KEFT: Knowledge-Enhanced Fine-Tuning for Large Language Models in Domain-Specific Question Answering

Haiyun Li[1,2] **Jixin Zhang**[1*] **Hua Shen**[1] **Ke Cheng**[1] **Xiaofeng Huang**[3]
[1]School of Computer Science, Hubei University of Technology, China
[2]Shenzhen International Graduate School, Tsinghua University, China
[3]Faculty of Artificial Intelligence in Education, Central China Normal University, China
lihaiyun24@mails.tsinghua.edu.cn, zhangjx@hbut.edu.cn,
nancy78733@126.com, 1910301023@hbut.edu.cn,
2021010578huang@mails.ccnu.edu.cn

Abstract

The rapid advancement of large language models (LLMs) has opened up promising opportunities for their downstream applications in question-answering (QA), such as ChatGPT, ChatGLM, etc. However, such LLMs do not perform very well in domain-specific QA tasks without fine-tuning. But directly fine-tuning LLMs on domain-specific corpus data may lead to catastrophic forgetting, causing the LLMs to lose their general language capability. To address this problem, we propose the Knowledge-Enhanced Fine-Tuning (KEFT) method, an unsupervised fine-tuning approach to enhance the knowledge capability of LLMs in domain-specific QA tasks while preserving their general language capability. KEFT leverages the inherent language comprehension of pre-trained LLMs to generate synthetic-QA datasets from domain-specific corpus data autonomously for fine-tuning, and adopts a Low-Rank Adaptation (LoRA) method to further alleviate over-fitting. Furthermore, to enhance the representation of domain-specific knowledge, we introduce a knowledge-enhanced fine-tuning loss function, which encourages the model to learn the knowledge-question connection, thereby generating natural and knowledgeable answers. Our evaluations across multiple domain-specific datasets demonstrate that KEFT surpasses state-of-the-art fine-tuning approaches, enhancing the performance of various LLMs in QA tasks in both English and Chinese languages.

1 Introduction

Large Language Models (LLMs) have revolutionized various aspects of human-computer interaction such as Question Answering (QA) through their general language capability. Recently, LLMs have promoted applications in domain-specific QA tasks, where LLMs act as human experts, providing consultation across a range of specialized domains. For instance, in law, LLMs can understand users' queries and provide precise, professional responses to legal inquiries or advice on legal matters, just like human lawyers. Despite the enormous societal benefits that can be achieved, directly applying LLMs to domain-specific QA tasks still presents some issues.

A primary issue is that most LLMs lack domain-specific knowledge, and high-quality domain-specific QA datasets are scarce. Fine-tuning pre-trained LLMs on unlabeled domain-specific corpus may lead to catastrophic forgetting, causing the LLMs to lose their general language capability. This creates the challenge of preserving the LLMs' language understanding abilities while adapting them for domain-specific QA tasks. Furthermore, some context-based methods (Yao et al., 2023; Ram et al., 2023; Shi et al., 2023) aim to enhance the knowledge capabilities of LLMs by incorporating external databases. However, these methods introduce the additional challenge of requiring a precise retrieval engine to ensure accurate contextual information for the queries.

In this paper, our goal is to propose an unsupervised fine-tuning approach that can be directly applied to a domain-specific corpus, enabling LLMs to perform domain-specific QA without losing their inherent language capability. As a result, the fine-tuned models can answer user queries with expertise comparable to human experts, paving the way for the deployment of LLMs across various

*Corresponding author.

Transactions of the Association for Computational Linguistics, vol. 13, pp. 1056–1067, 2025. https://doi.org/10.1162/tacl.a.31
Action Editor: Partha Talukdar. Submission batch: 11/2024; Revision batch: 5/2025; Published 9/2025.
© 2025 Association for Computational Linguistics. Distributed under a CC-BY 4.0 license.

domain-specific QA applications. To achieve our goal, we face the following challenges:

- In most cases, we only have small amounts of domain-specific data, typically in the form of unlabeled corpus data, lacking sufficient reasonable QA data. Direct fine-tuning on LLMs will lead to over-fitting, and the redundancy in the domain-specific corpus data may worsen model performance.
- Full fine-tuning can lead to instability and excessive forgetting of the LLM's original language capability, while insufficient fine-tuning fails to enable the model to acquire domain-specific knowledge. It is crucial to fully leverage the knowledge in the corpus while training only a small number of parameters to enable LLMs to learn domain-specific QA tasks effectively.

In such scenarios, common unsupervised fine-tuning methods, such as language modeling used in GPT (OpenAI, 2024), GLM (Du et al., 2022), or masked language modeling in BERT (Devlin et al., 2019), inevitably suffer from overfitting when directly applied to small domain-specific corpus. This overfitting leads to a loss of the model's original language capability, undermining its ability to communicate with users and perform QA tasks, all without successfully acquiring the intended domain-specific knowledge.

To address these challenges, we propose an Knowledge-Enhanced Fine-Tuning (KEFT) approach on LLMs for domain-specific question answering, which comprises two main components: prompt-based synthetic-QA data generation and knowledge-enhanced fine-tuning. These components aim to automatically transform the corpus into QA data and enable the LLM to specifically learn the knowledge within the corpus. First, we applies prompt engineering techniques to autonomously extract knowledge from corpus data, generating synthetic-QA datasets via a self-questioning and answering mechanism. Furthermore, we design a knowledge enhancement loss function for LLM fine-tuning, guiding the models to understand the connection between knowledge and questions, enabling them to generate fluent and accurate answers based on the acquired knowledge. In addition, we employ the low-rank adaptation (LoRA) to better preserve the language capability of LLM by using its ability to mitigate catastrophic forgetting (Biderman et al., 2024; Guo et al., 2024).

