Retrieval as Attention: End-to-end Learning of Retrieval and Reading within a Single Transformer

Zhengbao Jiang[♡,*] **Luyu Gao**[♡,*] **Jun Araki**[♢] **Haibo Ding**[♢,†]
Zhiruo Wang[♡] **Jamie Callan**[♡] **Graham Neubig**[♡]
[♡]Language Technologies Institute, Carnegie Mellon University
[♢]Bosch Research North America
{zhengbaj,luyug,zhiruow,callan,gneubig}@cs.cmu.edu
{jun.araki,haibo.ding}@us.bosch.com

Abstract

Systems for knowledge-intensive tasks such as open-domain question answering (QA) usually consist of two stages: efficient retrieval of relevant documents from a large corpus and detailed reading of the selected documents to generate answers. Retrievers and readers are usually modeled separately, which necessitates a cumbersome implementation and is hard to train and adapt in an end-to-end fashion. In this paper, we revisit this design and eschew the separate architecture and training in favor of a single Transformer that performs **Re**trieval as **Att**ention (**ReAtt**), and end-to-end training solely based on supervision from the end QA task. We demonstrate for the first time that a single model trained end-to-end can achieve both competitive retrieval and QA performance, matching or slightly outperforming state-of-the-art separately trained retrievers and readers. Moreover, end-to-end adaptation significantly boosts its performance on out-of-domain datasets in both supervised and unsupervised settings, making our model a simple and adaptable solution for knowledge-intensive tasks. Code and models are available at https://github.com/jzbjyb/ReAtt.

1 Introduction

Knowledge-intensive tasks such as question answering (QA), fact checking, and dialogue generation require models to gather relevant information from potentially enormous knowledge corpora (e.g., Wikipedia) and generate answers based on gathered evidence. A widely used solution is to first *retrieve* a small number of relevant documents from the corpus with a *bi-encoder* architecture which encodes queries and documents independently for efficiency purposes, then *read* the retrieved documents in a more careful and expansive way with a *cross-encoder* architecture which encodes queries and documents jointly (Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022). The distinction between retrieval and reading leads to the widely adopted paradigm of treating retrievers and readers separately. Retrievers and readers are usually two separate models with heterogeneous architectures and different training recipes, which is cumbersome to train. Even though two models can be combined in an ad-hoc way for downstream tasks, it hinders effective end-to-end learning and adaptation to new domains.

There have been several attempts to connect up reader and retriever training (Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020; Sachan et al., 2021; Lee et al., 2021a; Izacard et al., 2022). However, retrievers in these works are not learned in a fully end-to-end way. They require either initialization from existing supervisedly trained dense retrievers (Lewis et al., 2020), or expensive unsupervised retrieval pretraining as warm-up (Lee et al., 2019; Guu et al., 2020; Sachan et al., 2021; Lee et al., 2021a; Izacard et al., 2022). The reliance on retrieval-specific warm-up and the ad-hoc combination of retrievers and readers makes them less of a unified solution and potentially hinders their domain adaptation ability. With the ultimate goal of facilitating downstream tasks, retriever and reader should instead be fused more organically and learned in a fully end-to-end way.

In this paper, we focus on one of the most important knowledge-intensive tasks, open-domain QA. We ask the following question: is it possible to perform both retrieval and reading *within a single Transformer model*, and train the model in a *fully end-to-end* fashion to achieve competitive performance from both perspectives? Such a single-model end-to-end solution eliminates the need for retrieval-specific annotation and warm-up and simplifies retrieval-augmented training, making adaptation to new domains easier. Based on the analogy between self-attention which relates dif-

[*]The first two authors contributed equally.
[†]Haibo Ding is now at Amazon.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2336 - 2349
December 7-11, 2022

ferent tokens in a single sequence (Vaswani et al., 2017) and the goal of retrieval which is to relate queries with relevant documents, we hypothesize that self-attention could be a natural fit for retrieval, and it allows an organic fusion of retriever and reader within a single Transformer.

