Retrieve What You Need: A Mutual Learning Framework for Open-domain Question Answering

Dingmin Wang[1*] **Qiuyuan Huang**[2] **Matthew Jackson**[1] **Jianfeng Gao**[2]
[1]University of Oxford, UK [2]Microsoft Research, Redmond, USA
dingmin.wang@cs.ox.ac.uk, jackson@robots.ox.ac.uk
{qihua,jfgao}@microsoft.com

Abstract

An open-domain question answering (QA) system usually follows a *retrieve-then-read* paradigm, in which a *retriever* is used to retrieve relevant passages from a large corpus, and then a *reader* generates answers based on the retrieved passages and the original question. In this paper, we propose a simple and novel mutual learning framework to improve the performance of *retrieve-then-read*-style models via an intermediate module named the *knowledge selector*, which we train with reinforcement learning. The key benefits of our proposed intermediate module are: 1) no requirement for additional annotated question-passage pairs; 2) improvements in both retrieval and QA performance, as well as computational efficiency, compared to prior competitive *retrieve-then-read* models; 3) with no finetuning, improvement in the *zero-shot* performance of large-scale pre-trained language models, e.g., ChatGPT, by encapsulating the input with relevant knowledge without violating the input length constraint.

1 Introduction

Recently, there has been a revival of interest in tasks requiring large amounts of knowledge of the world. In such real-world scenarios, an efficient information retrieval system, capable of finding a small subset of relevant and non-redundant information, is needed for applications such as open-domain question answering, in which external knowledge (e.g., Wikidata and ConceptNet [Speer et al., 2017]) must be integrated in order to generate correct answers. Even in the era of Large Language Models like ChatGPT and GPT-4, which are capable of encoding extensive knowledge into their parameters, there are still scenarios where retrieval is indispensable, such as when answering questions about the most current news events. However, hand-labeling data for training such a retriever is expensive and time consuming, and many datasets and applications lack such annotations. Hence, an efficient framework should be capable of learning a retriever, without supervision from annotated query-passage pairs.

In this paper, we focus on improving both the inference performance and efficiency of *retrieve-then-read* frameworks. Retrieve-then-read frameworks have dominated over current open-domain question answering systems (Oguz et al., 2022; Izacard and Grave, 2021; Cheng et al., 2021; Ma et al., 2022b) as well as other knowledge-intensive tasks such as fact checking (Petroni et al., 2021; Martín et al., 2022) and dialogue systems (Zhang et al., 2021). For example, CORE (Ma et al., 2022a), a state-of-the-art open-domain question-answering system, starts by using a dense *retriever* (Karpukhin et al., 2020a) to retrieve a subset of support passages and tables from a large knowledge source, such as Wikipedia. Then, a generative encoder-decoder (*reader*) model produces an answer, conditioned on the question and the retrieved knowledge.

Previous studies (Yu et al., 2022b; Varshney et al., 2022) have shown that using a large number of support passages will lead to a significant increase in memory requirement and training time cost. According to Varshney et al. (2022), FiD (Izacard and Grave, 2020) requires approximately 7×10^{12} floating-point operations (FLOPs) for inference on 100 passages. This high inference cost limits the widespread adoption of such systems in real-world applications, which must trade-off performance for decreased latency. In addition to this, empirical results from previous work (Clark and Gardner, 2018; Yang et al., 2019; Wang et al., 2019; Lewis et al., 2020b) have suggested that, beyond a threshold number of passages, supplying the reader with additional passages yields only

*Work done when interning at Microsoft Research.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 247–263, 2024. https://doi.org/10.1162/tacl_a_00646
Action Editor: Roi Reichart. Submission batch: 4/2023; Revision batch: 10/2023; Published 3/2024.

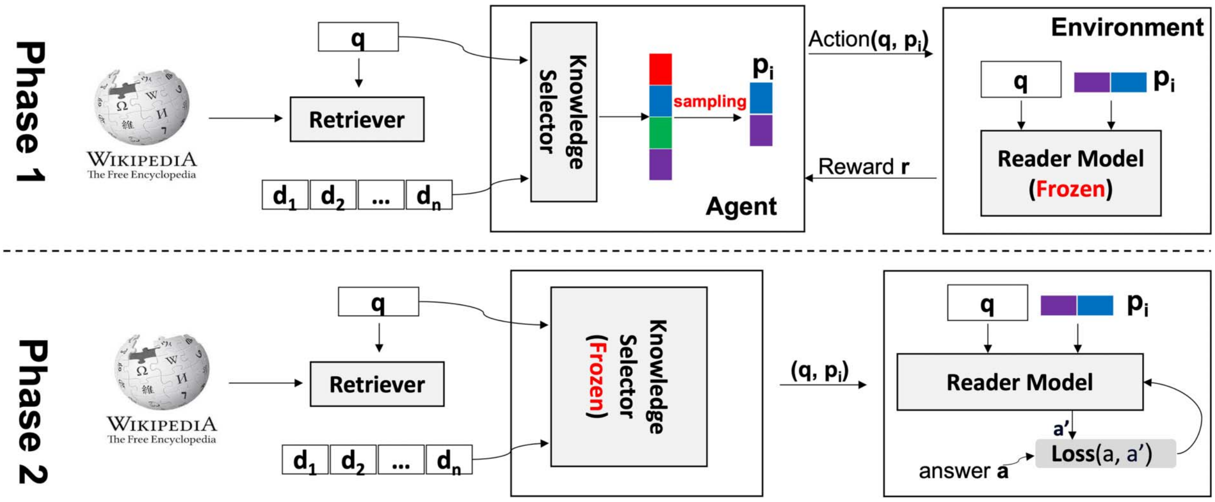

Figure 1: Architecture of our proposed mutual learning framework. In each epoch, **Phase 1** and **Phase 2** are executed alternately. During **Phase 1**, the parameters of the reader model remain fixed, and only the weights of the knowledge selector are updated. Conversely, during **Phase 2**, the reader model's parameters are adjusted, while the knowledge selector's weights remain frozen.

a minimal improvement or even decline in the overall accuracy of the end-to-end QA systems. These two points motivate us to explore whether it is possible to reduce the number of required support passages without compromising the model's performance. To this end, we conducted two preliminary experiments:

Preliminary Experiment 1: Given a sample from the TQA (Joshi et al., 2017) dataset in which each question is accompanied with 100 passages retrieved by DPR (Karpukhin et al., 2020b), we achieved an exact match (EM) score of 65.0% using a Fusion-in-Decoder model (T5-base). We then calculated the average EM scores when using 10 passages under a range of selection strategies. Firstly, by randomly sampling 10 out of the 100 passages retrieved by DPR, the EM scored decreases from 65.0% to 53.3%. Selecting the top 10 passages ranked by DPR outperformed this random sampling, however the EM score still degraded to 59.6%. Finally, to further substantiate our hypothesis that retrieving more relevant passages can significantly enhance the reader's generation performance, we employed Contriever.[1] Contriever is an advanced unsupervised dense information retrieval system trained on extensive corpora. Utilizing Contriever to select a set of 10 passages, we observed an EM score of 65.4%,[2] a slight improvement against the original 100 passages. To a certain extent, it shows that the importance of retrieved results lies more in their quality rather than their quantity.