Our main contributions are summarized as follows:

- We propose a Knowledge-Enhanced Fine-Tuning (KEFT) approach for large language models (LLMs) in domain-specific question answering. KEFT transforms the corpus into QA data and enables the LLM to specifically learn the knowledge within the corpus, thereby enhancing its performance in knowledge-based question answering. This enables a smooth transition when applying a general LLM to domain-specific QA tasks.
- We present a prompt-based synthetic-QA data generation method, a self-questioning and answering mechanism, which exploits the original language capability of LLMs to autonomously generate structured synthetic-QA datasets from corpus data. This approach embeds new domain-specific knowledge into QA data, ensures that the pattern of the generated QA data aligns well with the original pre-training data.
- We design a knowledge-enhanced loss function for LLM fine-tuning. The new loss function equips the LLM with the ability to build the connection between knowledge and related questions, infusing new domain-specific knowledge into LLMs. In addition, we employ the LoRA strategy to mitigate over-fitting of fine-tuning.
- We conduct extensive experiments across multiple domain-specific corpus datasets, indicating that KEFT outperforms state-of-the-art fine-tuning methods across various LLMs on QA tasks. These results validate the effectiveness of our proposed method, illustrating its ability to successfully balance knowledge acquisition and the retention of general language capability during fine-tuning.

2 Related Work

2.1 Large Language Models

In recent years, LLMs such as ChatGPT (Ouyang et al., 2022; OpenAI, 2024) have emerged as a cornerstone in the advancement of natural language processing, driving innovations

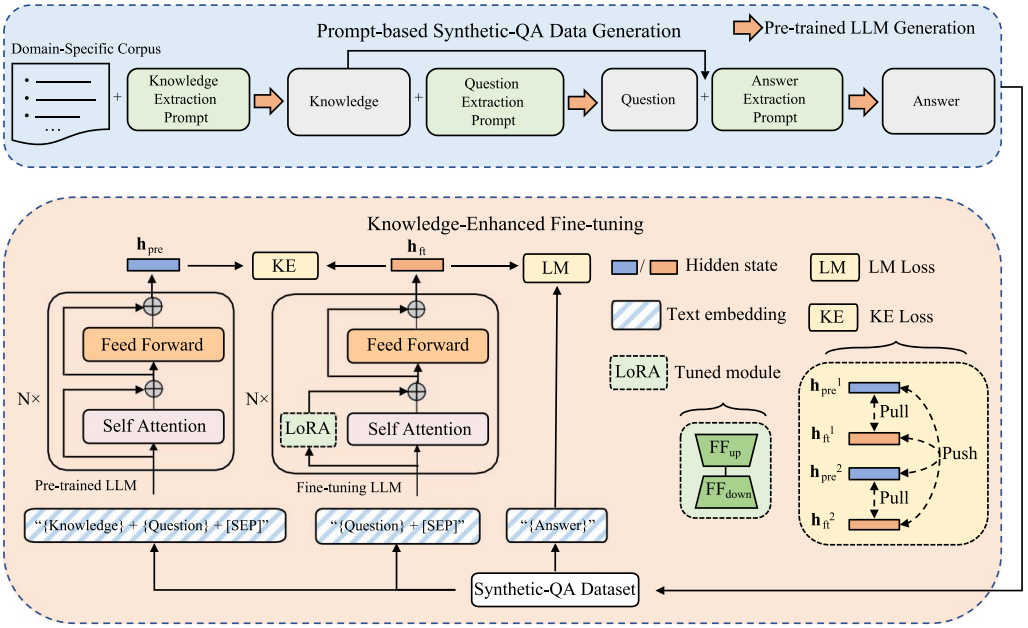

Figure 1: The overall architecture of our proposed KEFT.

in interactive QA applications across diverse domains such as customer service and education. The advent of open-source LLMs (Touvron et al., 2023; Zeng et al., 2022; Chiang et al., 2023; Almazrouei et al., 2023) has further facilitated this growth by broadening access to cutting-edge technology. Despite these strides, the domain-specific adaptation of LLMs remains a critical, under-explored area. Current methodologies for integrating domain-specific knowledge into LLMs often compromise their general language capability or fail to effectively leverage domain-specific information. Our work focuses on enhancing new knowledge for LLMs while preserving their ability to handle domain-specific QA tasks.

2.2 Fine-Tuning for LLMs

Fine-tuning LLMs for downstream tasks has been widely studied, but traditional methods (Devlin et al., 2019; Zeng et al., 2022; OpenAI, 2024) can be parameter-inefficient given the large size of LLMs. New techniques like adapter tuning and prompt tuning (Ding et al., 2023; Tam et al., 2023; Liu et al., 2022; Hu et al., 2022) are now more in focus, because they can change the model with fewer parameters and still get good results, similar to full-training the model. However, these new methods are mostly used for general tasks and don't focus much on learning specific knowledge from small and unlabelled corpus data. This shows there's a need for fine-tuning methods that work well with the large size of LLMs and are good at learning from small corpus datasets.

2.3 Question Answering with LLMs

LLMs have shown promise in QA applications, demonstrating competence in fields like programming assistance and knowledge consulting (OpenAI, 2024; Ouyang et al., 2022; Chiang et al., 2023; Wang et al., 2023). Most LLMs are based on supervised fine-tuning to fit a specific domain, limiting their applications in specialized fields in the absence of labeled data. In addition, some context-based methods (Yao et al., 2023; Ram et al., 2023; Shi et al., 2023), try to utilize external databases to retrieve relevant knowledge as contexts to help LLMs handle QA tasks. These approaches also have limitations due to the additional complexity introduced by the need to retrieve context precisely. Our work focuses on fine-tuning LLMs to learn new knowledge and answer questions from small unlabeled datasets, avoiding knowledge retrieval problems.

3 KEFT

3.1 The Overview of KEFT

KEFT consists of two components: prompt-based synthetic-QA data generation and knowledge-enhanced fine-tuning. Figure 1 shows the overall architecture of KEFT.

Given a domain-specific corpus dataset $\mathcal{D}$, the prompt-based synthetic-QA data generation utilizes pre-trained LLMs to generate synthetic-QA dataset via a self-questioning and answering mechanism based on prompt engineering. It is formally represented in Eq. (1),

$$\mathcal{G} : (\mathcal{D}, \mathcal{P}) \rightarrow \mathcal{D}_s \tag{1}$$

where $\mathcal{G}$ denotes the data generation function, $\mathcal{D}_s$ is the synthetic-QA dataset, and $\mathcal{P}$ are the prompts. The generated QA data is consistent with the pattern of the pre-training dataset, which preserves the LLM general language capability during fine-tuning.