Specifically, we start from a encode-decoder T5 (Raffel et al., 2020) and use it as both retriever and reader. We use the first B encoder layers as bi-encoder to encode queries and documents independently, and the attention score at layer $B+1$ (denoted as *retrieval attention*) to compute relevance scores, as shown in Fig. 1. We found that directly using self-attention for retrieval underperforms strong retrievers, which we conjecture is because self-attention pretrained on local context is not sufficient to identify relevant information in the large representation space of the whole corpus. To solve this, we propose to compute retrieval attention between a query and a large number of documents and *adjust the retrieval attention across documents*. For each query, we compute retrieval attention over both close documents that potentially contain positive and hard negative documents, and documents of other queries in the same batch as random negatives. The retrieval attention is adjusted by minimizing its discrepancy from the cross-attention between the decoder and encoder (denoted as *target attention*), which is indicative of the usefulness of each document in generating answers (Izacard and Grave, 2021a). The resulting **Re**trieval as **Att**ention model (**ReAtt**) is a single T5 trained based on only QA annotations and simultaneously learns to promote useful documents through cross-document adjustment.

We train ReAtt on Natural Questions dataset (NQ) (Kwiatkowski et al., 2019) in a fully end-to-end manner. It achieves both competitive retrieval and QA performance, matching or slightly outperforming state-of-the-art retriever ColBERT-NQ (Khattab et al., 2020) trained with explicit retrieval annotations and strong QA model FiD (Izacard and Grave, 2021b,a), demonstrating for the first time end-to-end training can produce competitive retriever and reader within a single model. To further test ReAtt's generalization and end-to-end adaptation ability, we conduct zero-shot, supervised, and unsupervised adaptation experiments on 7 datasets from the BEIR benchmark (Thakur et al., 2021). In all settings, end-to-end adaptation improves the retrieval performance usually by a large margin, achieving comparable or superior performance to strong retrieval adaptation and pretraining methods.

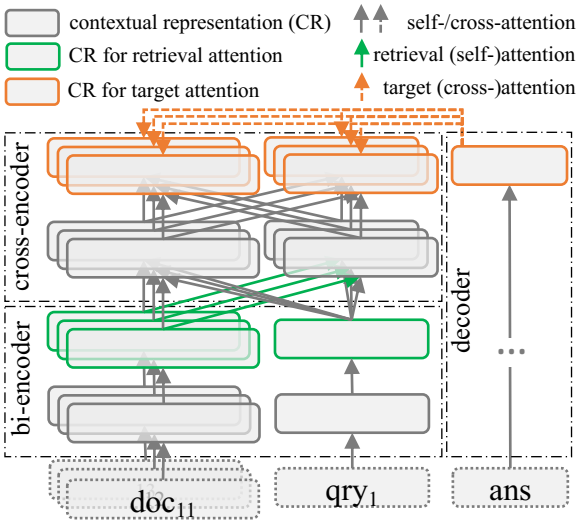

Figure 1: Illustration of Retrieval as Attention (ReAtt) with the first B=2 encoder layers as bi-encoder (i.e., retriever) and the rest L-B=2 layers as cross-encoder. During training, the retrieval attention between a query $\boldsymbol{q}_1$ and documents $\boldsymbol{d}_{11,12,13}$ is adjusted by minimizing its discrepancy from the target attention. For simplicity, we use a single arrow to represent attention of a single head between multiple tokens.

2 Retrieval as Attention (ReAtt)

With the goal of developing a single Transformer that can perform both retrieval and reading, and the analogy between retrieval and self-attention, we first introduce architecture changes to allow retrieval as attention (§ 2.2), then examine how well attention as-is can be directly used to perform retrieval (§ 2.3).

