Long-Tailed Question Answering in an Open World

Yi Dai[1*†], **Hao Lang**[2†‡], **Yinhe Zheng**[2], **Fei Huang**[2], **Yongbin Li**[2‡]
[1] Department of Computer Science and Technology, Tsinghua University [2] Alibaba Group
{hao.lang, f.huang, shuide.lyb}@alibaba-inc.com,
dai-y21@mails.tsinghua.edu.cn, zhengyinhe1@163.com

Abstract

Real-world data often have an open long-tailed distribution, and building a unified QA model supporting various tasks is vital for practical QA applications. However, it is non-trivial to extend previous QA approaches since they either require access to seen tasks of adequate samples or do not explicitly model samples from unseen tasks. In this paper, we define Open Long-Tailed QA (OLTQA) as learning from long-tailed distributed data and optimizing performance over seen and unseen QA tasks. We propose an OLTQA model that encourages knowledge sharing between head, tail and unseen tasks, and explicitly mines knowledge from a large pre-trained language model (LM). Specifically, we organize our model through a pool of fine-grained components and dynamically combine these components for an input to facilitate knowledge sharing. A retrieve-then-rerank frame is further introduced to select in-context examples, which guild the LM to generate text that express knowledge for QA tasks. Moreover, a two-stage training approach is introduced to pre-train the framework by knowledge distillation (KD) from the LM and then jointly train the frame and a QA model through an adaptive mutual KD method. On a large-scale OLTQA dataset we curate from 43 existing QA datasets, our model consistently outperforms the state-of-the-art. We release the code and data at `https://github.com/AlibabaResearch/DAMO-ConvAI/tree/main/oltqa`.

1 Introduction

Real-world data often have a long-tailed and open-ended distribution (Liu et al., 2019b). As a cornerstone for AI applications (Yang et al., 2019), Question Answering (QA) is widely investigated to tackle various QA tasks involving diverse formats and domains (Khashabi et al., 2020b; Zhong et al., 2022a). The frequency distribution of QA tasks in our daily life is long-tailed (Reed, 2001), with a few head tasks of adequate samples and many more tail tasks of limited samples, and we continuously encounter new tasks that are not seen during training in an open world.

We formally study *Open Long-Tailed QA* (OLTQA) emerging in natural data settings. A practical QA system shall learn from long-tailed distributed data, i.e., a few head tasks and many tail tasks, and it is expected to perform well over a balanced test set which include head, tail, and unseen tasks.

OLTQA must handle not only few-shot learning for tail tasks in the closed world (Shu et al., 2017), but also zero-shot learning for unseen tasks in an open world (Scheirer et al., 2012) with one unified model. A major challenge for OLTQA is the lack of knowledge required for the language understanding and reasoning abilities of QA tasks, especially under such low resource conditions (Yan et al., 2020). Therefore, it is important for an OLTQA model to share knowledge between head, tail, and unseen QA tasks (Zaremoodi et al., 2018), and mine knowledge from external resources (Liu et al., 2022b).

However, it is non-trivial to directly extend previous methods to the OLTQA setting. Specifically, an effective implementation of knowledge sharing is the multi-task learning (MTL) approach (Liu et al., 2019a; Raffel et al., 2020), in which task-specific components are maintained to preserve learned knowledge (Aghajanyan et al., 2021; Karimi Mahabadi et al., 2021). As we constantly encounter new tasks in practice, it is challenging to directly apply MTL methods since they do not explicitly model samples from unseen tasks.

Another challenge is the absence of samples from unseen tasks in the training process, which leads to poor prior knowledge about unseen tasks. Fortunately, a large pre-trained language model

[*] Work done while the author was interning at Alibaba.
[†] Equal contribution.
[‡] Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6362–6382
July 9-14, 2023 ©2023 Association for Computational Linguistics

(LM) embeds broad-coverage knowledge that can help a variety of tasks (Rubin et al., 2022). One key ingredient in LM knowledge mining is to select demonstrative in-context examples, which guild the LM to generate text that express knowledge for downstream tasks (Liu et al., 2022a). However, few studies have explored selecting in-context examples to directly optimize QA performance in the OLTQA setting.

In this study, we propose an OLTQA model to address challenges mentioned above for the OLTQA setting. Specifically, to encourage knowledge sharing between head and tail tasks while acknowledging the emergence of unseen tasks, we organize our model at the instance-level and use a dynamic architecture for each input (Wiwatcharakoses and Berrar, 2020), i.e., a pool of fine-grained components are maintained and dynamically combined in each forward pass based on the input (Wang et al., 2021). This scheme tackles unseen tasks, since the learned knowledge is distributed into different model components (Trauble et al., 2022).

We further mine knowledge from a large pre-trained LM. Concretely, we employ a retrieve-then-rerank frame (Ren et al., 2021) to select demonstrative in-context examples for a test instance, which guide the LM to decode the output (Brown et al., 2020). The LM outputs are viewed as hints for QA tasks (Zhang and Wan, 2022) and leveraged for improving QA performance. The retrieve-then-rerank frame consists of an efficient retriever and an effective re-ranker (Zamani et al., 2022), which is optimized by a two-stage training approach. The first stage pre-trains the retrieve-then-rerank framework by knowledge distillation from a pre-trained LM (Izacard et al., 2022). The second stage jointly train the above framework and an encoder-decoder QA model through adaptive mutual knowledge distillation (Xie and Du, 2022) to allow information exchange between each other. Our key contributions are summarized as follows:

- We formally define the OLTQA task, which learns from natural long-tail distributed data and optimizes the performance over seen and unseen tasks. We curate a large OLTQA dataset according to a long-tail distribution from 43 existing representative QA datasets.

- We propose an OLTQA model, consisting of knowledge sharing and knowledge mining components to address challenges of OLTQA. An instance-level knowledge sharing mechanism is introduced, and a retrieve-then-rerank frame is employed to mine knowledge from a large pre-trained LM through a novel two-stage knowledge distillation training process.

- Our extensive experimentation on the OLTQA dataset demonstrates that our model consistently outperforms the state-of-the-art.

2 Related Work

Question Answering (QA) is important for advanced AI applications (Yang et al., 2019). Recent approaches try to build unified QA models by casting different QA tasks into a unified text-to-text format (McCann et al., 2019; Khashabi et al., 2020b; Zhong et al., 2022a). Some works try to improve QA performance under the low-resource conditions (Yan et al., 2020; Van et al., 2021; Bai et al., 2022). Some approaches also attempt to solve the open-domain QA problem, aiming at answering general domain questions through an extensive collection of documents (Voorhees et al., 1999; Chen et al., 2017; Singh et al., 2021; Cheng et al., 2021). These approaches do not learn from natural long-tail distributed data.

Long-Tailed Learning focuses on long-tail distributed data (Liu et al., 2019b). Recent approaches for long-tailed learning include re-balancing (Zhang et al., 2021), information augmentation (He et al., 2021), and module improvement (Cui et al., 2021). In this study, we attempt to build a QA model from long-tail distributed data by knowledge sharing and knowledge mining.

Knowledge Mining from external resources is essential for building robust QA models (Pan et al., 2019). Wikipedia and knowledge bases are used to improve QA performance (Bi et al., 2019; Banerjee et al., 2019). Large pre-trained LMs store rich knowledge, which is used to solve various tasks via conditioned generation (Petroni et al., 2019). Recent approaches build prompt retrievers to select in-context examples from a training set to optimize LM generation performance (Rubin et al., 2022). However, these approaches cannot directly optimize our OLTQA model. In this study, we jointly train a retrieve-then-rerank framework and a QA model to enhance QA performance.

Knowledge distillation (KD) is often employed to learn a student model using the knowledge distilled from a teacher model by enforcing the agreement of outputs between the two models (Hinton et al., 2015). Mutual KD helps a group of models mutually generate knowledge to train each other (Zhao and Han, 2021). Our OLTQA model jointly trains the retrieve-then-rerank frame and the QA model through adaptive mutual KD, encouraging them to collaborate with each other (Xie and Du, 2022).

3 Method

3.1 Problem Setup

In this study, we aim to learn from n QA tasks $\{T_1, \cdots, T_n\}$, in which training sets follow a long-tailed Zipf distribution with power value α, i.e., a few head tasks of adequate samples and many tail tasks of limited samples. Each sample of T_i is a tuple of a context $\boldsymbol{c}$, a question $\boldsymbol{q}$, and an answer $\boldsymbol{a}$: $\langle \boldsymbol{c}, \boldsymbol{q}, \boldsymbol{a} \rangle$. Our QA model F is built to predict $\boldsymbol{a}$ based on $\boldsymbol{c}$ and $\boldsymbol{q}$. We also consider a more challenging setting in an open world, i.e., model F needs to predict answers for unseen tasks. Therefore, we collect another $\widetilde{n}$ unseen tasks $\{T_{n+1}, \cdots, T_{n+\widetilde{n}}\}$ that are only used for testing.