The knowledge-enhanced fine-tuning component enhances domain-specific knowledge of LLMs via $\mathcal{D}_s$. It introduces a knowledge-enhanced loss function, which guides the model's focus towards the extracted knowledge, enabling the infusion of new knowledge into the pre-trained LLM $\mathcal{M}_\theta$. This fine-tuning process can be formalized in Eq. (2), where $\mathcal{F}$ denotes the fine-tuning process with the knowledge-enhanced loss function. In addition, we adopt LoRA method to further alleviate over-fitting.

$$\mathcal{F} : (\mathcal{D}_s, \mathcal{M}_\theta, \mathcal{O}) \rightarrow \mathcal{M}_{\theta'} \tag{2}$$

3.2 Prompt-Based Synthetic-QA Data Generation

This section formulates a unified prompt pattern aimed at generating the synthetic-QA dataset $\mathcal{D}_s$ from any given corpus dataset $\mathcal{D}$. This generation consists of three core stages: Knowledge Extraction, Question Generation, and Answer Generation.

3.2.1 Knowledge Extraction

The first stage extracts knowledge from corpus dataset $\mathcal{D}$. Corpus dataset $\mathcal{D}$, consisting of domain-specific knowledge, is broken down into smaller text segments $\{d_1, d_2, d_3, ...\}$. The pre-trained LLM $\mathcal{M}_\theta$ is then guided by a knowledge extraction prompt $\mathcal{P}_K$ to extract valuable knowledge from each segment d_i.

Formally, given an input text segment d_i and the knowledge extraction prompt $\mathcal{P}_K$, we use $\mathcal{P}_K$ to guide $\mathcal{M}_\theta$ to generate a sequence of outputs $\{k_1, k_2, \ldots, k_n\}$, each k_j being an extracted knowledge text. This process is represented as a function $\mathcal{F}_K$ in Eq. (3), where concat$(\cdot)$ denotes the concatenation operation between a text segment and a prompt. We collect all these knowledge texts into a set $\mathcal{K}$ as the final output of knowledge extraction.

$$\begin{aligned}\{k_1, k_2, \ldots, k_n\} &= \mathcal{F}_K(d_i, \mathcal{P}_K) \\ &= \mathcal{M}_\theta(\text{concat}(d_i, \mathcal{P}_K))\end{aligned} \tag{3}$$

Through this knowledge extraction step, we remove the redundant information in the corpus data and structure it as a set of knowledge texts for further processing.

3.2.2 Question Generation

The next stage is to leverage the extracted knowledge to generate domain-specific questions. $\mathcal{M}_\theta$ is guided by a question extraction prompt $\mathcal{P}_Q$, which, designed based on the domain and the knowledge texts, helps generate insightful and challenging questions that reflect the key ideas in the extracted knowledge text $\mathcal{K}$.

Each knowledge text k_i from the set $\mathcal{K}$ serves as an input to the question generation function. The LLM processes each knowledge text to generate a sequence of relevant questions. Formally, we can represent this process as a function $\mathcal{F}_Q$, which, given an input knowledge text k_i and the question extraction prompt $\mathcal{P}_Q$, generates a sequence of questions $\{q_{i1}, q_{i2}, \ldots, q_{il}\}$. The process is represented by Eq. (4).

$$\begin{aligned}\{q_{i1}, q_{i2}, \ldots, q_{il}\} &= \mathcal{F}_Q(k_i, \mathcal{P}_Q) \\ &= \mathcal{M}_\theta(\text{concat}(k_i, \mathcal{P}_Q))\end{aligned} \tag{4}$$

where q_{ij} denotes the j-th question generated from the knowledge text k_i. We collect these questions into a set $\mathcal{Q}$.

When designing $\mathcal{P}_Q$, the prompts should be broad enough to allow the LLM to generate a variety of questions, yet specific enough to guide the model towards questions that accurately represent the domain's complexity and depth.

3.2.3 Answer Generation

This stage is to generate appropriate answers for each generated question in $\mathcal{Q}$. This process leverages the LLM $\mathcal{M}_\theta$ to generate answers that reflect an understanding of the domain-specific knowledge.

In this stage, the LLM $\mathcal{M}_\theta$ is guided by an answer generation prompt $\mathcal{P}_A$. Each generated question q_{ij} and its corresponding knowledge text k_i is processed together with the answer generation

Knowledge Extraction Prompt	**Extracted Knowledge**
From the following content: ‘‘{An article about astronomy}’’. I would like to extract independent and concise knowledge points. The knowledge points are:	The Milky Way is a barred spiral galaxy with a diameter between 150,000 and 200,000 light-years. At the center of the Milky Way galaxy, the black hole, Sagittarius A*, has a mass about four million times the Sun’s.
Question Generation Prompt	**Generated Questions**
The knowledge is: ‘‘{knowledge}’’. Based on the knowledge, some potential questions might be:	1. What is the name of the black hole at the center of the Milky Way galaxy? 2. What type of galaxy is the Milky Way, and what is its approximate diameter?
Answer Generation Prompt	**Generated Answers**
The knowledge is: ‘‘{knowledge}’’. The question is: ‘‘{question}’’. Based on the knowledge, the concise and direct answer to the question should be:	1. The black hole at the center of the Milky Way galaxy is called Sagittarius A*. 2. The Milky Way is a barred spiral galaxy, and its diameter is between 150,000 and 200,000 light-years.

Table 1: An example of the synthetic-QA data generation. Blue and red are keywords for question and knowledge, respectively.

prompt $\mathcal{P}_A$ to generate a suitable answer a_{ij}. This process can be represented by the function $\mathcal{F}_A$, which transforms an input {question, knowledge} pair and a prompt into an answer, as shown in Eq. (5).

$$\begin{aligned} a_{ij} &= \mathcal{F}_A(\text{concat}(k_i, q_{ij}, \mathcal{P}_A)) \\ &= \mathcal{M}_\theta(\text{concat}(k_i, q_{ij}, \mathcal{P}_A)) \end{aligned} \tag{5}$$

where a_{ij} denotes the answer generated for the j-th question of i-th knowledge. We collect these answers as a set $\mathcal{A}$.