Preliminary Experiment 2: We randomly chose 20 questions, each with 100 retrieved passages using DPR. We then presented three student volunteers with the question-passage pairs, and asked them to estimate how many passages they would require to obtain the answer. From their response, we observed an average of 7.5 passages required to answer the question, suggesting that a large portion of retrieved passages are redundant.

The above two preliminary results align with our conjecture that selecting a smaller portion of relevant support passages—instead of feeding a large number of passages to the reader—is a viable research direction. To this end, we propose a novel mutual learning framework (Figure 1) that improves both the quality of the retrieved passages and the performance of the reader. The key novelty of our framework is the introduction of a "*knowledge selector*" module, which interfaces between the retriever and reader. The goal of the

[1]Contriever is available at `https://github.com/facebookresearch/contriever`, which has demonstrated its ability to achieve competitive retrieval performance across various QA benchmarks. We use Contriever-msmarco version due to its competitive retrieval performance.

[2]Notably, our statistical analysis reveals that for 98.4% of the testing samples, the 10 passages selected by Contriever are already encompassed within the original 100 passages retrieved by DPR, although the ranking may vary slightly.

knowledge selector is to further refine the set of passages selected by the retriever, which we frame as a reinforcement learning problem. We train this system by iterating between two phases, which train the knowledge selector and reader respectively. In the first phase (**Phase 1**), we use policy gradients to train the knowledge selector to select the optimal subset of support passages, with the goal of maximizing the prediction rewards when passed to the reader (whose parameters are frozen at this phase). Following this, in **Phase 2**, we freeze the weights of the knowledge selector and train the reader using supervised learning over pairs of questions and K passages selected by the knowledge selector.

We validate the effectiveness of our proposed method on three benchmarks of knowledge-grounded open-domain question answering: Natural Questions (NQs) (Kwiatkowski et al., 2019), TQA (Joshi et al., 2017), and WEBQUESTIONS (WebQ) (Berant et al., 2013). Evaluation results on these benchmarks demonstrate that our framework achieves superior performance to existing models, thus setting a new state-of-the-art. Furthermore, we carried out experiments to showcase the generalizability of our trained knowledge selection module in different retrievers and readers in a zero-shot fashion. Our results indicate that this module can boost the generation performance of large-scale language models (LLMs) such as GPT-3 and ChatGPT when used in conjunction with retrievers. We hypothesize that this enhancement is due to the module's ability to select more relevant external knowledge, thereby empowering the LLMs to produce more precise answers.

2 Our Method

To improve both the inference efficiency and prediction accuracy we propose a simple and novel mutual learning framework for training an open-domain question answering system. Our framework inserts a *knowledge selector* module between the *retriever* and the *reader*. Crucially, this module requires no additional annotated data and is compatible with any *retrieve-then-read* models.

Specifically, given a question q_i, the retriever first selects a fixed number of passages D_i from a large knowledge source. Then, the knowledge selector prunes D_i to obtain a smaller subset of passages p_i, where $p_i \subseteq D_i$. Finally, p_i is processed by the reader, along with the question, to generate an answer. For the retriever, we use DPR (Karpukhin et al., 2020a), which has been demonstrated to perform better than sparse-representation-based methods, such as BM25 (Robertson et al., 2009), in many prior works (Izacard and Grave, 2020, 2021). For the reader module, we use the Fusion-in-Decoder (FiD) model (Izacard and Grave, 2020), a sequence-to-sequence architecture which we initialize from a pre-trained model such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020a).

Information retrieval has been studied for many years and there exists an abundance of off-the-shelf retrieval models. After reviewing previous work in open-domain question answering, we find three main classes of retriever: 1) sparse retrievers (e.g., BM25), where both passages and queries are represented as sparse vectors, with each dimension corresponding to a different term; 2) unsupervised dense retrievers (e.g., Contriever [Izacard et al., 2021]), which are trained without using annotated query-passage pairs; and 3) supervised dense retrievers (e.g., DPR), which represent a cluster of supervised dense retrieval model directly trained on annotated datasets. Since it is not the main focus of our work, we directly adopt DPR as our retriever, a state-of-the-art retrieval model.

In the following two sections, we outline the training details of the two remaining modules: *knowledge selector* (§2.1) and *reader* (§2.2).

2.1 Knowledge Selector Agent

A key novelty of this work is to train the *knowledge selector* without requiring a task-specific annotated training dataset. By framing the passage selection problem as a contextual multi-arm bandit (Robbins, 1952), we propose training the knowledge selector using a policy gradient strategy. This avoids brute-force search over all passage combinations or task-specific heuristics.

Given a question and a passage set from the passage retriever, the knowledge selector chooses a fixed number of relevant passages from the passage set. This refined passage set and the original question are then fed to the reader model, which produces an answer. Finally, the answer is evaluated against the ground truth, from which an associated loss is computed. In this setting, the knowledge selector follows the dynamics of a multi-arm bandit, where the context consists of the question and the action space is composed of all subsets of the passage set (of a given size). Crucially, this is not an unrestricted Markov decision process (Sutton and Barto, 2018), since there is no temporal dependency between question-answer pairs.

Formally, the *knowledge selector* π_θ is built on BERT (Devlin et al., 2019) together with a small linear layer on top of BERT. The parameters of BERT are fixed and only the appended linear is updated, i.e., θ is composed of learnable parameters $\mathbf{W}$ and $\mathbf{b}$. Given a question q_i and a set of candidate passages $\mathcal{D}_i$ retrieved by the aforementioned retriever, we want the ''agent'' to find the K best performing passage set p_i from the candidate pool $\mathcal{D}_i$. The agent's goal is that the reader can generate an answer $\hat{a}_i$ based on (q_i, p_i), obtaining the maximum reward $r(\hat{a}_i|q_i, p_i)$.