3.2 Overview

Our model tackles the open long-tailed QA problem by training a prompt-enhanced encoder-decoder QA model F on long-tailed distributed data. There are mainly two challenges to be addressed: (1) How to alleviate the low-resource problem and share knowledge between head, tail, and unseen tasks; (2) How to mine knowledge from external resources. These two issues are tackled with two key ingredients in our model (see Figure 1): 1. An instance-level knowledge sharing method (Section 3.3); 2. A knowledge mining method from a pre-trained language model (Section 3.4).

We follow previous approaches to serialize the context $\boldsymbol{c}$, question $\boldsymbol{q}$, and answer $\boldsymbol{a}$ into text sequences (Khashabi et al., 2020b; Zhong et al., 2022b). For each training sample $\langle \boldsymbol{c}, \boldsymbol{q}, \boldsymbol{a} \rangle$, we first construct a prompt $\mathcal{P}$ based on $\boldsymbol{c}$ and $\boldsymbol{q}$, and then the encoder takes in the concatenation of $\mathcal{P}$, $\boldsymbol{c}$, and $\boldsymbol{q}$ and the decoder predicts $\boldsymbol{a}$, i.e., $p(\boldsymbol{a}|[\mathcal{P}; \boldsymbol{c}; \boldsymbol{q}])$, where $[;]$ denotes the sequence concatenation operation. Specifically, $\mathcal{P}$ is a concatenation of two kinds of prompts, i.e., a meta prompt $\mathcal{P}_m$ and a knowledge prompt $\mathcal{P}_k$. To capture fine-grained knowledge distributed in each input sample, we maintain s meta prompts $\{\mathcal{P}_m^i\}_{i=1}^s$ and dynamically combine these prompts based on $\boldsymbol{c}$ and $\boldsymbol{q}$ to obtain $\mathcal{P}_m$ (Wang et al., 2021). We associate a key vector $\boldsymbol{k}_m^i$ for each meta prompt $\mathcal{P}_m^i$, respectively. A fixed query function h is built to map $\boldsymbol{c}$ and $\boldsymbol{q}$ to a query vector $\boldsymbol{x} = h(\boldsymbol{c}, \boldsymbol{q})$. h is initialized by a fixed pre-trained LM and not tuned in the training phase. $\mathcal{P}_m$ can be determined by retrieving the most similar key vectors $\boldsymbol{k}_m^i$ using $\boldsymbol{x}$. Note that $\mathcal{P}_m$ is a soft prompt, i.e., a sequence of trainable embeddings that is randomly initialized and optimized when training QA model F (Liu et al., 2021).

We also mine knowledge from a large pre-trained LM g to construct knowledge prompt $\mathcal{P}_k$. Liu et al. (2022a) showed that the efficacy of output generated by an LM could vary widely depending on the choice of in-context examples. In this study, we introduce a retrieve-then-rerank framework $\langle R_1, R_2 \rangle$ (Ren et al., 2021) to select in-context examples from a training set $\mathcal{D}_{tr}$, consisting of a retriever R_1 and a re-ranker R_2 (Zamani et al., 2022). The retriever R_1 is implemented as an efficient dual-encoder (Xiong et al., 2021). The re-ranker R_2 is built as a more effective cross-encoder (Luan et al., 2021). For a test instance $\langle \boldsymbol{c}, \boldsymbol{q} \rangle$, we mine knowledge following three steps: **1.** R_1 retrieves a subset of l candidate examples $\{\boldsymbol{e}_i = \langle \boldsymbol{c}_i, \boldsymbol{q}_i, \boldsymbol{a}_i \rangle\}_{i=1}^l$ from training set $\mathcal{D}_{tr}$; **2.** LM g produces a text $\boldsymbol{h}_i$ for each example $\boldsymbol{e}_i$ by conditional generation $p_g(\boldsymbol{h}_i|[\boldsymbol{e}_i; \boldsymbol{c}; \boldsymbol{q}])$, which can serve as a hint for the test instance; **3.** R_2 further select top $\widetilde{l}$ hints $\{\boldsymbol{h}_i\}_{i=1}^{\widetilde{l}}$ to obtain the knowledge prompt $\mathcal{P}_k$ ($\widetilde{l} \ll l$), in which the scoring function measures the similarity between $\langle \boldsymbol{c}, \boldsymbol{q} \rangle$ and $\langle \boldsymbol{e}_i, \boldsymbol{h}_i \rangle$. Note that $\mathcal{P}_k$ is a hard prompt (Jiang et al., 2020), which is a concatenation of texts in $\{\boldsymbol{h}_i\}_{i=1}^{\widetilde{l}}$.

3.3 Instance-level Knowledge Sharing

To facilitate knowledge sharing between head, tail, and unseen tasks at the instance level, we maintain a pool of prompts and optimize key vectors assigned to these prompts. Specifically, for each input $\langle \boldsymbol{c}, \boldsymbol{q} \rangle$, we select $\widetilde{s}$ prompt keys that are closest to the query vector $\boldsymbol{x} = h(\boldsymbol{c}, \boldsymbol{q})$ and concatenate these $\widetilde{s}$ associated meta prompts to obtain $\mathcal{P}_m$. Intuitively, the knowledge associated with the input sample is distributed in these $\widetilde{s}$ meta prompts.

When learning meta prompt keys, we assume the distribution of these keys should balance diversity and locality. Concretely, meta prompts are expected to distribute to the whole vector space so that every meta prompt can be involved in the training process, while similar prompt keys are grouped into clusters so that the knowledge of each sample

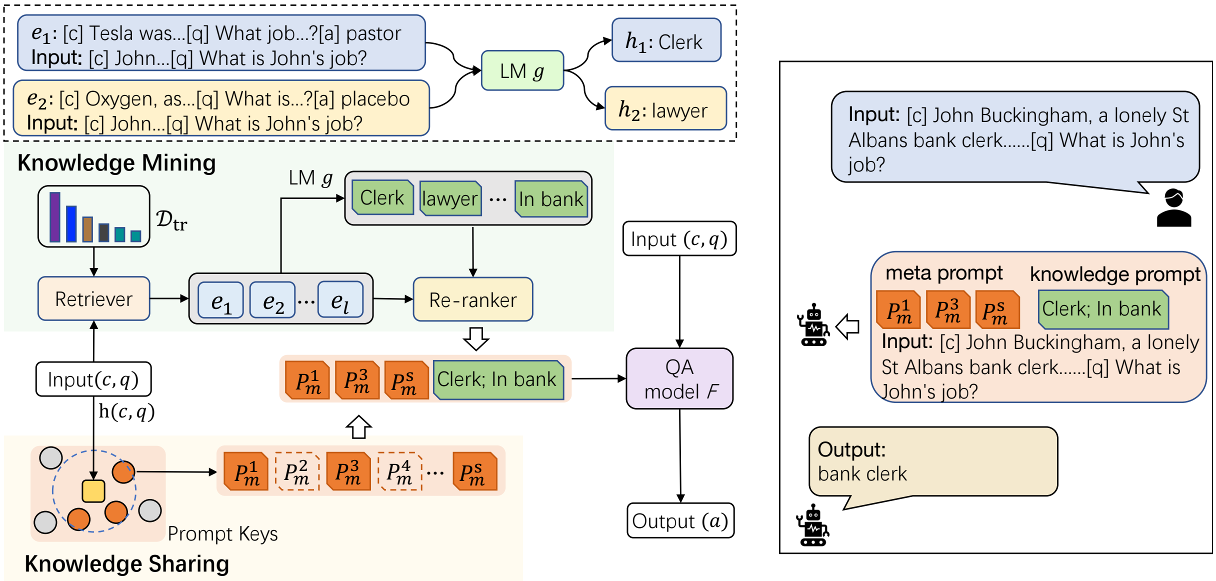

Figure 1: Two key ingredients introduced in our model:(a) Knowledge sharing between head, tail, and unseen tasks at the instance level by maintaining a pool of prompts $\{\mathcal{P}_m^i\}_{i=1}^{s}$; (b) Knowledge mining from a pre-trained LM g using a retrieve-then-rerank framework.

can be better shared. We propose the following loss to enforce the above two properties:

$$\mathcal{L}_m = \mathop{\mathbb{E}}_{\langle \boldsymbol{c},\boldsymbol{q},\boldsymbol{a}\rangle \in \mathcal{D}_{tr}} \Big(\sum_{i \in \mathcal{S}(\boldsymbol{x})} \max(0, ||\boldsymbol{k}_m^i, \boldsymbol{x}|| - \eta) + \sum_{i,j \in \mathcal{S}(\boldsymbol{x})} \max(0, \gamma - ||\boldsymbol{k}_m^i, \boldsymbol{k}_m^j||)/\widetilde{s}^2\Big), \quad (1)$$

where the operator $||\cdot,\cdot||$ determines the distance between two input vectors (here we use cosine distance), $\mathcal{D}_{tr}$ is the training set of all seen tasks, $\mathcal{S}(\boldsymbol{x})$ is the index set of $\widetilde{s}$ selected meta prompt keys that are closest to $\boldsymbol{x}$, η and γ are scalar hyper-parameters to control the distance margin. Specifically, the first term in the above equation pulls these selected meta prompt keys around the query vector. The second term pushes these keys away from each other to occupy the whole vector space.