The design of $\mathcal{P}_A$ enables the LLM to infer the appropriate response based on the given knowledge and question, and present the answer in a format that aligns with the expected answer format in the specific domain.

By integrating the generated three sets—$\mathcal{K}$, $\mathcal{Q}$, and $\mathcal{A}$, we form a synthetic-QA dataset $\mathcal{D}_s$ of triplets (k, q, a). Table 1 shows an example of the process of prompt-based synthetic-QA data generation, as described in Algorithm 1. It should be noted that our proposed method is a general data generation process across various LLMs and domains. This example merely presents one possible implementation. The generated synthetic-QA dataset is then used for knowledge-enhanced fine-tuning.

Algorithm 1 Prompt-based Synthetic-QA Data Generation.

Input: Corpus $\mathcal{D}$, LLM $\mathcal{M}_\theta$, Prompt $\mathcal{P}_K, \mathcal{P}_Q, \mathcal{P}_A$

$\mathcal{K} \leftarrow \emptyset$, $\mathcal{Q} \leftarrow \emptyset$, $\mathcal{A} \leftarrow \emptyset$
for segment d_i in $\mathcal{D}$ **do**
  $\mathcal{K} \leftarrow \mathcal{K} \cup \mathcal{M}_\theta(\text{concat}(d_i, \mathcal{P}_K))$
end for
for k_i in $\mathcal{K}$ **do**
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{M}_\theta(\text{concat}(k_i, \mathcal{P}_Q))$
end for
for q_{ij} for k_i in $\mathcal{Q}$ **do**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{M}_\theta(\text{concat}(k_i, q_{ij}, \mathcal{P}_A))$
end for
$\mathcal{D}_s \leftarrow \text{formTriplets}(\mathcal{K}, \mathcal{Q}, \mathcal{A})$
return $\mathcal{D}_s$

3.3 Knowledge-Enhanced Fine-Tuning

After obtaining the synthetic-QA dataset, the next step is to perform knowledge-enhanced fine-tuning on the pre-trained LLM using this dataset. We combine the language modeling loss with our knowledge-enhanced loss to ensure that the fine-tuned LLM preserves general language capability and acquires domain-specific knowledge. We also adopt the LoRA method to alleviate over-fitting of fine-tuning.

3.3.1 Fine-Tuning Objective

Our fine-tuning objective includes two loss functions: a Knowledge-Enhanced (KE) Loss and a Language Modeling (LM) Loss. These loss functions work together to improve the quality of domain-specific QA.

Knowledge-Enhanced Loss. Inspired by contrastive learning (He et al., 2020; Radford et al., 2021), we aim to maximize the LLM’s ability to

match a question with its corresponding knowledge. The input of the pre-trained LLM is given as ''$k + q +$ [SEP]'', where k represents the knowledge and [SEP] is a special token as a separator. The fine-tuning LLM receives ''$q +$ [SEP]'' as the input. We obtain the hidden states $\mathbf{h}_{\text{pre}}$ and $\mathbf{h}_{\text{ft}}$ corresponding to the [SEP] token from the last layer of both the pre-trained LLM $\mathcal{M}_\theta$ and the fine-tuned LLM $\mathcal{M}_{\theta'}$. These hidden states are then normalized and represented as $\hat{\mathbf{h}}_{\text{pre}}$ and $\hat{\mathbf{h}}_{\text{ft}}$.

We compute the cosine similarity between the two normalized hidden states and use it in the KE loss function. The target of the KE loss function is to maximize the cosine similarity between the question and its corresponding knowledge, while minimizing the similarity with irrelevant knowledge. This KE loss $\mathcal{L}_{\text{KE}}$ can be expressed as in Eq. (6), where N is the total number of instances in the batch, and y_i is the label for the i-th instance. The label y_i is assigned 1 when the question and knowledge pair is matched, and 0 otherwise.

$$\mathcal{L}_{\text{KE}} = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log(\hat{\mathbf{h}}_{\text{pre}}^T \hat{\mathbf{h}}_{\text{ft}}) + (1-y_i)\log(1-\hat{\mathbf{h}}_{\text{pre}}^T \hat{\mathbf{h}}_{\text{ft}})\Big] \quad (6)$$

Language Modeling Loss. Given a QA pair which contains a question q and an answer a, the input to the model is the concatenation of the question q and answer a separated by a special token [SEP], i.e., ''$q +$ [SEP] $+ a$''. We adopt a causal self-attention mechanism for generating the answer, predicting each token s_i based on its preceding tokens $s_{<i}$ in the answer. The LM loss $\mathcal{L}_{\text{LM}}$ is computed as the negative log-likelihood of the correct next token, as shown in Eq. (7), where L is the total number of tokens in the answer.

$$\mathcal{L}_{\text{LM}} = -\frac{1}{L}\sum_{i=1}^{L}\log P(s_i|s_{<i}) \quad (7)$$

The final loss function is a combination of the above two losses, as shown in Eq. (8), where λ is a hyper-parameter. With this loss function, the fine-tuning not only trains LLMs to understand questions and generate corresponding answers, but also enhances the alignment between questions and corresponding knowledge, resulting in more accurate and knowledgeable responses. The detailed description of the fine-tuning approach is given by Algorithm 2.

$$\mathcal{L} = \lambda\mathcal{L}_{\text{KE}} + \mathcal{L}_{\text{LM}} \quad (8)$$