2.1 Formal Definition

We first briefly define the task of retrieval and question answering. As mentioned in the introduction, queries and documents need to be represented independently for efficient retrieval which implies a bi-encoder architecture that has no interaction between queries and documents. Without loss of generality, we use $E_{\boldsymbol{d}} = \text{biencoder}(\boldsymbol{d})$ to denote one or multiple representations generated by a bi-encoder based on a document from a corpus $\boldsymbol{d} \in \mathcal{D}$, and likewise $E_{\boldsymbol{q}} = \text{biencoder}(\boldsymbol{q})$ to denote query representations.[1] The top-k documents most relevant to a query are retrieved by $\mathcal{D}_{\boldsymbol{q}}^{\text{ret}} = \arg\text{topk}_{\boldsymbol{d}\in\mathcal{D}} r(E_{\boldsymbol{q}}, E_{\boldsymbol{d}})$, where function r

[1]Queries and documents can use different bi-encoders but we use one notation for simplicity.

computes relevance based on query and document representations which can be as simple as a dot product if queries and documents are encoded into a single vector, and $\mathcal{D}_{\boldsymbol{q}}^{\text{ret}}$ stands for the returned documents. We consider encoder-decoder-based generative question answering in this paper, which jointly represents queries and retrieved documents with the encoder $E_{\boldsymbol{q},\boldsymbol{d}} = \text{crossencoder}(\boldsymbol{q}, \boldsymbol{d})$, and generates the answer $\boldsymbol{a}$ autoregressively with the decoder $P^{\text{gen}}(\boldsymbol{a}|\boldsymbol{q}, \boldsymbol{d}) = P^{\text{gen}}(\boldsymbol{a}|E_{\boldsymbol{q},\boldsymbol{d}})$. To handle multiple retrieved documents, we follow the fusion-in-decoder model (FiD) (Izacard and Grave, 2021b) which encodes each query-document pair independently and fuse these representations in decoder through cross-attention $P^{\text{gen}}(\boldsymbol{a}|\boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}}^{\text{ret}}) = P^{\text{gen}}(\boldsymbol{a}|E_{\boldsymbol{q},\boldsymbol{d}_1}, ..., E_{\boldsymbol{q},\boldsymbol{d}_{|\mathcal{D}_{\boldsymbol{q}}^{\text{ret}}|}})$. Negative log likelihood (NLL) is used in optimization $\mathcal{L}_{\text{QA}} = -\log P^{\text{gen}}(\boldsymbol{a}|\boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}}^{\text{ret}})$.

2.2 Leveraging Attention for Retrieval

Next, we introduce our method that directly uses self-attention between queries and documents as retrieval scores.

Putting the Retriever into Transformers As illustrated in Fig. 1, we choose T5 (Raffel et al., 2020) as our base model, use the first B layers of the encoder as the *bi-encoder "retriever"* by disabling self-attention between queries and documents, and the remaining $L-B$ layers as the *cross-encoder "reader"*. We use the self-attention paid from query tokens to document tokens at the $B+1$-th layer as the retrieval score, which is denoted as *retrieval attention* (green arrows in Fig. 1). It is computed based on the independent query and document contextual representations from the last (B-th) layer of the bi-encoder (green blocks in Fig. 1). Formally for an H-head Transformer, document and query representations are:

$$E_{\boldsymbol{d}} = \{K_{\boldsymbol{d}}^{B+1,h} \in \mathbb{R}^{|\boldsymbol{d}|\times e}\}_{h=1}^{H},$$
$$E_{\boldsymbol{q}} = \{Q_{\boldsymbol{q}}^{B+1,h} \in \mathbb{R}^{|\boldsymbol{q}|\times e}\}_{h=1}^{H},$$

where K and Q are key and query vectors of the token sequence used in self-attention, $|\boldsymbol{d}|$ and $|\boldsymbol{q}|$ are document and query length, and e is the dimensionality of each head. The retrieval attention matrix from query tokens to document before softmax for one head is computed by:

$$A_{\boldsymbol{q},\boldsymbol{d}}^{B+1,h} = Q_{\boldsymbol{q}}^{B+1,h} \times K_{\boldsymbol{d}}^{B+1,h^T} \in \mathbb{R}^{|\boldsymbol{q}|\times|\boldsymbol{d}|}.$$

Directly using attention for retrieval can not only leverage its ability to identify relatedness, it is also a natural and simple way to achieve both retrieval and reading in a single Transformer with minimal architectural changes, which facilitates our final goal of end-to-end learning.