3.4 Pre-trained LM Knowledge Mining

To further enhance QA performance, we also mine knowledge from a large pre-trained LM g. We employ a retrieve-then-rerank framework $\langle R_1, R_2\rangle$ to retrieve in-context examples from a training set $\mathcal{D}_{tr}$ and further select hints for the test instance that are generated by LM g. We propose a two-stage knowledge distillation method to jointly train the framework $\langle R_1, R_2\rangle$ and QA model F.

Stage I. We pre-train R_1 and R_2 by knowledge distillation from a pre-trained LM g, inspired by Rubin et al. (2022). We first construct a set of c candidate examples $\{\boldsymbol{e}_i = \langle \boldsymbol{c}_i, \boldsymbol{q}_i, \boldsymbol{a}_i\rangle\}_{i=1}^{c}$ for a traning instance $\langle \boldsymbol{c}, \boldsymbol{q}, \boldsymbol{a}\rangle$ with BM25 (Robertson et al., 2009) . Then, we score each candidate example $\boldsymbol{e}_i$ and calculate a distribution of candidate examples by applying the Softmax operator over the resulting scores, based on scoring functions of LM g, R_1, and R_2, respectively. Specifically, the distribution for the LM g scoring function is:

$$p_{lm}(\boldsymbol{e}_k) = \frac{\exp(\log(p_g(\boldsymbol{a}|[\boldsymbol{e}_k; \boldsymbol{c}; \boldsymbol{q}])))}{\sum_{i=1}^{c} \exp(\log(p_g(\boldsymbol{a}|[\boldsymbol{e}_i; \boldsymbol{c}; \boldsymbol{q}])))},$$

where $p_g(\boldsymbol{a}|[\boldsymbol{e}_k; \boldsymbol{c}; \boldsymbol{q}])$ is the score for candidate $\boldsymbol{e}_k$, which is the probability under LM g of output sequence conditioned on the candidate example and the training instance. In a similar manner, we calculate distributions p_{r1} and p_{r2} based on scoring functions of R_1 and R_2, respectively. We optimize R_1 and R_2 by minimizing KL-divergence of p_{lm} from p_{r1} and p_{r2} (Izacard et al., 2022):

$$\mathcal{L}_{lm} = \mathop{\mathbb{E}}_{\langle \boldsymbol{c},\boldsymbol{q},\boldsymbol{a}\rangle \in \mathcal{D}_{tr}} (\mathrm{KL}(\dashv [p_{lm}]||p_{r1}) + \mathrm{KL}(\dashv [p_{lm}]||p_{r2})), \quad (2)$$

where $\dashv [\cdot]$ is a stopgrad operator that sets the gradient of its operand to zero.

Stage II. We jointly train $\langle R_1, R_2\rangle$ and the QA model F. For each training sample $\langle \boldsymbol{c}, \boldsymbol{q}, \boldsymbol{a}\rangle$, we first construct prompt $\mathcal{P}_m$ and $\mathcal{P}_k$, and then optimize the encoder-decoder QA model F together with $\mathcal{P}_m$ using the following loss:

$$\mathcal{L}_f = \mathop{\mathbb{E}}_{\langle \boldsymbol{c},\boldsymbol{q},\boldsymbol{a}\rangle \in \mathcal{D}_{tr}} (-\log\, p_F(\boldsymbol{a}|[\mathcal{P}_m; \mathcal{P}_k; \boldsymbol{c}; \boldsymbol{q}])). \tag{3}$$

To allow information exchange and encourage agreement between $\langle R_1, R_2\rangle$ and QA model F, mutual knowledge distillation is introduced to refine R_1, R_2, and F by knowledge distillation from each other (Zhao and Han, 2021). However, in this case, a worse-performing model is allowed to generate knowledge to train a better-performing model, which may lead to collective failures (Xie and Du, 2022). Therefore, we propose an adaptive mutual knowledge distillation method to allow a model to generate knowledge for training another model only if it performs better.

Therefore, we evaluate the performance of R_1, R_2, and F on a validation set $\mathcal{D}_{val}$ before mutual knowledge distillation. Specifically, we select top $\widetilde{l}$ hints $\{\boldsymbol{h}_i\}_{i=1}^{\widetilde{l}}$ from the c candidate examples $\{\boldsymbol{e}_i\}_{i=1}^{c}$ of a validation instance $\langle \boldsymbol{c}, \boldsymbol{q}, \boldsymbol{a}\rangle$ based on scoring functions of R_1, R_2, F, and then obtain knowledge prompt $\mathcal{P}_k^{r1}$, $\mathcal{P}_k^{r2}$ and $\mathcal{P}_k^{f}$, respectively. The scoring function of QA model F is $p_F(\boldsymbol{a}|[\mathcal{P}_m; \boldsymbol{h}_i; \boldsymbol{c}; \boldsymbol{q}])$, where $\boldsymbol{h}_i$ is a hint for example $\boldsymbol{e}_i$ and acts as a pseudo knowledge prompt. We evaluate R_1, R_2, and F as follows:

$$v_i = \mathop{\mathbb{E}}_{\langle \boldsymbol{c},\boldsymbol{q},\boldsymbol{a}\rangle \in \mathcal{D}_{val}} \log p_F(\boldsymbol{a}|[\mathcal{P}_m; \mathcal{P}_k^i; \boldsymbol{c}; \boldsymbol{q}]), \tag{4}$$

where $i \in \{r1, r2, f\}$ denotes a specific model. Lastly, we optimize the adaptive mutual knowledge distillation loss as follows:

$$\mathcal{L}_{mkd} = \mathop{\mathbb{E}}_{\langle \boldsymbol{c},\boldsymbol{q},\boldsymbol{a}\rangle \in \mathcal{D}_{tr}} \sum_{i,j\in\{r1,r2,f\}} \mathrm{KL}(\dashv [p_i]\|p_j) \cdot \mathbb{I}(v_i > v_j), \tag{5}$$

where p_f is the distribution of candidate examples based on the scoring function of QA model F.

The whole training process of our model is summarized in Algorithm 1.

Algorithm 1: The training process

Input: Training data $\mathcal{D}_{tr}$, validation data $\mathcal{D}_{val}$.
Output: QA model F, meta prompts $\{\mathcal{P}_m^i\}_{i=1}^s$, prompt keys $\{\boldsymbol{k}_m^i\}_{i=1}^s$, framework $\langle R_1, R_2\rangle$.

```
// Stage I
```
1 Train R_1 and R_2 using $\mathcal{L}_{lm}$ (Eq. 2).
```
// Stage II
```
2 Train $\{\boldsymbol{k}_m^i\}_{i=1}^s$ using $\mathcal{L}_m$ (Eq. 1).
3 Train F and $\{\mathcal{P}_m^i\}_{i=1}^s$ using $\mathcal{L}_f$ (Eq. 3).
4 Evaluate R_1, R_2 and F (Eq. 4).
5 Train R_1, R_2, F, $\{\mathcal{P}_m^i\}_{i=1}^s$ using $\mathcal{L}_{mkd}$ (Eq. 5).

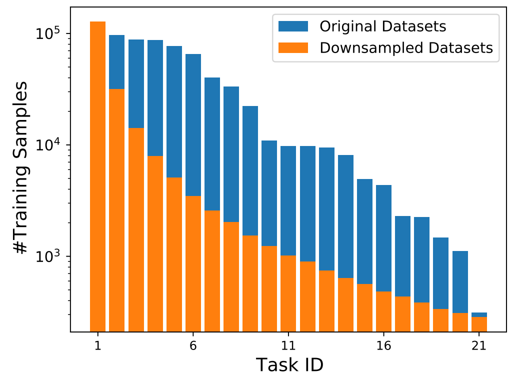

Figure 2: Training dataset statistics of long-tailed QA tasks. Blue bars represent the original dataset sizes of 21 seen tasks and orange bars denote down-sampled dataset sizes.