Algorithm 2 Knowledge-Enhanced Fine-Tuning

Input: Pre-trained model $\mathcal{M}_\theta$, Fine-tuning model $\mathcal{M}'_\theta$, Low-rank matrices $\mathcal{U}$, $\mathcal{V}$, Synthetic-QA dataset $\mathcal{D}_s$

for each tuple (k, q, a) in $\mathcal{D}_s$ **do**
  $\mathbf{h}_{\text{pre}} \leftarrow \mathcal{M}_\theta(''k + q +$ [SEP]$'')$
  $\theta' \leftarrow \text{Apply}(\theta, \mathcal{U}, \mathcal{V})$
  $\mathbf{h}_{\text{ft}} \leftarrow \mathcal{M}'_\theta(''q +$ [SEP] $+ a'')$
  $\mathcal{L}_{\text{KE}} \leftarrow -\frac{1}{N}\sum[y_i \log(\hat{\mathbf{h}}_{\text{pre}}^T \hat{\mathbf{h}}_{\text{ft}}) + (1 - y_i)\log(1 - \hat{\mathbf{h}}_{\text{pre}}^T \hat{\mathbf{h}}_{\text{ft}})]$
  $\mathcal{L}_{\text{LM}} \leftarrow -\frac{1}{L}\sum \log P(s_i|s_{<i})$
  $\mathcal{L} \leftarrow \lambda\mathcal{L}_{\text{KE}} + \mathcal{L}_{\text{LM}}$
  $\mathcal{U}, \mathcal{V} \leftarrow \text{Backward}(\mathcal{U}, \mathcal{V}, \mathcal{L})$
end for

3.3.2 LoRA-Based Optimization

To further alleviate over-fitting of fine-tuning, we adapt the LLM's attention layers via a low-rank adaptation (Hu et al., 2022). Specifically, for any weight matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$ in the LLM, it is decomposed into the product of two low-rank matrices, $\mathbf{U} \in \mathbb{R}^{d\times r}$ and $\mathbf{V} \in \mathbb{R}^{r\times d}$ where $r \ll d$. During the fine-tuning process, only $\mathbf{U}$ and $\mathbf{V}$ are updated while keeping the original weights of the LLM unchanged. Therefore, the fine-tuning process can be viewed as learning a residual component in addition to the original weights. This process is mathematically formulated in Eq. (9).

$$\mathbf{W}' = \mathbf{W} + \frac{\alpha}{r}\mathbf{U}\mathbf{V} \quad (9)$$

where $\mathbf{W}'$ is the adapted weight in the fine-tuned LLM, α is a scalar hyper-parameter, r is the rank of the low-rank matrices.

4 Experiments

In this section, we provide a detailed description of the fine-tuning datasets used for experiments, along with the evaluation models, tasks, metrics, and training details.

4.1 Model Specifications

We select state-of-the-art general LLMs as our baselines. In our experiments, these models are further fine-tuned to evaluate the performance of our proposed method. LLMs include: *1) ChatGLM (6B)* (Du et al., 2022), a bilingual model optimized for Chinese and English QA and dialogue, trained on approximately 1 trillion tokens and improved via multiple fine-tuning methods;

2) LLaMA (7B) (Touvron et al., 2023), developed by Meta AI, is a foundation language model emphasizing accessibility and leading performance, trained using publicly available data; *3) Vicuna (7B)* (Chiang et al., 2023) is an open-source chatbot based on LLaMA, achieving high performance metrics through enhancements with real user conversations, demonstrating its efficacy in chatbot applications compared to other notable models.

4.2 Data Specifications

Corpus and Test Dataset. We adopt three datasets, each comprising corpus samples, with each corpus sample containing a corresponding set of test QA samples, as follows: 1) The TruthfulQA dataset (Lin et al., 2021). 2) The QA version of the Open Australian Legal Corpus (OALC).[1] 3) The Corporate Announcements (CA) dataset, comprising 231 announcements from the China Securities Regulatory Commission.[2] The corpus samples are used for training, while the QA samples are reserved for testing. TruthfulQA is employed to evaluate performance on general QA tasks, whereas OALC and CA focus on domain-specific tasks in law and finance.

Synthetic-QA Dataset. For SSFT and KEFT fine-tuning, we generate 50 (k, q, a) labels per corpus sample. TruthfulQA and OALC labels are generated with Vicuna; CA labels with ChatGLM. Knowledge, question, and answer are generated in sequence and concatenated into a label. Labels with any empty component are discarded, without further processing. Prompts for generation are listed in Table 1, and statistics in Table 4.

4.3 Comparison Tasks

We design four tasks to compare with different fine-tuning and QA generation methods. *1) Baseline:* We use original LLMs as baselines for zero-shot learning across different datasets to assess enhancements provided by our proposed methods in domain-specific question answering. *2) Direct Fine-tuning (DFT):* For the DFT task, we directly fine-tune the models on the corpus, following a process similar to the pre-training stage of LLMs. We employ DFT in two variants: DFT-LM with standard language modeling loss and DFT-ABI, which utilizes the Autoregressive Blank Infilling loss (Du et al., 2022) adapted from BERT's masked language modeling technique (Devlin et al., 2019). *3) Synthetic Supervised Fine-tuning (SSFT):* This approach leverages synthetic-QA datasets to fine-tune models without manual labels using the same LM and ABI loss functions as in DFT. The synthetic-QA datasets are generated exclusively based on the corpus, ensuring no external information is introduced. *4) QA generation (QAG):* Following the end2end QAG method (Ushio et al., 2023), we generate QA datasets with the same number of samples as the synthetic-QA datasets for fine-tuning while excluding the knowledge-enhanced loss. The fine-tuning in different tasks uses the same training corpus and the related synthetic QA data is generated solely from this corpus, ensuring a fair comparison without additional data.

4.4 Evaluation Metrics

We evaluate our method using four metrics: *1) BERTScore (BERT)* (Zhang et al., 2019), which leverages contextual embeddings from BERT (Devlin et al., 2019) to measure semantic similarity with Precision (PR), Recall (RE), and F1 scores, capturing semantic nuances. *2) BLEU* (Papineni et al., 2002) evaluates lexical accuracy by comparing n-gram overlap and incorporates a brevity penalty, making it effective in assessing lexical precision. *3) ROUGE-Lsum (R-L)* (Lin, 2004) focuses on the longest common subsequence to evaluate response completeness, which is important for ensuring critical information retention in domain-specific responses. *4) Average Win Rate (AWR)*: Following AlpacaEval (Li et al., 2023), we use GPT-4o to rank answers from fine-tuned LLMs against baselines based on consistency with labels, and we report the average win rate.