Mathematically, the agent samples the passage set p_i according to the policy

$$p_i \sim \pi_\theta(p_i|q_i), \quad p_i \subseteq \mathcal{D}_i \tag{1}$$

Here, the policy π_θ computes the sampling probability for each passage $d \in \mathcal{D}_i$ as

$$f(d|q_i) = \frac{\exp\left[\mathbf{h}(d) \cdot \mathbf{h}(q_i)\right]}{\sum_{d_j \in \mathcal{D}_i} \exp\left[\mathbf{h}(d_j) \cdot \mathbf{h}(q_i))\right]} \tag{2}$$

where $\mathbf{h}(x) = \mathbf{W}(\text{BERT}(x)) + \mathbf{b}$. The policy then samples K passages $d_k \sim f(d|q_i)$ from this distribution without replacement, giving the passage set $p_i = \{d_1, \ldots, d_K\}$.

In this phase, the answer $\hat{a}_i$ is generated by a fixed-parameter reader, whose input contains the question q_i and the passage set p_i. More details about the reader will be illustrated in next section (§2.2). The reward $r(\hat{a}_i|q_i, p_i)$ is obtained by evaluating the generated answer $\hat{a}_i$ against the ground truth answer list A_i.[3] Specifically, we use a 0–1 loss as our reward function, which is defined as follows,

$$r(\hat{a}_i|q_i, p_i) = \begin{cases} 1, & \hat{a}_i \in A_i \\ 0, & \hat{a}_i \notin A_i \end{cases} \tag{3}$$

Note that the proper design of reward functions, a.k.a. reward engineering, is critical for training efficiency in reinforcement learning (Sutton and Barto, 2018). While different reward functions might further improve the performance, we leave this as an area for future work.

We optimize the agent with the REINFORCE policy gradient operator (Williams, 1992), maximizing the following objective function:

$$\mathcal{J}(\theta) = \mathbb{E}_{(q_i, p_i) \sim \pi_\theta(p_i|q_i)}[r(\hat{a}_i|q_i, p_i)] \tag{4}$$

Intuitively, we update the policy to increase the probability of sampling the selected passages if the predicted answer is correct, and decrease their probability if the predicted answer is incorrect.

2.2 FiD-based Reader

The *reader* takes the selected passages from *knowledge selector* and the question as input and generates an answer. To make the input compatible with sequence-to-sequence models like T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a), one way is to concatenate the question with all the passages and let the self-attention in the Transformer module do the cross-passage reasoning. However, this can be inefficient when the number of retrieved passages is very large because of the quadratic computation complexity in self-attention. To achieve both cross-passage modeling and computation efficiency, we take as our reader FiD model (Izacard and Grave, 2020), which achieves state-of-the-art performance and is widely adopted by prior work (Ma et al., 2022a; Izacard and Grave, 2021). The underlying architecture is a sequence-to-sequence model, composed of an encoder and a decoder, and initialized from pre-trained models such as T5 or BART.

For a given question q_i and a set of passages p_i of size K, we concatenate question q_i with each passage, thus resulting in K question-passage pairs. In particular, following Izacard and Grave (2020), for each question and a passage, we add sentinel tokens `question:`, `title:`, and `context:` before the question, the passage title, and the passage content separately. The encoder independently processes K different question-passage pairs. The token embeddings of all passages output from the last layer of the encoder are concatenated as a global representation $\mathbf{H}$ of dimension $(\sum_{k=1}^{K} \ell_k) \times d$, where ℓ_k is the length of the k-th question-passage pair and d is the dimension of the embeddings and hidden representations of the model. $\mathbf{H}$ is then sent to the decoder to generate the expected answer in a regular

[3] A question might correspond to one or multiple answers.

Algorithm 1: Two-phase Training.

Input : $\mathcal{D}$: question-answer pairs , $\mathcal{E}$: an external source , $epochs$: number of epochs, Φ : fixed-parameter retriever , initialized *knowledge selector* π_θ and *reader* Ψ_ϕ, n : number of passages retrieved by Φ, K : number of passages selected by π.

for *for* $e = 1$ *to* $epochs$ **do**

Phase 1: (train *knowledge selector*)

for *each question* $(q_i, a_i) \in \mathcal{D}$ **do**

① retrieve n passages from $\mathcal{E}$ via Φ;

② select K passages p_i out of the n retrieved passages by $\pi_\theta(p_i|q_i)$;

③ generate $\hat{a}_i$ by $\Psi_\phi(\hat{a}_i|q_i, p_i)$;

④ compute the gradient of π_θ:

$$r(\hat{a}_i|q_i, p_i)\nabla_\theta[\log \pi_\theta(p_i|q_i)]$$

⑤ Update the parameters of π_θ;

end

Phase 2: (train FiD-based *reader*)

for *each question* $(q_i, a_i) \in \mathcal{D}$ **do**

① retrieve n passages from $\mathcal{E}$ via Φ;

② select K passages p_i out of the n retrieved passages by $\pi_\theta(p_i|q_i)$;

③ generate $\hat{a}_i$ by $\Psi_\phi(\hat{a}_i|q_i, p_i)$

④ compute the gradient of Ψ_ϕ:

$$\nabla_\phi \Psi_\phi(\hat{a}_i|q_i, p_i)$$

⑤ Update the parameters of Ψ_ϕ;

end

Save the optimal parameters of both π_θ and Ψ_ϕ by evaluating the validation dataset.

end

autoregressive manner, alternating self-attention, cross-attention and feed-forward modules.

By concatenating the encoder output embeddings, the decoder can generate outputs based on joint modeling of multiple passages. In this way, it means that the computation time of the model grows linearly with the number of used passages, instead of quadratically. Besides, processing passages jointly in the decoder allows to better aggregate evidence from multiple passages.

2.3 Two-phase Training Framework

We present our two-phase mutual-learning training framework in Algorithm 1. For each epoch, it goes through the whole training dataset twice for optimizing the parameters of *knowledge selector* π_θ and *reader* Ψ_ϕ, respectively.

At the first phase, we adopt a reinforcement learning (RL) approach to train our knowledge selector. The reason for choosing an RL-based approach contains mainly come from two considerations: One is that there are no annotated pairs of questions and the corresponding list of support passages, so we are unable to train the *knowledge selector* in a standard supervised training paradigm; another is that based on some prior works (Izacard and Grave, 2020, 2021) showing that the quality of the retrieved passages greatly influences the performance of the reader, we conjecture that the reward calculated based on the reader's prediction performance can serve as a good proxy for the relevance of support passages.