From Token Attention to Document Relevance Given the token-level attention scores $A_{\boldsymbol{q},\boldsymbol{d}}^{B+1,h}$, the relevance between $\boldsymbol{q}$ and $\boldsymbol{d}$ is computed by avg-max aggregation: choosing the most relevant document token for each query token (i.e., max) then averaging across query tokens:

$$r_h(\boldsymbol{q}, \boldsymbol{d}) = \text{avg}_0\big(\text{max}_1(A_{\boldsymbol{q},\boldsymbol{d}}^{B+1,h})\big), \qquad (1)$$

where 1 and 0 refer to the dimension over which the operation is applied. This is similar to the MaxSim and sum operators used in ColBERT (Khattab and Zaharia, 2020), with the intuition that a relevant document should match as many query tokens as possible with the best-matching token. The final relevance is a weighted sum over all heads:

$$r(\boldsymbol{q}, \boldsymbol{d}) = \sum_{h=1}^{H} P_h^{\text{head}} \cdot r_h(\boldsymbol{q}, \boldsymbol{d}),$$

where P_h is a learnable weight that sums to one. As explained in the next section, we empirically find only a few attention heads with non-random retrieval performance, and among them one particular head is significantly better than the others. Given this observation, we introduce a low temperature τ to promote this sparsity $P_h^{\text{head}} = \frac{\exp(w_h/\tau)}{\sum_{h'}\exp(w_{h'}/\tau)}$, which always ends with *a single head* with the great majority of the weight, which is denoted as *retrieval head* h^*. As a result, the learned head weights are practically a head selector, a fact that can also be exploited to make test-time retrieval more efficient.

End-to-end Retrieval with Attention To perform retrieval over a corpus, we first generate key vectors $K_{\boldsymbol{d}}^{B+1,h^*}$ of retrieval head for all document tokens offline and index them with FAISS library (Johnson et al., 2021). For each query token, we issue its vector ($Q_{\boldsymbol{q}}^{B+1,h^*}$) to the index to retrieve top-K' document tokens, which yields a filtered set of documents, each of which has at least one token retrieved by a query token. We then fetch all tokens of filtered documents, compute relevance scores following Eq. 1, and return top-K documents with the highest scores $r_{h^*}(\boldsymbol{q}, \boldsymbol{d})$. This is similar to the two-stage retrieval in ColBERT (Khattab and Zaharia, 2020), and we reuse their successful practice

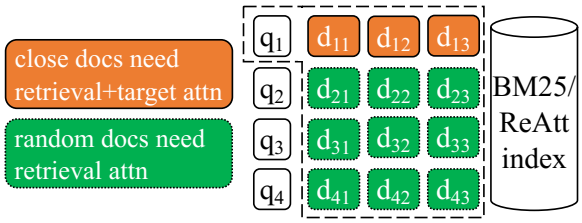

Figure 2: Illustration of approximate attention over the corpus with $|\mathcal{Q}|$=4 queries in a batch and K=3 close documents per query. We use $\boldsymbol{q}_1$ as an example to illustrate the required computation, where close documents require both retrieval and target attention while random documents only require retrieval attention.

in index compression and search approximation to make test-time retrieval efficient, which we refer to Santhanam et al. (2021) for details.