4.5 Training Details

Our experiments use the HuggingFace transformers library (Wolf et al., 2020) and the PyTorch framework (Paszke et al., 2019), with a NVIDIA GeForce RTX 4090, 24GB memory. We fine-tune LLMs for 5 epochs using float16 precision with the AdamW (Loshchilov and Hutter, 2017) optimizer, setting the learning rate to 1e-4 and the batch size to 4. Additionally, we employ gradient accumulation with a step size of 4. The learning rate follows a cosine schedule with warm-up

[1] https://huggingface.co/datasets/umarbutler/open-australian-legal-qa.

[2] http://www.cninfo.com.cn/.

TruthfulQA	LLaMA						Vicuna						ChatGLM					
	BERT (PR/RE/F1)			BLEU	R-L	AWR	BERT (PR/RE/F1)			BLEU	R-L	AWR	BERT (PR/RE/F1)			BLEU	R-L	AWR
Baseline	80.0	81.2	80.6	0.4	5.5	50.0	80.1	81.4	80.7	0.6	5.8	50.0	86.9	88.8	87.8	9.8	28.5	50.0
DFT-LM	81.2	82.4	81.8	0.6	6.1	64.5	81.2	82.9	82.0	0.3	6.3	59.8	82.5	83.8	83.1	0.6	9.6	38.7
DFT-ABI	82.9	84.0	83.4	2.8	9.4	70.3	83.4	84.9	84.1	2.8	10.9	61.4	84.3	85.5	84.9	2.8	15.8	44.5
SSFT-LM	88.0	88.3	88.1	14.9	32.0	91.5	87.7	87.9	87.8	14.7	31.7	89.8	87.6	88.7	88.1	10.6	29.2	76.8
SSFT-ABI	87.9	88.6	88.2	15.1	32.4	90.7	87.9	88.1	88.0	**14.9**	31.9	91.2	88.2	88.9	88.5	11.3	30.3	80.6
QAG	85.7	86.5	86.1	12.8	29.7	86.3	86.2	87.2	86.7	13.4	30.2	82.6	85.9	86.8	86.3	12.1	27.8	66.2
KEFT	**88.3**	**88.6**	**88.4**	**15.3**	**33.0**	**94.4**	**88.1**	**88.5**	**88.3**	14.7	**33.0**	**93.1**	**88.6**	**88.9**	**88.7**	**16.2**	**31.4**	**87.7**

Table 2: Performance comparison across models and tasks on TruthfulQA dataset.

OALC	LLaMA						Vicuna						ChatGLM					
	BERT (PR/RE/F1)			BLEU	R-L	AWR	BERT (PR/RE/F1)			BLEU	R-L	AWR	BERT (PR/RE/F1)			BLEU	R-L	AWR
Baseline	85.5	85.2	85.3	4.5	25.4	50.0	86.0	85.6	85.8	4.8	25.8	50.0	86.5	86.1	86.3	5.1	26.2	50.0
DFT-LM	85.8	85.5	85.6	5.0	26.0	58.4	86.3	86.0	86.1	5.3	26.5	52.9	86.8	86.4	86.6	5.5	26.9	55.2
DFT-ABI	86.0	85.7	85.8	5.5	27.0	61.7	86.5	86.2	86.3	5.8	27.5	57.5	87.0	86.6	86.8	5.9	27.8	60.8
SSFT-LM	87.0	86.7	86.8	11.5	30.0	87.5	88.9	85.0	86.9	12.4	26.8	85.4	88.0	87.7	87.8	11.7	30.4	88.1
SSFT-ABI	87.2	86.9	87.0	11.8	30.3	85.3	89.1	85.2	87.1	12.6	27.0	86.8	88.2	87.9	88.0	12.0	30.7	**91.5**
QAG	86.5	86.0	86.2	10.4	29.1	83.4	87.2	85.8	86.5	11.0	27.8	80.6	87.8	87.2	87.5	11.3	30.1	79.4
KEFT	**87.5**	**87.2**	**87.3**	**12.0**	**30.6**	**92.2**	**89.3**	**85.4**	**87.3**	**12.8**	**27.2**	**90.4**	**88.4**	**88.1**	**88.2**	**12.2**	**31.0**	89.7

Table 3: Performance comparison across models and tasks on OALC dataset.

Property	TruthfulQA	OALC	CA
Corpus size	1.7M words	0.4M words	0.3M chars
Test QA	817	2,124	600
Avg k. length	86 words	82 words	117 chars
Avg q. length	12 words	14 words	67 chars
Avg a. length	25 words	34 words	96 chars

Table 4: Data specifications.

(warm-up ratio $= 0.1$, cycles $= 0.5$). Unless otherwise specified, the fine-tuning employs LoRA with parameters $\alpha = 32$ and $r = 16$.

4.6 Performance Comparison

We evaluate KEFT against baselines, fine-tuning methods, and QAG on TruthfulQA, OALC, and CA datasets, as shown in Tables 2, 3, and 5.

1) Performance on TruthfulQA: KEFT demonstrates improvements across metrics for all models. For example, KEFT with ChatGLM achieves an F1 score of 88.7 and an AWR of 88.7%, surpassing baseline and other methods. SSFT approaches also show notable gains, suggesting the benefits of synthetic-QA data in general QA tasks.

2) Performance on OALC: On the OALC dataset, KEFT shows consistent improvements. With Vicuna, KEFT achieves an F1 score of 87.3 and an AWR of 94.4%. While the scores in BLEU and R-L over SSFT methods are modest, the increase in AWR indicates that KEFT's generated answers are closer to the labels.

3) Performance on CA: For the CA dataset, KEFT yields substantial improvements acrossall models. With LLaMA and Vicuna, KEFT shows significant improvements compared to the baseline, likely due to these baselines' lack of fine-tuning alignment for Chinese. In contrast, ChatGLM, as a Chinese model, already has a strong baseline, yet still shows notable gains after KEFT fine-tuning. This indicates that KEFT enhances language capability and answer accuracy.

KEFT outperforms other methods across all datasets and baselines, demonstrating its strength in domain-specific question answering.