Ideally, we wold like the *knowledge selector* to select the best K performing passages from the whole external source $\mathcal{E}$. In practice, however, querying a large knowledge source is time- and memory-consuming, Thus, we use an off-the-shelf retrieval model to first retrieve n passages, which are expected to contain the most relevant passages if n is large enough (n=200). Then, we apply our trained *knowledge selector* to filter out some irrelevant passages to obtain a smaller set of passages p_i, which will then be sent together with the question q_i to the *reader* Ψ_ϕ for generating an answer $\hat{a}_i$. At this phase, $\hat{a}_i$ generated by Ψ_ϕ is only used to calculate the reward, which is then used to update the parameters of π_θ while keeping all the parameters of Ψ_ϕ unchanged.

At the second phase, we train the *reader* Ψ_ϕ together with our improved *knowledge selector* from the first phase. For Ψ_ϕ, we use the FiD model (Izacard and Grave, 2020), which has proven to be a state-of-the-art architecture by many prior studies (Izacard and Grave, 2021; Ma et al., 2022a). By processing passages independently in the encoder, but jointly in the decoder, this architecture allows to scale to large number of contexts, and meanwhile, the computation time of the model grows linearly with the number of passages, instead of quadratically.

3 Experiments

Datasets We evaluate our mutual learning framework by performing experiments on TriviaQA (TQA) (Joshi et al., 2017), NaturalQuestions

(NQ) (Kwiatkowski et al., 2019), and Web Questions (WebQ) (Berant et al., 2013) tasks:

- TQA contains a set of trivia questions with answers that were originally scraped from trivia and quiz-league websites. The original split uses 78,785 examples for training, 8,837 for validating, and 11,313 for testing.
- NQ were mined from real Google search queries with answers from Wikipedia articles identified by human annotators. The original split uses 79,168 examples for training, 8,757 for validating, and 3,610 for testing.
- WebQ consists of questions selected using Google Suggest API, where the answers are obtained via Amazon Mechanical Turk. The original split uses 3,478 examples for training, 300 for validating, and 2,032 for testing.

We use the Wikipedia dump from Dec. 20, 2018 for support passages, splitting articles into non-overlapping passages of 100 tokens, and applying the same pre-processing as Chen et al. (2017).

Evaluation Metrics The model performance is assessed in two ways. First, we report the top-k retrieval accuracy (R@k), which is the percentage of questions for which at least one passage of the top-k retrieved passages contains the gold answer. Additionally, we report the final end-to-end performance of the question-answering system composed of the retriever and reader modules. Predicted answers are evaluated with the standard exact-match metric (EM), as introduced by (Rajpurkar et al., 2016). An answer is considered to be correct if it is exact match with any of the reference answer strings after minor normalization such as lowercasing, following evaluation scripts from DrQA (Chen et al., 2017).

Unlike prior studies, we also consider floating-point operations (FLOPs) as the metric to evaluate computational efficiency. FLOPs are system-independent and hence a reliable metric for comparison. We compute these and other FLOP values using the *thop*[4] Python library.

3.1 Implementation Details

Off-the-shelf Retriever In this paper, unless otherwise specified, we use the DPR retriever (the $multi$-dataset version) by default, which is obtained using the script provided in the DPR official GitHub repository.[5] As described in Karpukhin et al. (2020b), the retriever ($multi$-dataset encoder) was trained over a combined training data of multiple datasets including NQ, TQA, and WebQ, using the in-batch negative setting. Since the retriever training is not the primary focus of this paper, we kindly refer readers to the comprehensive details provided in the paper (Karpukhin et al., 2020b). Additionally, for the BM25 retrieval method, we use the implementation from Apache Lucene[6] with default parameters, and tokenize questions and passages with SpaCy.[7]

Knowledge Selector and Reader We use the BERT large model with parameters fixed in the knowledge selection and the one trainable linear layer is parameterized with $\mathbf{W} \in \mathbb{R}^{1024\times1024}$ and the bias $\mathbf{b} \in \mathbb{R}^{1024}$. Similar to DPR (Karpukhin et al., 2020b), we use the combined training of NQ, TQA, and WebQ to train the knowledge selector. Namely, we only have one knowledge selector in our evaluation stage across the three different benchmarks. Unless otherwise specified, the reader is initialized with the T5 base model.

Both the knowledge selector and the reader are trained using the Adam algorithm (Kingma and Ba, 2014) linear scheduling with warm-up and dropout rate 0.1. The learning rate for the knowledge selector and the reader is 10^{-5} and 10^{-4}, respectively. The batch size is 8. In total, we ran 20 epochs using 8 Tesla V100 32GB, which took about 84 GPU hours. In each epoch, we run the two phases alternatively. The best pair of the knowledge selector and the reader models is selected based on the validation performance after the two-phase training at each epoch.

3.2 Main Results

Following previous work (Karpukhin et al., 2020b; Khattab et al., 2021; Izacard and Grave, 2021), we report the top-k retrieval accuracy. Table 1 compares four different passage retrieval schemes on three benchmark datasets, using the top-10 accuracy. Overall, the baseline retriever, whether employing BM25 or DPR, coupled with our specially trained knowledge selector, consistently attains superior scores compared to its

[4] https://github.com/Lyken17/pytorch-OpCounter.

[5] https://github.com/facebookresearch/DPR/blob/main/dpr/data/download_data.py#L258C5-L258C5.

[6] https://lucene.apache.org/.

[7] https://spacy.io/.

	NQ	TQA	WebQ
BM25	59.4	60.5	56.3
DPR	67.4	69.3	60.2
BM25 + KS	65.4	70.4	62.4
DPR + KS	**71.8**	**74.6**	**68.5**

Table 1: R@10 scores of four different retrieving schemes over three benchmark datasets.

baseline performance. However, as pointed out in Izacard and Grave (2021), the effectiveness of this metric in assessing the retriever's performance remains somewhat uncertain. This is due to the possibility of answers being present within a passage without a direct connection to the given question. Consequently, our next focus is on presenting the ultimate, end-to-end performance of the question-answering system, which encompasses both the retriever and reader modules. This is the metric that truly captures our primary interest.

In Table 2, we report the performance of our approach, as well as existing state-of-the-art systems on NQ, TQA, and WebQ with two different numbers of retrieved passages. The goal of this experiment is to validate whether the knowledge selector can effectively retain the passages required by the reader while filtering irrelevant passages, thus achieving the goal of improving the inference efficiency. From the experimental results in Table 2, we observe that models trained under our mutual learning framework achieve better overall performance than the previously published SOTA methods, even when limited to 10 passages only. This validates our assumption that it is possible to obtain a strong combination of the retriever and the knowledge selector, without requiring the supervision of annotated pairs of questions and passages.