4 Experiments

4.1 Datasets

We curate an open long-tailed question answering benchmark from 43 existing representative QA datasets (Khashabi et al., 2022) covering four QA formats (*Extractive* QA, *Abstractive* QA, *Multiple-choice* QA, and *Yes/No* QA). See Appendix A for more details of the datasets. We regard each dataset as an individual QA task and reserve $\widetilde{n} = 22$ as unseen tasks. Our model is trained on the rest of $n = 21$ seen tasks while tested on all 43 tasks. We down-sample the training sets of all seen tasks following a Zipf distribution with power value $\alpha = 2.0$ to construct the training data for our model. Figure 2 shows the training data statistics.

4.2 Metrics

The evaluation metric of each above task follows Khashabi et al. (2022) (see more details in Appendix A). We calculate the average performances over 21 seen tasks (A_{seen}) and 22 unseen tasks (A_{unseen}) to evaluate the QA performance. We also calculate the average scores over a subset of seen tasks with m largest training sets (Head@m) and n smallest training sets (Tail@n) to evaluate the performance of head and tail tasks, respectively.

Methods	SQuAD 2	NatQA	RACE	ARC-easy	MCTest	ARC-hard	MultiRC	Head@3	Tail@4	A_{seen}
UnifiedQA	77.80	40.25	56.97	36.84	77.19	31.77	80.45	58.34	56.56	55.21
ProQA	79.84	39.01	59.55	44.21	80.00	38.13	77.56	59.47	59.98	53.23
Muppet	79.41	40.83	57.13	38.07	79.06	31.34	85.57	59.12	58.51	56.13
Hyperformer++	79.52	40.24	58.24	40.18	76.88	31.10	86.86	59.33	58.76	56.81
EPR	44.14	39.50	38.82	51.81	55.00	39.80	56.41	40.82	50.76	47.97
Ours (w/o $\mathcal{P}_m$)	77.72	42.10	58.13	56.49	83.02	39.46	85.58	59.32	66.14	59.60
Ours (w/o $\mathcal{P}_k$)	78.89	40.20	59.34	39.82	76.25	33.11	85.90	59.48	58.77	56.51
Ours	**79.99**	**42.68**	**59.65**	**58.95**	**83.75**	**40.43**	**87.82**	**60.77**	**67.74**	**61.48**

Table 1: Comparison with competitive baselines and ablations on main components of our model in seven seen tasks (3 head tasks + 4 tail tasks). Bold numbers are superior results.

Methods	AdversarialQA dRoberta	RACE-C	MMMLU	OneStopQA Advanced	MCScript	DREAM	PubmedQA	A_{unseen}
UnifiedQA	18.16	49.86	28.77	54.01	67.97	59.56	50.53	46.70
ProQA	14.21	54.91	25.96	61.11	71.23	64.41	58.00	48.27
Muppet	17.33	50.00	30.42	54.79	70.91	58.61	56.73	46.98
Hyperformer++	16.99	52.11	25.26	59.88	71.51	59.31	53.00	47.21
EPR	27.74	35.39	28.77	60.49	65.56	53.92	59.67	46.57
Ours (w/o $\mathcal{P}_m$)	25.16	53.51	33.68	61.11	77.46	68.28	62.07	52.09
Ours (w/o $\mathcal{P}_k$)	17.12	53.23	31.23	56.70	70.80	60.29	56.27	48.37
Ours	**28.05**	**56.88**	**36.14**	**64.31**	**79.16**	**69.51**	**64.40**	**54.42**

Table 2: Comparison with competitive baselines and ablations on main components of our model in seven unseen tasks (randomly selected). Bold numbers are superior results.

4.3 Implementation Details

We use T5-base (Raffel et al., 2020) to initialize the QA model F. For knowledge sharing, we maintain totally $s = 30$ meta prompts, and set the length of each meta prompt to 10. We adopt a fixed T5-base encoder with an average pooling layer to generate the query vector. For each instance, we select $\widetilde{s} = 5$ meta prompts to construct $\mathcal{P}_m$. We set $\eta = 0.15$ and $\gamma = 0.3$ in Eq. 1. For knowledge mining, we use a dual-encoder as retriever, and a cross-encoder as re-ranker. Encoders in the retriever and the re-ranker are all initialized with Bert-base-uncased (Devlin et al., 2019). We use GLM-10B (Du et al., 2022) with 10B parameters as pre-trained LM g. For each instance, the retriever first selects $l = 64$ examples from the training dataset, and the re-ranker selects $\widetilde{l} = 4$ examples to construct $\mathcal{P}_k$. All hyper-parameters are tuned according to the average score on the validation set. All results reported in our paper are averages of 3 runs with different random seeds. We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e-4 and batch size of 32. Our model is trained for five epochs. All experiments are performed on 8 A100 GPUs. See Appendix D for more implementation details.

4.4 Baselines

We use the following competitive baselines: **1. UnifiedQA**: (Khashabi et al., 2020b) casts different QA tasks into a unified text-to-text format and builds a single model for all QA tasks; **2. ProQA**: (Zhong et al., 2022a) uses structural prompts to train a unified QA model with a QA-centric pre-training; **3. Muppet**: (Aghajanyan et al., 2021) maintains task-specific heads and learns QA tasks through multi-task learning; **4. Hyperformer++**: (Karimi Mahabadi et al., 2021) uses a hyper-network to generate task-specific adapters for multi-task learning; **5. EPR**: (Rubin et al., 2022) propose an efficient method to retrieve in-context examples for a test instance and use a pre-trained LM to directly decode the output based on the examples. Note that "Muppet" and "Hyperformer++" have no specific modules for unseen tasks. Thus, we select a task with the lowest perplexity across all seen tasks for an input from unseen tasks in the testing phase, following Madotto et al. (2021).

4.5 Main Results

Table 1 shows the result on seen tasks. Our model outperforms all competitive baselines in terms of Head@3, Tail@4, A_{seen}, and achieves SOTA results on all head and tail tasks. We can observe that: **1.** Our model achieves an even larger performance improvement for tail tasks, i.e., absolute improvement is 1.44 in Head@3 and 8.98 in Tail@4, compared to the best-performing baseline Hyperformer++. The performance gain precisely demonstrates the advantages of knowledge sharing between head and tail tasks and knowledge mining from external resources. **2.** Our model also outperforms the in-context learning baseline EPR without any parameter update of the pre-trained LM. This shows that leveraging knowledge mined from a pre-trained LM and directly optimizing QA tasks can lead to better QA performance. See Appendix B for more evaluation details of all 21 seen tasks.

Table 2 shows the result on unseen tasks. Our model yields the best performances on all metrics. We can also observe that: **1.** Our model that shares knowledge through fine-grained components (i.e., a pool of meta prompts) and mines knowledge from an LM generally obtain higher performance. **2.** EPR is on par with the other baselines trained on seen tasks. It shows that a pre-trained LM embeds a large amount of knowledge, which can help QA tasks potentially.

4.6 Ablation Studies

Model Main Components: Ablation studies are carried out to validate the effectiveness of each main component in our model. Specifically, the following variants are investigated: **1. w/o $\mathcal{P}_m$** removes the knowledge sharing component, i.e., meta prompt $\mathcal{P}_m$ is not used. **2. w/o $\mathcal{P}_k$** removes the knowledge mining component, i.e., knowledge prompt $\mathcal{P}_k$ is not used. Results in Table 1 and Table 2 indicate that our model outperforms all ablation variants. Specifically, we can also observe that: 1. Both knowledge sharing (see w/o $\mathcal{P}_m$) and knowledge mining (see w/o $\mathcal{P}_k$) components help to improve the QA performance. 2. Knowledge mining brings larger improvement compared to knowledge sharing component on both tail and unseen tasks. This further proves the importance of leveraging knowledge embedded in the pre-trained LM for the OLTQA setting. We provide examples where our model is correct and the variant without knowledge mining (i.e., w/o $\mathcal{P}_k$) is incorrect, together with 4 top hints selected by the retrieve-then-rerank framework in Appendix C.

Knowledge Mining Components: To evaluate design choices of retrieve-then-rerank framework $\langle R_1, R_2 \rangle$ and two-stage knowledge distillation (KD) in knowledge mining, we perform ablation on alternatives: **1. BM25 Retriever** uses the unsupervised retriever BM25 (Robertson et al., 2009) to replace retriever R_1. **2. EPR Retriever** trains R_1 by using a pre-trained LM as the scoring function (Rubin et al., 2022). **3. w/o Re-ranker** removes the re-ranker R_2, and directly uses R_1 to select examples and generate hints. **4. w/o MKD** removes the adaptive mutual KD loss $\mathcal{L}_{mkd}$. **5. Static MKD** removes $\mathcal{L}_{mkd}$, and performs mutual KD based on the performance of R_1, R_2, and F evaluated at the very beginning of training stage two. **6. Back KD** removes $\mathcal{L}_{mkd}$, and train R_1 and R_2 using knowledge distilled from F (Izacard et al., 2022).

