning leads in factuality, with specific, accurate details that match traditional recipes. The answer of SFT describes incorporates elements (like grated coconut and carrots) that are not typically found in the most traditional or widely recognized versions of potato masala for dosa. DPO emphasizes coconut milk, which is not a standard ingredient in either dish.
- KnowTuning also excels in logicality, methodically comparing the two masalas in a way that's easy to understand. SFT does not logically address the question, offering a non-comparative, repetitive analysis. DPO encounters problems in maintaining a coherent structure; it does not follow through with a detailed description of poori

[System prompt]
You are a helpful and precise assistant for checking the quality of the answer.

[User prompt]
[Question]
{Q}
[The Start of Assistant 1's response]
{R1}
[The End of Assistant 1's response]
[The Start of Assistant 2's response]
{R2}
[The End of Assistant 2's response]
We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above.
Please rate the Knowledge Completeness, Knowledge Factuality and Knowledge Logicality of their responses. Each aspect of each assistant receives an score on a scale of 1 to 10, where a higher score indicates better performance.
Please generate Knowledge Completeness, Knowledge Factuality and Knowledge Logicality scores for each assistant in order.
Please generate the scores in order and following format.
{'Knowledge Completeness':value,'Knowledge Factuality':value,'Knowledge Logicality':value}
Please first output two lines containing values indicating the Knowledge Completeness, Knowledge Factuality and Knowledge Logicality scores for Assistant 1 and 2, respectively.
In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

Figure 4: Prompts for GPT-4 evaluation.

You'll be presented with a series of questions. For each question, two answers will be provided. Your task is to read both answers carefully and decide which one you believe is better.
When judging, consider:
Completeness: It examines whether the answers provide comprehensive and sufficient knowledge relevant to the questions.
Factuality: It examines whether the knowledge in the answers is factually correct
Logicality: it examines whether the knowledge in the answers is logically rigorous and structured.

Question:
{Q}
Answer A:
{A1}
Answer B:
{A2}

Comparing these two answers, Comparing these two answers, in terms of completeness, factuality and logicality, respectively.
Give the win-tie-lose of Answer A compared to Answer B in each of the three aspects.

Figure 5: Instructions for human evaluation.

[System prompt]
You are an expert in extracting knowledge triples (Subject, Predicate, Object).

[User prompt]
Please follow the following requirements to extract knowledge triples:
1. Please extract all #Knowledge Triples# from the #Given Text#.
2. Please generate answers in JSON format.

```
{
  "Triplets": [
    {
      "Subject": "...",
      "Predicate": "...",
      "Object": "..."
    }
  ]
}
```

3. Please extract the knowledge triples according to the following definition.
Subject: In a triplet, the subject represents the primary entity or concept that the statement is about. It's akin to the focal point of the information being conveyed. The subject is usually a distinct entity identifiable within a given domain of knowledge.

Predicate: The predicate in a triplet functions as the relationship or property that connects the subject to the object. It defines the nature of the link or association between the two. Predicates are often verbs or verb phrases in natural language but can also be relationship terms in a database.

Object: The object in a triplet is the entity or concept that is linked to the subject through the predicate. It can be seen as the target or endpoint of the relationship initiated by the subject. The object can be a concrete entity, a value, or another abstract concept.
#Given Text#:
{A}
#Knowledge Triples#:

Figure 6: Prompts for extracting.

[System prompt]
You are an assistant skilled in writing an fluent answer based on given knowledge triples.

[User prompt]
Please rewrite the #Answer# based on the #Given Knowledge Triples#, and #Answer# should follow the content and logical order in the #Given Knowledge Triples#.

#Given Knowledge Triples#:
{Knowledge Triples}
#Answer#:

Figure 7: Prompts for rewriting.

[System prompt]
You are an expert at revising knowledge triples (Subject, Predicate, Object).

[User prompt]
Please revise the following knowledge triples to be factually incorrect according to the following rules:
1.Revise the knowledge triple to try to have the same word count as the original knowledge triple
#Given Knowledge Triples#:
{Knowledge Triples}
#Revised Knowledge Triples#:

Figure 8: Prompts for revising.

Question:
What is the difference between the potato masala for Dosa and the one that is served with poori?

SFT:
There are two ways of preparing potato masala for dosa. The first one is to make a paste of boiled potatoes, grated coconut, green chilies, and coriander. This paste is then mixed with grated carrots, ginger, garlic, and some salt. This mixture is then fried until the potatoes are cooked.

The second way of preparing potato masala for dosa is to make a paste of boiled potatoes, grated coconut, green chilies, and coriander. This paste is then mixed with grated carrots, ginger, garlic, and some salt. This mixture is then fried until the potatoes are cooked.

DPO:
There are a few key differences between the potato masala for dosa and the one served with poori.

The potato masala for dosa is usually made with less spices than the one served with poori. This is because the dosa is already spicy enough on its own, and the potato masala is meant to complement the spice level of the dosa.
The potato masala for dosa is usually made with more potatoes than the one served with poori. This is because the dosa is a more filling dish than the poori, and the potato masala helps to add more volume to the dish.\n* The potato masala for dosa is usually made with more coconut milk than the one served with poori. This is because the dosa is a creamier dish than the poori, and the coconut milk helps to add more creaminess to the dish.
The potato masala for dosa is usually made with more coriander leaves than the one served with poori. This is because the dosa is a more flavorful dish than the poori, and the coriander leaves help to add more flavor to the dish.
The potato masala for dosa is usually made with more cumin seeds than the one served with poori. This is because the dosa is a more aromatic dish than the poori, and the cumin seeds help to add more aroma to the dish.

Potato Masala for Dosa

The potato masala for dosa is a simple yet flavorful dish that is perfect for a quick and easy meal. The dish is made with potatoes, onions, tomatoes, and a few spices. The potatoes are boiled and then mashed, and then the onions, tomatoes, and spices are added to the mashed potatoes. The dish is then simmered until the potatoes are tender and the flavors have melded together. The potato masala for dosa is a delicious and easy dish that can be made in just a few minutes.

Potato Masala for Poori

The potato masala for poori is a popular dish in Indian cuisine. The dish is made.

KnowTuning:
There are two types of potato masala commonly served in India, one with dosa and the other with poori. Here are the key differences between the two:

1. Ingredients: The dosa potato masala is typically prepared with grated potatoes, onions, tomatoes, coriander, green chili, ginger, garlic, turmeric powder, red chili powder, cumin powder, coriander powder, salt, and water. The poori potato masala, on the other hand, is made with grated potatoes, onions, tomatoes, green chilies, ginger, garlic, red chili powder, coriander powder, salt, and water. While both recipes call for the same ingredients, the dosa masala has a distinct flavor profile due to the addition of turmeric powder and cumin powder.

2. Texture: The dosa masala has a creamy texture, while the poori masala has a chunky texture. The dosa masala is more of a dip or sauce that you can scoop up with the dosa, while the poori masala is a more substantial side dish that goes well with the soft and fluffy texture of the poori.

3. Taste: The dosa masala is typically spicier and more flavorful than the poori masala. The dosa masala is aromatic and has a tangy flavor from the tomatoes and the green chilies. The poori masala, on the other hand, is more subtle in flavor and is slightly sweet due to the addition of turmeric powder.

4. Serving: The dosa masala is usually served as a dip or side dish to be scooped up with the dosa. The poori masala, on the other hand, is served as a side dish to be eaten with the poori.

In conclusion, while both potato masalas are delicious and popular dishes in India, they are different in terms of their ingredients, texture, taste, and serving.

Figure 9: Case study for intuitive comparisons.

masala, which undermines the logicality.