Improvement in Inference Efficiency We quantify how much inference efficiency improves in our proposed framework when compared with the original FiD model requiring a large number of support passages (n=100). From Figure 2, we can find that for the NQ dataset, when the number of retrieved passages increases from 1 to 10, the performance gains increase accordingly; however, when we continue to increase the number of retrieved passages, the increase in the exact match value begins to plateau. A similar trend has also been observed in both TQA and WebQ datasets (i.e., a significant performance gain when increasing the number of the retrieved passages from 1 to 5, followed by a trivial improvement when increasing the number of retrieved passages beyond this). From this, we make the following three conclusions:

1. Once the number of support passages is sufficient to provide the reader enough evidence to generate the correct answer, increasing the number of passages does not necessarily improve model performance.
2. Our proposed model outperforming the original FiD model highlights that excessive external knowledge might distract the reader from giving correct answers.
3. Crucially, as demonstrated in Figure 2 with the red dotted line, our framework requires only 5 support passages to achieve comparable performance to with FiD models that use 100 support passages, while requiring significantly fewer FLOPs.

3.3 Ablation Study

In this section, we conduct ablation studies to answer the following four questions:

- Is the two-phase mutual training necessary?
- What is the significance of the policy-gradient method?
- How does the choice of different retrievers impact the results?
- What is the impact of employing different pretrained language models for the knowledge selector?

Is the Two-phase Mutual Training Necessary? In this paper, we present a novel mutual training strategy aimed at optimizing the parameters of both the knowledge selector and the reader through an alternating process. We hypothesize that the knowledge selector's primary function is to discern the most pertinent and valuable passages, while the reader's objective is to generate precise answers based on the selections made by the knowledge selector. Consequently, these two components should be fine-tuned in a collaborative manner. To further substantiate our hypothesis, we conducted an ablation study where we kept the parameters of a pre-trained reader fixed,

Model	NQ		TQA		WebQ	
	K=10	K=100	K=10	K=100	K=10	K=100
DPR (Karpukhin et al., 2020a)	–	41.5	–	57.9	–	41.1
ColBERT-QA (Khattab et al., 2021)	–	48.2	–	63.2	–	–
ORQA (Lee et al., 2019)	–	33.3	–	45.0	–	36.4
RAG-Token (Lewis et al., 2020b)	–	44.1	–	55.2	–	45.5
RAG-Seq (Lewis et al., 2020b)	–	44.5	–	56.8	–	45.2
REALM$_{wiki}$ (Guu et al., 2020)	–	39.2	–	–	–	40.2
REALM$_{news}$ (Guu et al., 2020)	–	40.4	–	–	–	40.7
FiD (T5 base) (Izacard and Grave, 2020)	42.3	48.2	61.1	65.0	45.2	47.2
FiD (T5 large) (Izacard and Grave, 2020)	45.6	51.4	63.2	67.6	47.1	50.5
FiD-KD (T5 base) (Izacard and Grave, 2021)	49.2	50.1	68.7	69.3	49.2	51.2
FiD-KD (T5 large) (Izacard and Grave, 2021)	52.7	54.4	72.5	72.5	49.8	52.7
Ours (T5 base)	52.1	–	69.8	–	52.5	–
Ours (T5 large)	**56.1**	–	**74.1**	–	**53.7**	–

Table 2: EM scores of prior state-of-the-art models and our models on NQ, TQA, and WebQ. Note that this work aims at reducing the number of retrieved passages without compromising the model's performance, so we do not report experimental results ($K = 100$) of our method because it means that the knowledge selector is not needed.

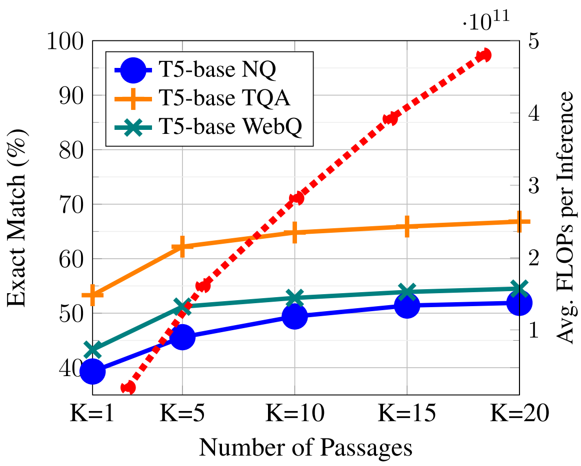

Figure 2: Accuracy-cost curves of the proposed system for different K on NQ, TQA, and WebQS, respectively. The dotted red line represents the average FLOPs for an inference under different numbers of passages.

focusing solely on the optimization of the knowledge selector using the policy gradient method. The results of this experiment are depicted in Figure 3.

We conducted experiments using three trained readers[8] to assess whether updating only the parameters of the knowledge selector (referred to as 'One-phase', as illustrated in Figure 3) is sufficient. It is important to note that in the one-phase setting, all training hyperparameters, including the base retriever, remain the same as in the two-phase setting, with the exception of eliminating the second phase. The results unequivocally demonstrate that two-phase training consistently yields superior performance, with improvements of 3.7% ↑, 5.6% ↑, and 5.7% ↑, respectively, compared to the one-phase setting. This ablation study results provide additional validation of the efficacy of our mutual two-phase training strategy.

Policy-gradient vs Supervised training In order to train the knowledge selector through supervised learning, it is necessary to have pairs of questions and their corresponding passages that contain relevant information. However, manually creating labeled data can be a time-consuming process, resulting in a lack of annotations for many datasets and applications. An alternative method is to utilize heuristics or weakly supervised learning, for example, by designating all documents containing the answer as positive samples. Thus, to assess the viability of this intuitive alternative approach, we employ it to construct a training dataset for knowledge selector training, referred to as the *supervised approach*. Using these ''ground truth'' labels, we can directly train the knowledge selector in a supervised manner.

[8] FiD-KD models are initialized using the T5-small and T5-large pretrained models, respectively, while RAG-Token model is initialized with the BART-large pretrained model.