Categories	Variants	A_{seen}	A_{unseen}
Retriever	BM25 Retriever	58.06	51.44
	EPR Retriever	59.24	52.14
Re-ranker	w/o Re-ranker	58.41	51.01
Knowledge Distillation	w/o MKD	59.82	50.90
	Static MKD	60.09	51.88
	Back KD	60.21	52.35
Ours		**61.48**	**54.42**

Table 3: Ablation on knowledge mining components.

Data	Methods	Tail@16	A_{unseen}
w/o head tasks	w/o $\mathcal{P}_m$	59.00	50.55
	Ours	59.54 (+0.54)	51.05(+0.50)
w/ head tasks	w/o $\mathcal{P}_m$	59.56	52.09
	Ours	61.32 (**+1.76**)	54.42(**+2.33**)

Table 4: Effect of $\mathcal{P}_m$ in different data distributions.

Results in Table 3 show that the knowledge mining approach used in our model performs better than all other variants. We can further observe that: **1.** Retrieving in-context examples using other approaches (i.e., BM25 Retriever and EPR Retriever) degenerates the model performance by a large margin. This shows the effectiveness of the two-stage training of R_1 in our model. **2.** Re-ranking hints generated by an LM help to improve the QA performance (see w/o Re-ranker). **3.** Removing the adaptive mutual KD loss (i.e., w/o MKD) degenerates the QA performance. This proves the effectiveness of information exchange between the two branches of our model. **4.** Variants of $\mathcal{L}_{mkd}$ lead to limited QA performance (see Static MKD and Back KD). This shows the importance of performance-aware for mutual knowledge distillation.

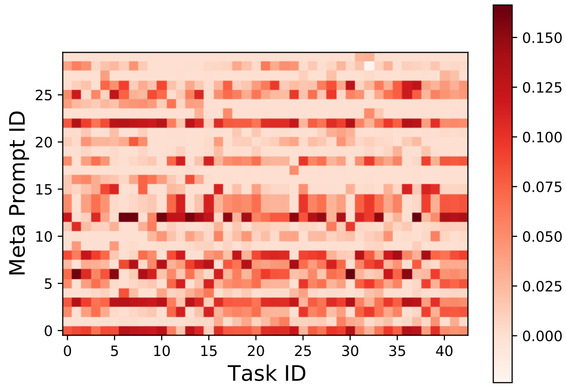

Figure 3: Visualization of $\mathcal{P}_m$ selection mechanism.

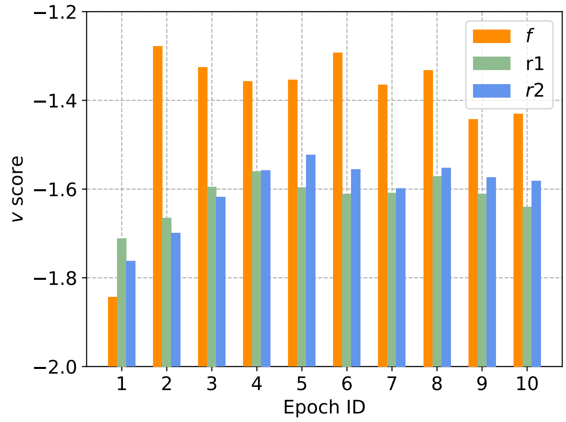

Figure 4: Performance of retriever, re-ranker and the QA model in training stage two.

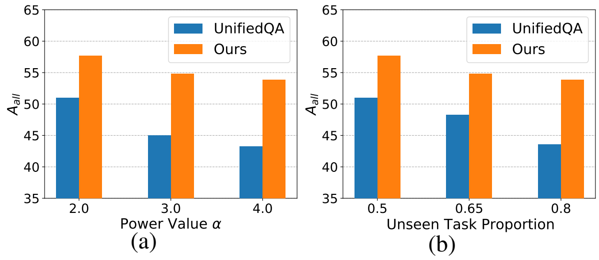

Figure 5: The influence of (a) dataset longtail-ness and (b) proportion of unseen tasks over all 43 tasks.

4.7 Further Analysis

Effect of $\mathcal{P}_m$ in Different Data Distributions We also validate the effectiveness of meta prompt $\mathcal{P}_m$ for knowledge sharing in different data distributions. Specifically, we construct a variant of the training set (and denote it as "w/o head") by discarding samples from head tasks, which consist of samples from 16 tail tasks. We also denote the original training set as "w/ head". The performance of our model on these two datasets is tested with and without $\mathcal{P}_m$.

Results in Table 4 show that our model benefits more from $\mathcal{P}_m$ with samples from head tasks. This further validates our claim that meta prompt $\mathcal{P}_m$ helps to facilitate knowledge sharing between head, tail, and unseen tasks.

Analysis on $\mathcal{P}_m$ Selection Mechanism We plot the heat map of meta prompt $\mathcal{P}_m$ selection frequency for each task in Figure 3. We can observe that: **1.** Some hot meta prompts are shared by most tasks, which probably encode common knowledge for question answering. **2.** Other meta prompts are shared by a few tasks, which might contain task-specific knowledge.

Analysis on Adaptive Mutual KD We visualize the performance of R_1, R_2, and QA model F on the validation set $\mathcal{D}_{val}$ which are evaluated (Eq. 4) at the beginning of each epoch during training stage two in Figure 4. We can observe that: **1.** Initially, R_1 and R_2 are allowed to generate knowledge for training F because they are pre-trained in training stage one. After epoch one, F performs better than R_1 and R_2, and starts to teach student model R_1 and R_2 as a teacher model. **2.** During training, R_2 gradually outperforms R_1. Overall, the relative performance of R_1, R_2, and QA model F compared to each other is not stable during training. Thus, to avoid collective failures, being aware of individual performance is essential to perform mutual knowledge distillation.

Influence of Dataset Longtail-ness The longtail-ness of the dataset (i.e., the degree of imbalance of task distribution in training) could have an impact on the model performance. Figure 5(a) shows that as the dataset becomes more imbalanced (i.e., α of Zipf distribution increases), our model only undergoes a moderate performance drop compared to UnifiedQA. Here, the performance is evaluated on a test set from all 43 tasks.

Influence of Proportion of Unseen Tasks The performance change w.r.t. proportion of unseen tasks is shown in Figure 5(b). Compared to UnifiedQA, the performance of our model changes steadily as the proportion of unseen tasks rises. The knowledge sharing and knowledge mining components of our model enhance robustness to unseen tasks.

5 Conclusion

We introduce the open long-tailed QA (OLTQA) task that learns from natural long-tail distributed data and optimizes the performance over seen and unseen tasks. We propose an OLTQA model to address the challenges of OLTQA. An instance-level knowledge sharing mechanism is introduced, and a retrieve-then-rerank frame is employed to mine knowledge from a large pre-trained LM through a two-stage knowledge distillation training process. We validate our model on a curated OLTQA benchmark. Our publicly available data would enable future research that is directly transferable to real-world applications.

Limitations

We identify the major limitation of this work is its input modality. Specifically, our model only considers textual inputs, ignoring question answering tasks in vision and audio. A multi-modal question answering model under realistic open long-tailed scenario is worth further exploration. Fortunately, through multi-modal pre-training models (Xu et al., 2021; Huo et al., 2021) and question answering methods (Kim et al., 2020), we can equip our model with multi-modal question answering ability. For future work, learning multi-modal question answering in an open (including out of distribution data (Lang et al., 2022, 2023a,b)) long-tailed scenario still remains a challenge, and we will continue to work on it.

Ethics Statement

This work does not raise any direct ethical issues. In the proposed work, we seek to develop a method for long-tailed question answering in an open world, and we believe this work can benefit the field of question answering, with the potential to benefit other fields involving open long-tailed problem. All experiments are conducted on open datasets.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. Prost: Physical reasoning about objects through space and time. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4597–4608.

Ziwei Bai, Baoxun Wang, Zongsheng Wang, Caixia Yuan, and Xiaojie Wang. 2022. Domain adaptive multi-task transformer for low-resource machine reading comprehension. *Neurocomputing*, 509:46–55.

Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, Florence, Italy. Association for Computational Linguistics.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Yevgeni Berzak, Jonathan Malmaud, and Roger Levy. 2020. STARC: Structured annotations for reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5726–5735, Online. Association for Computational Linguistics.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2521–2530, Hong Kong, China. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, 05, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. UnitedQA: A hybrid approach for open domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3080–3090, Online. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *ArXiv*, abs/1803.05457.

Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. 2021. Parametric contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 715–724.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

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.

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, Dublin, Ireland. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. 2021. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 235–244.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Yuqi Huo, Manli Zhang, Guangzhen Liu, Haoyu Lu, Yizhao Gao, Guoxing Yang, Jingyuan Wen, Heng Zhang, Baogui Xu, Weihao Zheng, Zongzheng Xi, Yueqian Yang, Anwen Hu, Jinming Zhao, Ruichen Li, Yida Zhao, Liang Zhang, Yuqing Song, Xin Hong, Wanqing Cui, Dan Yang Hou, Yingyan Li, Junyi Li, Peiyu Liu, Zheng Gong, Chuhao Jin, Yuchong Sun, Shizhe Chen, Zhiwu Lu, Zhicheng Dou, Qin Jin, Yanyan Lan, Wayne Xin Zhao, Ruihua Song, and Ji-Rong Wen. 2021. Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *CoRR*, abs/2103.06561.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane A. Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *ArXiv*, abs/2208.03299.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020a. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online. Association for Computational Linguistics.

Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. 2022. Unifiedqa-v2: Stronger generalization via broader cross-format training.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020b. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D. Yoo. 2020. Modality shifting attention network for multi-modal video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Hao Lang, Yinhe Zheng, Binyuan Hui, Fei Huang, and Yongbin Li. 2023a. Out-of-domain intent detection considering multi-turn dialogue contexts. *arXiv preprint arXiv:2305.03237*.

Hao Lang, Yinhe Zheng, Yixuan Li, Jian Sun, Fei Huang, and Yongbin Li. 2023b. A survey on out-of-distribution detection in nlp. *arXiv preprint arXiv:2305.03236*.

Hao Lang, Yinhe Zheng, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022. Estimating soft labels for out-of-domain intent detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 261–276, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yichan Liang, Jianheng Li, and Jian Yin. 2019. A new multi-choice reading comprehension dataset for curriculum learning. In *Asian Conference on Machine Learning*, pages 742–757. PMLR.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

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 the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019b. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. The natural language decathlon: Multitask learning as question answering.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. MCScript2.0: A machine comprehension corpus focused on script events and participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving question answering with external knowledge. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 27–37, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

William J Reed. 2001. The pareto, zipf and other power laws. *Economics letters*, 74(1):15–19.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8722–8731.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2012. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772.

Lei Shu, Hu Xu, and Bing Liu. 2017. Doc: Deep open classification of text documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2911–2916.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Advances in Neural Information Processing Systems*, volume 34, pages 25968–25981. Curran Associates, Inc.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Frederik Trauble, Anirudh Goyal, Nasim Rahaman, Michael Curtis Mozer, Kenji Kawaguchi, Yoshua Bengio, and Bernhard Scholkopf. 2022. Discrete key-value bottleneck. *ArXiv*, abs/2207.11240.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Hoang Van, Vikas Yadav, and Mihai Surdeanu. 2021. Cheap and good? simple and effective data augmentation for low resource machine reading. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2116–2120.

David Vilares and Carlos Gómez-Rodríguez. 2019. HEAD-QA: A healthcare dataset for complex reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 960–966, Florence, Italy. Association for Computational Linguistics.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. 2021. Learning to prompt for continual learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149.

Chayut Wiwatcharakoses and Daniel P. Berrar. 2020. Soinn+, a self-organizing incremental neural network for unsupervised learning from noisy data streams. *Expert Syst. Appl.*, 143.

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. 2022. QAConv: Question answering on informative conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5389–5411, Dublin, Ireland. Association for Computational Linguistics.

Pengtao Xie and Xuefeng Du. 2022. Performance-aware mutual knowledge distillation for improving neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11922–11932.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Ming Yan, Hao Zhang, Di Jin, and Joey Tianyi Zhou. 2020. Multi-source meta transfer for low resource multiple-choice question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7331–7341, Online. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*.

Hamed Zamani, Michael Bendersky, Donald Metzler, Honglei Zhuang, and Xuanhui Wang. 2022. Stochastic retrieval-conditioned reranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 81–91.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 656–661, Melbourne, Australia. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Xing Zhang, Zuxuan Wu, Zejia Weng, Huazhu Fu, Jingjing Chen, Yu-Gang Jiang, and Larry S. Davis. 2021. Videolt: Large-scale long-tailed video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7960–7969.

Yunxiang Zhang and Xiaojun Wan. 2022. Birdqa: A bilingual dataset for question answering on tricky riddles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10, pages 11748–11756.

Bingchen Zhao and Kai Han. 2021. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. *Advances in Neural Information Processing Systems*, 34:22982–22994.

Wanjun Zhong, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022a. ProQA: Structural prompt-based pre-training for unified question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4230–4243, Seattle, United States. Association for Computational Linguistics.

Wanjun Zhong, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022b. Proqa: Structural prompt-based pre-training for unified question answering.

A Datasets and Metrics

Datasets. We carry out experiments on the following datasets:

- Extractive: SQuAD 1.1 (Rajpurkar et al., 2016), SQuAD 2 (Rajpurkar et al., 2018), NewsQA (Trischler et al., 2017) Quoref (Dasigi et al., 2019), ROPES (Lin et al., 2019), AdversarialQA (Bartolo et al., 2020), ReCoRD (Zhang et al., 2018),

- Abstractive: DROP (Dua et al., 2019) NarrativeQA/NarQA (Kočiský et al., 2018), the open-domain version of NaturalQuestions/-NatQA (Kwiatkowski et al., 2019), QA-Conv (Wu et al., 2022), TweetQA (Xiong et al., 2019),

- Multiple-choice: HeadQA (Vilares and Gómez-Rodríguez, 2019), RACE-C (Liang et al., 2019), MCTest (Richardson et al., 2013), RACE (Lai et al., 2017), OpenBookQA (Mihaylov et al., 2018) ARC (Clark et al., 2018, 2016), QASC (Khot et al., 2020), CommonsenseQA/CQA (Talmor et al., 2019), Winogrande (Sakaguchi et al., 2020), MMMLU (Hendrycks et al., 2021), ReClor (Yu et al., 2020), Quail (Rogers et al., 2020), OneStopQA (Berzak et al., 2020), MCScript (Ostermann et al., 2018), MCScript 2.0 (Ostermann et al., 2019), CosmosQA (Huang et al., 2019), ProcessBank (Berant et al., 2014), DREAM (Sun et al., 2019), PROST (Aroca-Ouellette et al., 2021), PhysicalIQA/PIQA (Bisk et al., 2020), SocialIQA/SIQA (Sap et al., 2019)

- Yes/no: BoolQ (Clark et al., 2019), BoolQ-NP (Khashabi et al., 2020a) the binary (yes/no) subset of MultiRC (Khashabi et al., 2018), StrategyQA (Geva et al., 2021), PubmedQA (Jin et al., 2019).

The statistics of these datasets are summarized in Table 8. Note that we follow the pre-process scheme released by Khashabi et al. (2020b) to tackle these datasets. As 22 tasks are unseen in the training phase, we only use the training and validation sets of the other 21 tasks to build our framework.

Metrics. The evaluation for each task follows Khashabi et al. (2022). Specifically, for Multiple-choice tasks, we use accuracy. For Extractive tasks, we use the F1 token overlap between the answer text and golden truth. For Abstractive tasks, we use ROUGE-L for NarrativeQA, BLEU for TweetQA, and F1 for the other tasks. For Yes/no questions, we also use the F1 token overlap.

B Overall Results

We compare our OLTQA model with competitive baselines and ablation variants on each component. The full results of our model, baselines and ablation variants under 21 seen tasks are shown in Table 5, while the results under 22 unseen tasks are shown in Table 6. Bold numbers are superior results.

C Case Study

We provide examples from tail and unseen tasks, where our model is correct and the variant without knowledge mining (i.e., w/o $\mathcal{P}_k$) is incorrect, together with top hints selected by the retrieve-then-rerank framework. Table 7 demonstrates that hints yielded by our model are related to the ground truth which effectively corrects the predicted answer.

D More Implementation Details

We use T5-base (Raffel et al., 2020) to initialize our encoder-decoder QA model (12 layers, 768-dimensional hidden size, and 12 attention heads). In knowledge sharing, we maintain totally $s = 30$ meta prompts, and set the length of each meta prompt to 10. We adopt a fixed T5-base encoder with an average pooling layer to generate the query vector. For each instance $\langle \boldsymbol{c}, \boldsymbol{q}, \boldsymbol{a} \rangle$, we select $\widetilde{s} = 5$ meta prompts to construct $\mathcal{P}_m$. For meta prompt key training, we set $\eta = 0.15$ and $\gamma = 0.3$ in Eq. 1.