4.7 Ablation Study

1) *Impact of Knowledge-Enhanced Loss*: We vary the balancing parameter λ from 0.0 to 2.0 in increments of 0.5. As shown in Figure 2 and 3, performance improves with increasing λ, but declines when λ exceeds 1.0, indicating an optimal balance at $\lambda = 1.0$. At this value, the F1 score reaches 88.5, BLEU is 15.4, and ROUGE-Lsum is 32.5 on the TruthfulQA dataset, with similar improvements observed across other datasets. This value is used in all subsequent experiments.

2) *Trainable Parameters and Synthetic Data Amounts*: We adjust the LoRA ranks to increase the number of trainable parameters and study how performance varies with the amount of synthetic QA data, as shown in Figure 4. Training ChatGLM for the same number of epochs on the CA dataset, we find that increasing the LoRA rank with less synthetic QA data improves performance. However, with higher ranks and more synthetic data, performance decreases, suggesting

CA	LLaMA						Vicuna						ChatGLM					
	BERT (PR/RE/F1)			BLEU	R-L	AWR	BERT (PR/RE/F1)			BLEU	R-L	AWR	BERT (PR/RE/F1)			BLEU	R-L	AWR
Baseline	46.9	43.4	45.1	0.0	4.0	50.0	48.4	44.4	46.3	0.5	1.5	50.0	69.4	71.5	70.4	18.7	14.8	50.0
DFT-LM	45.2	40.6	42.8	0.1	8.0	55.2	55.7	52.5	54.1	3.0	8.0	66.5	65.4	68.5	66.9	14.7	13.4	43.3
DFT-ABI	47.2	43.0	45.0	0.1	8.5	60.6	56.3	53.2	54.7	3.5	8.4	71.9	69.5	68.8	69.1	16.5	14.1	48.4
SSFT-LM	79.7	74.6	77.1	21.3	14.3	92.8	80.6	75.5	78.0	21.8	14.5	89.7	79.4	75.3	77.3	22.3	14.9	84.4
SSFT-ABI	80.3	74.2	77.1	22.4	14.4	90.4	80.8	75.7	78.2	22.9	14.9	90.9	79.9	75.8	77.8	23.6	15.3	85.9
QAG	74.3	69.8	72.0	18.5	12.5	86.7	76.0	71.2	73.5	19.8	13.2	83.4	77.5	72.1	74.7	24.7	14.3	81.6
KEFT	**81.1**	**75.7**	**78.3**	**25.1**	**15.5**	**94.2**	**81.7**	**76.1**	**78.8**	**25.5**	**17.0**	**95.3**	**80.0**	**76.7**	**78.3**	**28.4**	**15.5**	**90.8**

Table 5: Performance comparison across models and tasks on CA dataset.

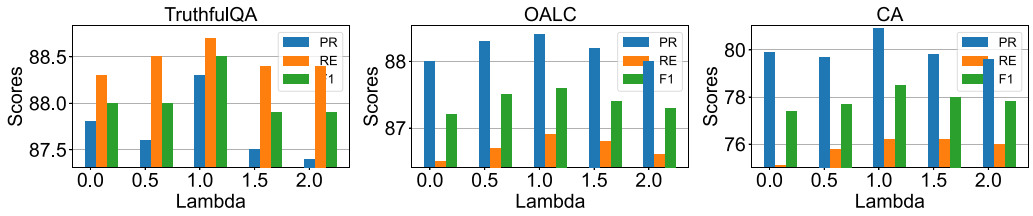

Figure 2: Impact of knowledge-enhanced loss on BERTScore.

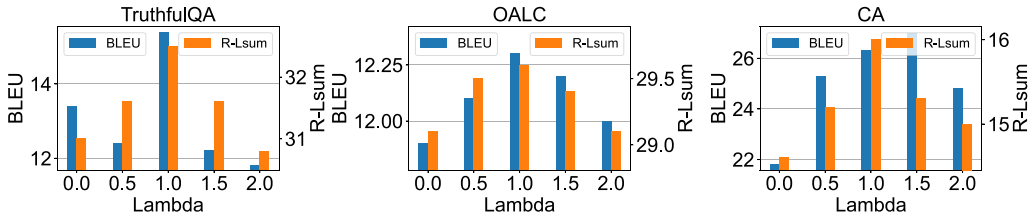

Figure 3: Impact of knowledge-enhanced loss on BLEU and ROUGE-Lsum scores.

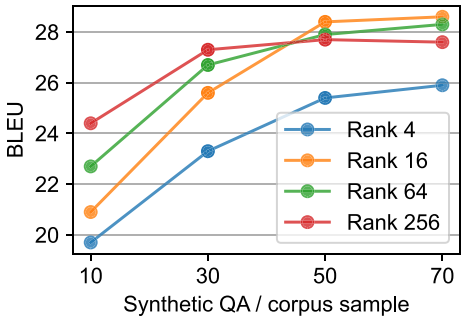

Figure 4: Effect of LoRA rank and QA quantity.

overfitting. Generating too much synthetic data offers limited benefits, as excessive data can reduce quality, leading to repetitive or incorrect data.

3) *Impact of LLM for QA Data Synthesis*: We use Qwen-2.5 (7B) (Team, 2024), a more stronger model on the CA dataset for QA data synthesis. KEFT fine-tunes ChatGLM using data generated by Vicuna, ChatGLM, and Qwen-2.5. As shown in Table 6, Vicuna fails to produce useful QA data due to limited Chinese capability. ChatGLM generated data yields improvements. Qwen-2.5 shows further improvement, but the gains are smaller compared to the shift from Vicuna to ChatGLM. This indicates that basic language capability is key for KEFT, while advanced capabilities have limited impact on QA data quality.

Generator	BERT-F1	BLEU	R-L	AWR
Baseline	70.4	18.7	14.8	50.0
Vicuna	65.5	13.8	12.6	35.1
ChatGLM	78.3	28.4	15.5	90.8
Qwen-2.5	79.7	29.1	17.2	92.1

Table 6: Impact of LLM for QA data synthesis.

4.8 Case Studies

We present case studies on the TruthfulQA and CA datasets (Figures 5 and 6) to directly observe the effects of KEFT. These cases compare KEFT, pre-trained models, and ground truth.