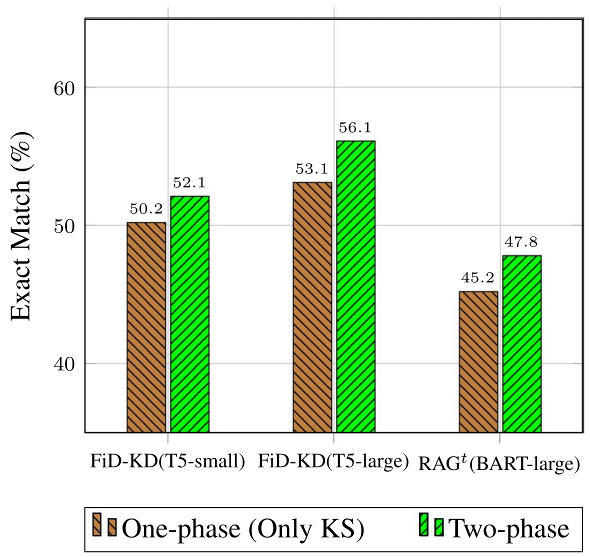

Figure 3: Results of employing one-phase and two-phase training with various trained readers on the NQ dataset. RAG^t denotes the RAG-Token Model. The number of retrieved passages is 10. In One-phase, the reader is initialized with the relevant pre-trained model, and its parameters remain fixed while we employ RL to optimize the parameters of the knowledge selector. In Two-phase, both the parameters of the initialized reader and the knowledge selectors are updated alternatively.

	NQ	TQA	WebQ
Supervised (T5-based, *K*=5)	46.1	59.2	44.8
Policy-gradient (T5-base, *K*=5)	49.8	63.1	47.9
Supervised (T5-base, *K*=10)	50.2	64.2	49.2
Policy gradient (T5-base, *K*=10)	**52.1**	**69.8**	**52.5**

Table 3: Comparsion results of the policy-gradient based framework and the supervised approach.

Table 3 shows the experimental results of the two approaches under two different numbers (5 and 10) of passages. We observe that our policy-gradient-based method performs much better than the supervised learning in both settings. Two possible reasons are: 1) Frequent answers or entities might lead to false-positive examples. For example, we can consider the question ''which university did Barack Obama obtain graduate from?'' alongside the passage ''...Barack Obama gave a speech in Harvard University... '', which would be considered a positive example as it contains the answer ''Harvard''. In this case, the supervised training approach might suffer from the false-positive labels during the training stage. For our proposed framework, although it is also possible for the 0-1 reward to give a false-positive signal when an irrelevant document included by the knowledge selector. However, as the policy is optimized it will naturally perform credit assignment and lower the value of irrelevant documents, since they may be excluded without lowering performance. As such, an optimal policy will not select irrelevant documents in the place of ones that would improve answer quality, since this action would have lower expected return. In contrast, the supervised approach will never learn to exclude false positives, as its objective is defined by the static labelling method. 2) A second limitation is that for some tasks, such as fact checking or long-form question answering, such heuristics might not be directly applicable.

	NQ	TQA	WebQ
BM25 + Reader	44.2	59.8	45.2
DPR + Reader	45.6	63.2	47.1
BM25+KS + Reader	52.4	68.4	50.1
DPR+KS + Reader	**56.1**	**74.1**	**53.7**

Table 4: EM scores (%) of four different *retrieve-then-read* schemes over three benchmark datasets. The reader is initialized with the T5 large model. Note that both KS and Reader were trained using the proposed method using DPR as the retriever, and the same models are used for all rows in the table.

Impacts of Using Different Retrievers Our framework maintains a versatile approach by remaining agnostic to the choice of the off-the-shelf retriever. In this context, we maintain the trained knowledge selector and the reader as constants while experimenting with various retrieval methods. The results are presented in Table 4.

Firstly, it is evident that the inclusion of the knowledge selector results in improved performance compared to counterparts that do not utilize this feature, showcasing an increase of 4.5% and 5.6%. This underlines the effectiveness of the knowledge selector in identifying more relevant passages, thereby enabling the reader model to generate accurate responses. Notably, these findings are consistent with those presented in Table 1, where the employment of the knowledge selector yields higher R@10 scores.

	NQ	TQA	WebQ
BERT-base (110M)	52.1	69.8	52.5
BERT-large (330M)	52.8	69.8	**53.0**
ALBERT (235M)	52.2	69.7	52.6
RoBERTa-base (125M)	52.0	69.7	52.1
RoBERTa-large (355M)	**52.9**	**69.9**	52.9

Table 5: Performance of different pretrained language models (parameter sizes are shown in the bracket) are used in the knowledge selector where the reader we use is T5-base and the number of passages is 10.

Exploration of Different Pretrained Language Models for the Knowledge Selector In our previous experiments, the knowledge selector is built on the BERT-base with its parameters fixed. In this part, we explore whether the knowledge selector can benefit from other pretrained language models with different parameter sizes.

From Table 5, we observe that there is no significant improvement on three benchmark datasets when using alternative pretrained language models of different sizes. For example, there is only a 0.7 increase in the EM score when we replace the 110M BERT-base model with 330M BERT-large. This suggests that using BERT-base is large enough to learn the relationship between the question and the passages under our mutual learning framework. In addition, one interesting phenomenon is that the EM score on the TQA dataset is almost unchanged for the chosen five different pretrained language models. One possible reason is that questions in TQA do not rely heavily on external knowledge, namely, many questions could be answered based on the parameters of the pretrained language models.

4 Zero-shot Transfer

Previous experimental results showed that our mutual learning framework could improve the model performance in the supervised fine-tuning setting. Here, we evaluate whether the trained *knowledge selector* module can also contribute to improving the generation performance of large-scale language models (LLMs) (e.g., GPT-3 and ChatGPT) in a zero-shot setting. In particular, we explore three different settings: 1) *without retrieval* means that we feed the question to LLMs directly without concatenating any other background knowledge; 2) *with **ONE** retrieved passage* denotes that we concatenate a passage retrieved by different methods to the question following the same prompt as P3 (Bach et al., 2022); 3) similarly, *with **TWO** retrieved passages* denotes retrieving two passages. All experimental results are reported in Table 6. Note that due to the length limitation, we only explore the settings of using one retrieved passages and two retrieved passages.

Methods	NQ	TQA	WebQS
**without retrieval*			
GPT-3	14.6	64.3	14.4
ChatGPT	20.9	67.5	18.6
with **ONE retrieved passage*			
GPT-3 (DPR, n=1)	22.4	67.9	34.5
GPT-3 (Ours, n=1)	24.2	69.3	36.1
ChatGPT (DPR, n=1)	24.8	70.5	36.2
ChatGPT (Ours, n=1)	26.1	72.1	37.8
with **TWO retrieved passage*			
GPT-3 (DPR, n=2)	26.1	69.2	36.4
GPT-3 (Ours, n=2)	28.9	71.8	39.8
ChatGPT (DPR, n=2)	29.2	71.3	40.9
ChatGPT (Ours, n=2)	**32.1**	**73.2**	**42.3**

Table 6: Experimental results of using GPT-3 and ChatGPT with one and two retrieved results. The prompt we used is from P3 (Bach et al., 2022) of the form *Refer to the passage below and answer the following question. Passage:* {*passages*} *Question:* {*question*}, where {*question*} and {*passages*} are replaced by the corresponding question and the retrieved passages.