In knowledge mining, we adopt GLM-10B (Du et al., 2022) with 10B parameters as a large pre-trained LM. For retrieve-then-rerank example selection, R_1 first retrieves $l = 64$ examples from all training examples, and R_2 selects $\widetilde{l} = 4$ examples among retrieval results. The retriever R_1 is implemented with two separate dense encoders $E_X(\cdot)$ and $E_D(\cdot)$ to map $\langle \boldsymbol{c}, \boldsymbol{q} \rangle$ and $\boldsymbol{e}_i$ into vectors. The score for $\boldsymbol{e}_i$ is then computed as $E_X([\boldsymbol{c}; \boldsymbol{q}])^T \cdot E_D(\boldsymbol{e}_i)$, which is the dot product of two vectors. The re-ranker R_2 is a dense encoder E_C combined with a linear layer f_c. Concretely, E_C transforms the concatenation of example $\boldsymbol{e}_i$, hint $\boldsymbol{h}_i$ and input $\langle \boldsymbol{c}, \boldsymbol{q} \rangle$ into a representation, which is fed into f_c to get the score, denoted as $f_c(E_C([\boldsymbol{e}_i; h_i; \boldsymbol{c}; \boldsymbol{q}]))$. E_C,E_D and E_X are all initialized with BERT base uncased (Devlin et al., 2019). In two-stage training, we leverage BM25 to select $c = 512$ example candidates.

All experiments are performed on 8 A100 GPUs (80GB). The batch size is set to 32. We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e-4 and batch size of 32. The dataset is trained for five epochs. All hyper-parameters are tuned according to the average score on the validation set. In our experiments, We perform 3 runs by setting the random seed to $\{42, 43, 44\}$ respectively. In this way, we report the average score of each method. Note that we only use the random seed 42 for tuning hyper-parameters. Our model has 551.59M tunable parameters.

To obtain the ROUGE-L score, we use the NLTK package for sentence tokenization, and python rouge-score package for evaluation. To obtain the BLEU score, we use the NLTK package for evaluation.

E Results under Different Random Seeds

We use random seed 42 and 43 to construct another two sets of head, tail, and unseen tasks, and compare our method with the baseline UnifiedQA. As shown in Table 9, our method is robust when using different tasks as head, tail or unseen tasks.

Methods	SQuAD 2	NatQA	RACE	SQuAD 1.1	DROP	NarQA	Winogrande	SIQA
UnifiedQA	77.80	40.25	56.97	85.32	32.50	44.69	54.93	50.15
ProQA	79.84	39.01	59.55	84.33	31.66	34.20	54.62	**54.50**
Muppet	79.41	40.83	57.13	85.64	32.62	45.30	55.49	52.63
Hyperformer++	79.52	40.24	58.24	87.13	32.17	51.88	54.93	52.46
EPR	44.14	39.50	38.82	87.12	29.22	46.02	51.70	45.96
Ours (w/o $\mathcal{P}_m$)	77.72	42.10	58.13	85.98	35.53	56.89	54.85	49.64
Ours (w/o $\mathcal{P}_k$)	78.89	40.20	59.34	86.02	32.80	44.56	54.78	51.76
Ours (w/o MKD)	78.81	42.13	58.95	87.39	35.59	55.86	54.62	49.85
Ours (BM25 Retriever)	78.49	41.82	58.22	84.96	34.62	56.63	49.64	50.41
Ours (EPR Retriever)	77.51	42.13	59.36	87.09	35.01	56.87	54.54	51.23
Ours (w/o Re-ranker)	77.94	41.50	57.64	86.73	34.54	56.04	55.56	50.67
Ours (Static MKD)	78.73	42.67	59.55	87.72	35.81	57.34	55.33	51.48
Ours (Back KD)	78.16	42.07	58.17	86.66	35.61	54.68	54.06	50.72
Ours	**79.99**	**42.68**	**59.65**	**87.88**	**36.42**	**57.59**	**55.64**	52.51

Methods	Quoref	ROPES	CQA	BoolQ-NP	BoolQ	QASC	OBQA	PIQA
UnifiedQA	56.28	57.90	51.92	67.69	73.28	34.88	36.73	54.35
ProQA	35.75	30.10	51.52	69.67	72.51	31.10	43.40	56.31
Muppet	57.66	55.42	53.79	68.84	74.27	32.62	39.47	55.47
Hyperformer++	60.80	57.04	53.24	67.66	73.58	33.15	41.00	55.60
EPR	48.54	47.96	45.30	59.43	70.70	38.09	38.07	55.55
Ours (w/o $\mathcal{P}_m$)	67.20	54.00	56.91	71.76	75.64	43.09	43.53	54.46
Ours (w/o $\mathcal{P}_k$)	56.32	57.96	52.50	70.64	74.62	36.83	39.53	55.98
Ours (w/o MKD)	69.00	52.66	55.61	71.77	76.18	46.00	43.80	55.22
Ours (BM25 Retriever)	68.09	54.10	52.66	71.07	72.84	42.76	39.00	**56.43**
Ours (EPR Retriever)	68.73	54.21	54.95	71.22	76.24	43.63	39.33	54.68
Ours (w/o Re-ranker)	65.38	53.28	52.83	72.18	73.17	39.52	39.67	53.70
Ours (Static MKD)	69.12	54.67	56.10	70.88	77.03	48.92	40.47	55.73
Ours (Back KD)	69.18	55.51	56.73	71.36	76.21	**51.08**	42.40	55.84
Ours	**69.42**	**58.64**	**57.08**	**73.41**	**78.78**	50.65	**44.27**	56.09

Methods	NewsQA	ARC-easy	MCTest	ARC-hard	MultiRC	Head@5	Tail@16	A_{seen}
UnifiedQA	57.48	36.84	77.19	31.77	80.45	58.57	54.16	55.21
ProQA	49.93	44.21	80.00	38.13	77.56	58.88	51.47	53.23
Muppet	58.11	38.07	79.06	31.34	85.57	59.13	55.19	56.13
Hyperformer++	59.45	40.18	76.88	31.10	86.86	59.46	55.99	56.81
EPR	18.26	51.81	55.00	39.80	56.41	47.76	48.04	47.97
Ours (w/o $\mathcal{P}_m$)	**59.70**	56.49	83.02	39.46	85.58	59.89	59.51	59.60
Ours (w/o $\mathcal{P}_k$)	58.87	39.82	76.25	33.11	85.90	59.45	55.59	56.51
Ours (w/o MKD)	58.88	57.37	82.19	39.46	84.94	60.57	59.59	59.82
Ours (BM25 Retriever)	59.20	53.16	81.56	34.78	78.85	59.62	57.57	58.06
Ours (EPR Retriever)	58.99	56.49	81.98	36.12	83.65	60.22	58.93	59.24
Ours (w/o Re-ranker)	59.49	51.58	80.94	37.15	87.18	59.67	58.02	58.41
Ours (Static MKD)	58.83	57.54	81.87	39.46	82.54	60.90	59.83	60.09
Ours (Back KD)	58.87	57.89	**85.63**	40.22	83.18	60.13	60.24	60.21
Ours	59.41	**58.95**	83.75	**40.43**	**87.82**	**61.32**	**61.53**	**61.48**

Table 5: Comparison with competitive baselines and all ablations of our model in 21 seen tasks. Bold numbers are superior results.

Methods	AdversarialQA dBERT	AdversarialQA dBiDAF	AdversarialQA dRoberta	ReCoRD	RACE-C	HeadQA	MMMLU	ReClor
UnifiedQA	24.39	44.24	18.16	19.62	49.86	29.14	28.77	35.73
ProQA	24.13	41.67	14.21	13.42	54.91	29.84	25.96	**37.60**
Muppet	22.10	43.35	17.33	16.71	50.00	29.04	30.42	33.53
Hyperformer++	20.09	45.30	16.99	17.74	52.11	28.62	25.26	35.47
EPR	37.00	53.76	27.74	8.98	35.39	32.21	28.77	25.07
Ours (w/o $\mathcal{P}_m$)	34.51	51.42	25.16	13.76	53.51	34.55	33.68	33.73
Ours (w/o $\mathcal{P}_k$)	24.29	43.71	17.12	**19.03**	53.23	29.36	31.23	32.60
Ours (w/o MKD)	32.94	52.86	24.54	13.72	49.30	**35.14**	32.63	35.40
Ours (BM25 Retriever)	35.10	53.57	25.96	11.15	50.14	32.87	32.98	32.67
Ours (EPR Retriever)	37.26	54.58	26.80	14.11	53.65	34.00	32.72	34.73
Ours (w/o Re-ranker)	36.93	53.99	27.33	15.55	53.65	32.77	31.93	35.80
Ours (Static MKD)	32.47	53.13	24.89	13.80	54.21	35.07	34.39	32.93
Ours (Back KD)	31.66	53.91	24.91	15.64	53.14	35.00	32.63	34.89
Ours	**39.51**	**55.12**	**28.05**	17.97	**56.88**	34.48	**36.14**	36.67