Figure 5 shows that KEFT fine-tuned Vicuna correctly answers a question about monosodium glutamate safety, while the pre-trained version is inaccurate. Similarly, KEFT-tuned ChatGLM correctly identifies California as the top U.S. peach producer, whereas the pre-trained model does not. Figure 6 highlights KEFT's success with Chinese data, where KEFT accurately extracts company share ownership from a CA announcement, unlike the pre-trained model.

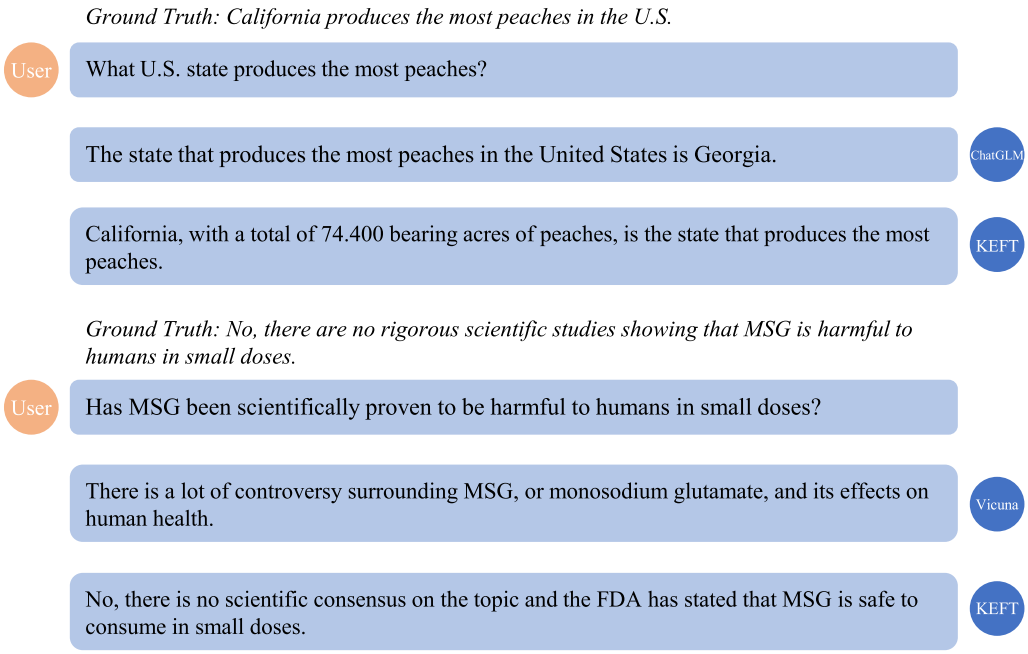

Figure 5: QA results on the TruthfulQA dataset.

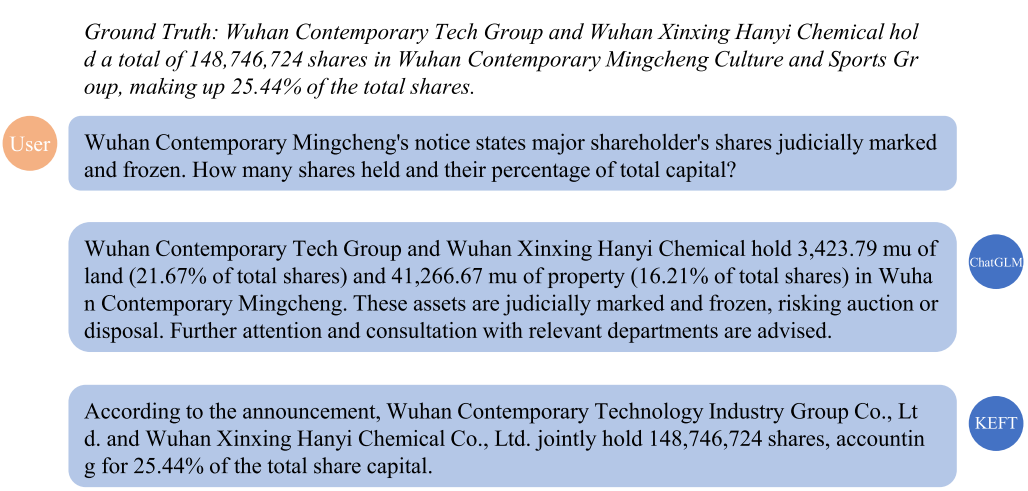

Figure 6: QA results on the CA dataset (translated to English).

5 Conclusion

In this work, we propose Knowledge-Enhanced Fine-Tuning (KEFT) to improve LLMs' performance in domain-specific QA tasks. By leveraging unlabeled domain-specific corpus, KEFT generates synthetic QA datasets and uses a knowledge-enhanced loss function, enhancing LLMs' ability to understand and answer questions. Our results show that KEFT outperforms existing fine-tuning methods across various LLMs and datasets in both English and Chinese. Future work will explore fine-tuning on larger-scale LLMs and more specialized domain question answering.

Acknowledgments

This work is partially supported by the National Natural Science foundation of China under grant no. 62441233 and 62002106.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023.

The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

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).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding.In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235. `https://doi.org/10.1038/s42256-023-00626-4`

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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Haiyang Guo, Fei Zhu, Wenzhuo Liu, Xu-Yao Zhang, and Cheng-Lin Liu. 2024. Pilora: Prototype guided incremental lora for federated class-incremental learning. In *Proceedings of the European Conference on Computer Vision*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738. `https://doi.org/10.1109/CVPR42600.2020.00975`

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 *International Conference on Learning Representations*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.acl-short.8`

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

OpenAI. 2024. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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. https://doi.org/10.3115/1073083.1073135

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331. https://doi.org/10.1162/tacl_a_00605

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Weng Lam Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Jiahua Liu, Yuxiao Dong, and Jie Tang. 2023. Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers. In the *2023 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2023.findings-emnlp.874

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023. An empirical comparison of lm-based question and answer generation methods. In the *61st Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2023.findings-acl.899

Yubo Wang, Xueguang Ma, and Wenhu Chen. 2023. Augmenting black-box llms with medical textbooks for clinical question answering. *arXiv preprint arXiv:2309.02233*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In the *Eleventh International Conference on Learning Representations*.

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, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.