From Table 6, we observe that adding the retrieved passage(s) to the question as the input to LLMs could obviously improve the generation information in both GPT-3 and ChatGPT. A similar phenomenon has also been noticed in Yu et al. (2022b). Besides, under the same number of retrieved passages, passages selected by our trained *knowledge selector* contribute more to the generation performance, as reflected from the exact match scores. To some extent, this demonstrates that the *knowledge selector* trained using our mutual learning framework is not model-specific, and can be used as a standalone tool for retrieving relevant passages in other frameworks.

Original question	**FiD-with-DPR's prediction**	**Our prediction**
Q: Who got the first nobel prize in physics?	Albert A. Michelson ✗	Wilhelm Conrad Röntgen ✓
Top-3 passages ranked by DPR:		
1. Albert A. Michelson was an American physicist known for his work on measuring the speed of light . . . In 1907 he received the Nobel Prize in Physics, becoming the first American to win the Nobel Prize . . .		
2. Nobel Prize in Physics The Nobel Prize in Physics is a yearly award given by the Royal Swedish Academy . . .		
3. The discovery of X-rays by physicist Wilhelm Conrad Röntgen, first winner of the Nobel Prize for Physics . . .		
Top-3 passages ranked by our method:		
1. Wilhelm Conrad Röntgen was a German mechanical engineer and physicist, who, on 8 November 1895 . . . range known as X-rays rays . . . an achievement that earned him the first Nobel Prize in Physics in 1901 . . .		
2. The discovery of X-rays by physicist Wilhelm Conrad Röntgen, first winner of the Nobel Prize for Physics . . .		
3. . . . when German physics professor Wilhelm Conrad Röntgen discovered the X-ray and noted that, while it could pass through human tissue . . . He received the first Nobel Prize in Physics for his discovery . . .		
Q: Who is the president of USA right now?	George W. Bush ✗	Barack Obama ✗
Top-3 passages ranked by DPR:		
1. . . . on January 20, 2009, when Barack Obama was inaugurated as the 44th President of the United States . . .		
2. . . . Donald Trump was formally elected by the Electoral College on December 19, 2016 . . .		
3. . . . January 20, 2001, when George W. Bush was inaugurated as the 43rd President of the United States . . .		
Top-3 passages ranked by our method:		
1. . . . on January 20, 2009, when Barack Obama was inaugurated as the 44th President of the United States . . .		
2. Barack Obama, a Democrat and former U.S. Senator from Illinois, was first elected president . . .		
3. Barack Obama is an American attorney and politician who served as the 44th President of U.S. . . .		

Table 7: Case study of retrieved documents and predicted results from FiD-with-DPR (Izacard and Grave, 2021) and our proposed framework. For the space limitation, we only illustrate the snapshots of the top three out of the ten retrieved Wiki passages from the two different approaches, specifically.

5 Case Study

To better understand why our proposed framework can help improve the predictive performance, we manually pick two representative examples as case studies. Examples where predicted results of our framework and a strong baseline (FiD-with-DPR) together with part of their used passages are in Table 7. Note that for both approaches, we set the number of retrieved passages as 10 for a fair comparison while we only showcase top threes retrieved passages due to the space limitation.

In the first case, we can observe that among the three top passages ranked by DPR, only one is relevant to the question and can provide evidence to generate the correct answer while the other two passages are either off-topic or even providing some incorrect information. For example, the top-1 retrieved passage conveys a seemingly relevant information about the first American winner of the Nobel Prize for physics, which is considered as a negative factor of leading the reader to generate an incorrect prediction with respect the given question without emphasizing the winner's nationality. In contrast, in terms of the relevance to the given question, we can notice that all the three passages from our method are talking about Wilhelm Conrad Röntgen, based on which the reader correctly gives the answer as we expect. We conjecture that the reader might be negatively distracted by irrelevant knowledge, thus making an incorrect predictions with respect to the given question.

In the second case, while the comparison between the two predictions with the ground truth answer (Donald Trump) is incorrect, the prediction itself should be considered as a correct answer for the question due to the time-dependent property of the question. According to Zhang and Choi (2021), the Natural Questions dataset contains a

Query: Who is the girl in green day 21 guns?	**Ground truth Answer:** Lisa Stelly
ChatGPT [No Passage]: Lauren German ✗	
With top-1 passage by DPR: 21 Guns is a song by American punk rock band Green Day. It was released as the second single from their eighth album . . .	
ChatGPT: Lauren German ✗	
With top-1 passage by our method: The 21 guns music video takes place with the band and the album's two protagonists Christian (Josh Boswell) and Gloria (Lisa Stelly) taking refuge . . .	
ChatGPT: Lisa Stelly ✓	

Table 8: Case study of predictions of ChatGPT w/o the top-1 passage from DPR or our method.

significant proportion, roughly 16.5%, of questions that have time-dependent answers. Another observation is that when compared to the baseline model, the retrieved passages from our approach are more consistent, all of which are related to Barack Obama, and we conjecture that such a bunch of topic-relevant passages might contribute more to the reader's generation.

Additionally, we give an example to show that for some knowledge-intensive tasks like open-domain question answering, providing some necessary context information relevant to the given question can bring some gains in improving the predictive performance for large and versatile language models like ChatGPT. One possible reason is that although the Wikipedia data have been seen during the training stage of ChatGPT, it is impossible to ''remember'' all training data in the form of their parameters. As shown in Table 8, with no contextual knowledge, ChatGPT gave an incorrect answer. However, when equipped with one passage containing the answer, ChatGPT can make a correct prediction. Hence, providing some necessary contextual information as a reference might help ChatGPT generate a correct prediction when meeting with some tough questions, thus indirectly showing the superiority of our trained knowledge selector over DPR.

6 Related Work

Open-domain Question Answering (ODQA) is an important task, aiming at providing precise answers in response to the user's questions in natural language. There are two common types of knowledge sources: One is unstructured textual documents available on the Internet, and another is a predefined structured data such as knowledge graphs which are often manually constructed. In this paper, we focus on the former, which is considered to be a more general and challenging task since available unstructured text to obtain answers are fairly common and easily accessible, such as Wikipedia, news articles and science books, etc.