Methods	Quail	OneStopQA elementary	OneStopQA intermediate	OneStopQA advanced	MCScript	MCScript 2.0	CosmosQA	DREAM
UnifiedQA	53.31	53.09	55.25	54.01	67.97	77.38	37.42	59.56
ProQA	54.16	62.35	62.65	61.11	71.23	76.44	39.23	64.41
Muppet	52.86	54.33	56.17	54.79	70.91	76.97	35.75	58.61
Hyperformer++	54.09	54.63	55.86	59.88	71.51	76.62	37.35	59.31
EPR	41.29	63.58	58.95	60.49	65.56	63.56	38.66	53.92
Ours (w/o $\mathcal{P}_m$)	56.17	60.19	62.96	61.11	77.46	76.88	45.09	68.28
Ours (w/o $\mathcal{P}_k$)	52.94	56.67	57.72	56.70	70.80	77.57	39.87	60.29
Ours (w/o MKD)	55.43	54.32	57.41	54.32	75.69	78.22	45.46	67.35
Ours (BM25 Retriever)	55.06	58.64	58.02	58.95	78.03	79.65	45.36	68.71
Ours (EPR Retriever)	55.20	60.80	60.49	60.19	76.97	76.98	45.96	69.17
Ours (w/o Re-ranker)	52.98	59.57	55.25	57.10	74.49	77.48	45.03	64.75
Ours (Static MKD)	55.29	61.73	60.49	59.26	74.63	77.97	43.92	68.82
Ours (Back KD)	**57.98**	61.16	59.88	60.60	77.18	**79.85**	45.78	69.40
Ours	56.96	**65.12**	**65.74**	**64.31**	**79.16**	78.27	**46.16**	**69.51**

Methods	ProcessBank	PROST	StrategyQA	PubmedQA	QAConv	TweetQA	A_{unseen}
UnifiedQA	70.75	31.73	40.50	50.53	61.43	64.52	46.70
ProQA	69.39	31.30	49.96	58.00	59.73	63.83	48.27
Muppet	73.47	28.99	43.62	56.73	61.82	66.02	46.98
Hyperformer++	72.79	32.34	49.52	53.00	58.93	61.44	47.22
EPR	70.07	30.33	42.08	59.67	60.72	66.65	46.57
Ours (w/o $\mathcal{P}_m$)	77.55	31.82	49.38	62.07	62.36	74.27	52.09
Ours (w/o $\mathcal{P}_k$)	75.51	32.80	49.39	56.27	60.99	66.02	48.37
Ours (w/o MKD)	74.83	31.66	**51.44**	61.60	62.18	73.33	50.90
Ours (BM25 Retriever)	75.28	31.43	51.35	58.93	61.39	76.44	51.44
Ours (EPR Retriever)	75.06	32.60	49.24	60.53	61.80	74.14	52.14
Ours (w/o Re-ranker)	73.02	29.80	51.31	61.60	62.26	69.53	51.01
Ours (Static MKD)	74.15	32.09	49.18	63.87	**63.46**	75.60	51.88
Ours (Back KD)	74.68	30.81	51.40	62.73	63.39	75.18	52.35
Ours	**78.91**	**33.68**	50.70	**64.40**	62.28	**77.17**	**54.42**

Table 6: Comparison with competitive baselines and all ablations of our model in 22 unseen tasks. Bold numbers are superior results.

Task	Ours (w/o $\mathcal{P}_k$)	Ours
NarQA	**Input**:The play begins with three...WHAT SENTENCE DID CYNTHIA GIVE TO THE SYMBOLIC VICES? **Ground Truth**:Make reperations and purify themselves.	
	Output:To make reparation and purify themselves by bathing in the spring.	**Hints**: To make reparation and to purify yourselves; Make reparation and to purify themselves by bathing in the spring at Mount Helicon.; Make reparation and purify yourselves.; Make reparation and purge yourselves **Output**:Make reparation and purify themselves
ARC-hard	**Input**:A daphnia population... To which factor is the daphnia population most likely responding? (A) the pH of... **Ground Truth**:the temperature of the water	
	Output:the pressure of the water	**Hints**: light intensity; temperature; the temperature; the temperature of the water. **Output**:the temperature of the water
NewsQA	**Input**:RIO DE JANEIRO, Brazil (CNN) – A Brazilian supreme court judge...When did the mother die? **Ground Truth**:September	
	Output:June 2004	**Hints**:in September; September.; during childbirth; to David Goldman. **Output**:September
MultiRC	**Input**:German art collector...Was the Gurlitt art collection returned after confiscation? **Ground Truth**:yes	
	Output:no	**Hints**: the surviving paintings were all returned; part of the collection was returned; part of it was; recently **Output**:yes
ReCoRD	**Input**:Lionel Messi is unattainable...Ariedo braida (pictured) says that it would be a mistake for _ to change teams.. **Ground Truth**:Lionel Messi	
	Output:it would be a mistake for _ to change teams	**Hints**: Barcelona; Lionel Messi is unattainable for most football clubs; change teams; Messi is an icon of world football **Output**:Lionel Messi
TweetQA	**Input**:The way they run to each other... what does the tweeter imply? **Ground Truth**:they like each other	
	Output:No Answer>	**Hints**: I had great time with my kids; they really like each other; They want to know each other.; they are attracted to each other. **Output**:they are attracted to each other.
StrategyQA	**Input**:(Gulf of Finland) The bottom of...Would the Titanic be well preserved at the bottom of the Gulf of Finland? **Ground Truth**:yes	
	Output:ships are relatively well preserved	**Hints**: yes; yes, it would be well preserved; Yes, it would.; well preserved **Output**:yes
RACE_C	**Input**:Many post-80s...Many post-80s couples can't go to the movies, shop or attend parties because _.? (A) they ... **Ground Truth**:they have to look after their kids	
	Output:they have to look after their parents	**Hints**: their kids are born; their kids were born; kids were born; they have to look after their kids **Output**:they have to look after their kids

Table 7: Case study from tail and unseen tasks where our model is correct and the variant without knowledge mining (i.e., w/o $\mathcal{P}_k$) is incorrect along with the top 4 hints selected by the retrieve-then-rerank framework.

Format	Dataset	Train set size	Val set size	Test set size
Extractive	SQuAD1.1	7978	886	10570
	SQuAD2	127319	3000	11873
	NewsQA	436	54	4341
	Quoref	1539	192	2768
	ROPES	1242	155	1688
	AdversarialQA(dBERT)	-	-	1000
	AdversarialQA(dBiDAF)	-	-	1000
	AdversarialQA(dRoberta)	-	-	1000
	ReCorD	-	-	9999
Abstractive	NarQA	3487	435	6922
	NQOpen	31843	3980	10693
	Drop	5095	636	9536
	QAConv	-	-	3414
	TweetQA	-	-	1086
Multiple-choice	RACE	14205	1775	4887
	OBQA	566	70	500
	MCTest	335	41	320
	ARC-easy	386	48	570
	ARC-hard	309	38	299
	CQA	1011	126	1221
	QASC	638	79	926
	PIQA	482	60	1838
	SIQA	2031	253	1954
	Winogrande	2573	321	1267
	RACE-C	-	-	712
	HeadQA	-	-	1366
	MMMLU	-	-	285
	ReClor	-	-	500
	QuAIL	-	-	2163
	OneStopQA elementary	-	-	324
	OneStopQA intermediate	-	-	324
	OneStopQA advanced	-	-	324
	MCScript	-	-	1411
	MCScript 2.0	-	-	2020
	CosmosQA	-	-	2985
	ProcessBank	-	-	147
	DREAM	-	-	2040
	PROST	-	-	18736
Yes/no	BoolQ	748	93	3270
	MultiRC	284	28	312
	BoolQ-NP	899	112	7596
	StrategyQA	-	-	2290
	PubmedQA	-	-	500

Table 8: Dataset Statistics.

Seed	Method	Head@3	Tail@4	A_{seen}	A_{unseen}
42	UnifiedQA	49.68	56.54	47.74	40.19
	Ours	**53.10**	**66.29**	**56.03**	**49.76**
43	UnifiedQA	56.71	50.05	50.65	42.67
	Ours	**62.08**	**66.68**	**59.98**	**51.05**

Table 9: Results on different random seeds.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section4, Appendix A, Appendix D

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4, Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4, Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Appendix D

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix D

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.