Next, we review two categories of approaches widely explored in current textual based ODQA literature. We refer the reader to Zhu et al. (2021) for a more exhaustive introduction to this topic.

Retrieval-free LLM-based Domain Question Answering Systems Large language models show impressive performance on a wide range of tasks. Prior studies (Petroni et al., 2019; Roberts et al., 2020; Brown et al., 2020) have shown that a large amount of knowledge learned from large-scale textual data can be stored in the underlying parameters, and thus these models are capable of answering questions without access to any external knowledge. For example, ChatGPT is able to correctly generate the answer given only a natural language question. However, although large language models demonstrate impressive performance on zero-shot learning abilities, their performance still lags behind the supervised settings (Yu et al., 2022b). Besides, some prior studies (Izacard et al., 2022) also demonstrate that retrieval augmented language models can achieve better performance in knowledge-intensive tasks.

Retrieve-then-read Open Domain Question Answering According to a detailed survey (Yu et al., 2022b), modern ODQA architectures mainly follow the *retriever-then-read* paradigm as well as the specific techniques adopted in each of the components. Given a question, this model first leverages a retriever over a large evidence corpus to fetch a set of relevant documents that may contain the answer. A reader is then used to peruse the retrieved documents and predict an answer. In this paradigm, we observe that recent follow-up work has focused on improving either the retriever (Sachan et al., 2022; Qu et al., 2021) or the reader (Yu et al., 2022a; Wang et al., 2018; Min et al., 2019). In particular, it is noteworthy that the concept of integrating a reranker to enhance retrieval performance has been previously explored in RocketQAv2 (Ren et al., 2021). However, a key distinction lies in the approach: RocketQAv2 employs a joint training method for both the passage retriever and the reranker, whereas in our work, we fix the retriever's parameters. Instead, our focus is solely on updating the reranker's parameters, thus enabling our framework to consistently benefit advanced retriever models as they become available. However, to the best of our knowledge, only a few prior studies have been carried out on training both the retriever and the reader in an end-to-end mode. Lee et al. (2019) introduced the *inverse cloze task* for pre-training retrievers, which are then fine-tuned end-to-end on question-answering tasks. One most related to our work is that of Izacard and Grave (2021), which uses the internal attention scores from the reader as synthetic labels to train the retriever. In this work, we also explore the method of using the reader's feedback to optimize the retriever without additional supervision besides available pairs of question and answer.

7 Conclusion

In this work, we explore how to improve the prediction performance and inference cost of *reader* models in current open-domain question-answer architectures. To this end, we introduce a fine-grained *knowledge selector* into the *retrieve-then-read* paradigm, whose goal is to construct a small subset of passages which retain question-relevant information. The knowledge selector is trained as a component of our novel mutual learning framework, which iteratively trains the knowledge selector and the reader. We adopt a simple and novel approach employing policy gradients to optimize the knowledge selector, using feedback from the reader to train it to select a small and informative set of passages. This approach avoids brute-force search or manually designed heuristics, without requiring any annotated query-document pairs for supervision. We show that iteratively training the reader and the knowledge selector leads to better predictive performance on some public open-domain question answering benchmarks. Finally, our approach matches the accuracy of the top-performing Fusion-in-Decoder reader, whilst utilizing just 18.32% of its reader inference cost (FLOPs).

Acknowledgments

We are grateful to the action editor (Roi Reichart), and all anonymous reviewers for their insightful comments. Additionally, we would like to thank Chenglong Wang, Baolin Peng, and Michel Galley for their insightful discussions and support.

References

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M. Saiful Bari, Thibault Févry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. Promptsource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104. https://doi.org/10.18653/v1/2022.acl-demo.9

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *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 ACL*. https://doi.org/10.18653/v1/P17-1171

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. https://doi.org/10.18653/v1/2021.acl-long.240

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *CoRR*, abs/2112.09118.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*. https://doi.org/10.18653/v1/2021.eacl-main.74

Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

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

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*. https://doi.org/10.18653/v1/P17-1147

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*. https://doi.org/10.18653/v1/2020.emnlp-main.550

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020b. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020*, pages 6769–6781. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.550

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for openqa with colbert. *Transactions of the Association for Computational Linguistics*, 9:929–944. https://doi.org/10.1162/tacl_a_00405

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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*, 7:453–466. https://doi.org/10.1162/tacl_a_00276

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022a. Open-domain question answering via chain of reasoning over heterogeneous knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7–11, 2022*, pages 5360–5374. Association for Computational Linguistics. https://aclanthology.org/2022.findings-emnlp.392

Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022b. Open domain question answering with a unified knowledge interface. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1620. https://doi.org/10.18653/v1/2022.acl-long.113

Alejandro Martín, Javier Huertas-Tato, Álvaro Huertas-García, Guillermo Villar-Rodríguez, and David Camacho. 2022. Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference. *Knowledge-Based Systems*, 251:109265. https://doi.org/10.1016/j.knosys.2022.109265

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546. https://doi.org/10.18653/v1/2022.findings-naacl.115

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. Kilt: A benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544. https://doi.org/10.18653/v1/2021.naacl-main.200

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. https://doi.org/10.18653/v1/D19-1250

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*. https://doi.org/10.18653/v1/D16-1264

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. https://doi.org/10.18653/v1/2021.emnlp-main.224

Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535. https://doi.org/10.1090/S0002-9904-1952-09620-8

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426. https://doi.org/10.18653/v1/2020.emnlp-main.437

Stephen Robertson, and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389. https://doi.org/10.1561/1500000019

Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. 2022. Questions are all you need to train a dense passage retriever. *arXiv preprint arXiv:2206.10658*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31. https://doi.org/10.1609/aaai.v31i1.11164

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Neeraj Varshney, Man Luo, and Chitta Baral. 2022. Can open-domain qa reader utilize external knowledge efficiently like humans? *arXiv preprint arXiv:2211.12707*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R^3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of AAAI*. https://doi.org/10.1609/aaai.v32i1.12053

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of EMNLP-IJCNLP*. https://doi.org/10.18653/v1/D19-1599

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256. https://doi.org/10.1007/BF00992696

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. https://doi.org/10.18653/v1/N19-4013

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022a. Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974. https://doi.org/10.18653/v1/2022.acl-long.340

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022b. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Jun Zhang, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2021. Kers: A knowledge-enhanced framework for recommendation dialog systems with multiple subgoals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1092–1101. https://doi.org/10.18653/v1/2021.findings-emnlp.94

Michael Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387. https://doi.org/10.18653/v1/2021.emnlp-main.586

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.