2.3 How Good is Attention As-is?

To examine this question, we use T5-large and test queries from the Natural Question dataset (NQ), retrieve 100 documents with BM25, compute relevance scores $r_h(\boldsymbol{q}, \boldsymbol{d})$ with half layers ($B = 12$) as bi-encoder, and measure its correlation with the gold binary annotation. We found that among $H = 24$ heads, 4 heads have non-trivial correlations of 0.137, 0.097, 0.082, and 0.059. We further perform end-to-end retrieval over Wikipedia using the best head, achieving top-10 retrieval accuracy of 43.5%, inferior to 55.5% of BM25. This demonstrates that there are indeed heads that can relate queries with relevant documents, but they are not competitive. We hypothesize that because self-attention is usually trained by comparing and relating tokens in a local context (512/1024 tokens) it cannot effectively identify relevant tokens in the enormous representation space of a corpus with millions of documents. This discrepancy motivates us to compute retrieval attention between queries and potentially all documents (i.e., attention over the corpus), and *adjust attention across documents to promote useful ones*.

3 Learning Retrieval as Attention

We first approximate attention over the corpus at training time by sub-sampling a manageable number of documents for each query containing both potentially relevant and random documents (§ 3.1). Next, we introduce our end-to-end training objective that optimizes a standard QA loss while also adding supervision to promote attention over documents that are useful for the end task (§ 3.2).

3.1 Approximate Attention over the Corpus

Encoding the entire corpus and computing attention between the query and all documents is very expensive. To make it practical, we propose to subsample a small set of documents for each query to approximate the whole corpus. Inspired by negative sampling methods used in dense retriever training (Karpukhin et al., 2020; Xiong et al., 2021; Khattab and Zaharia, 2020), we sub-sample both (1) documents close to queries that can be either relevant or hard negatives, and (2) random documents that are most likely to be easy negatives. This allows the model to distinguish between relevant and hard negative documents, while simultaneously preventing it from losing its ability to distinguish easy negatives, which form the majority of the corpus.

Iterative Close Document Sub-sampling To sample documents close to a query $\mathcal{D}_{\boldsymbol{q}}^{\text{close}}$, we start from widely used lexical retriever BM25 (Robertson and Zaragoza, 2009) to retrieve $K = 100$ documents, as shown by the orange blocks in Fig. 2. We set K to a relatively large number to better approximate the local region, inspired by Izacard and Grave (2021b)'s findings that QA performance increases as more documents are used.

This fixed set of close documents can become outdated and no longer close to the query anymore as the retrieval attention gets better. To provide dynamic close sub-samples, we re-index the corpus and retrieve a new set of K documents using the current retrieval attention after each iteration. It is similar in spirit to the hard negative mining methods used in Karpukhin et al. (2020); Khattab et al. (2020), with a major difference that we do not manually or heuristically annotate documents but instead learn from the end loss with cross-document adjustment, which will be explained in § 3.2.

In-batch Random Document Sub-sampling We use close documents of other queries in the same batch as the random documents of the current query $\mathcal{D}_{\boldsymbol{q}}^{\text{random}} = \cup_{\boldsymbol{q}' \in \mathcal{Q} \wedge \boldsymbol{q}' \neq \boldsymbol{q}} \mathcal{D}_{\boldsymbol{q}'}^{\text{close}}$ where $\mathcal{Q}$ contains all queries in a batch, as shown by the green blocks in Fig. 2, which has the advantage of reusing document representations across queries. This is similar to the in-batch negatives used in DPR (Karpukhin et al., 2020) with a major difference that we reuse a token representations ($K_{\boldsymbol{d}}^{B+1,h}, 1 \leq h \leq H$) across queries instead of a single-vector document representation.

3.2 Cross-document Adjustment with Decoder-to-Encoder Attention Distillation

Given the sub-sampled $|\mathcal{Q}| \times K$ documents $\mathcal{D}_{\boldsymbol{q}} = \mathcal{D}_{\boldsymbol{q}}^{\text{close}} \cup \mathcal{D}_{\boldsymbol{q}}^{\text{random}}$ for each query $\boldsymbol{q}$, we compute the retrieval attention-based relevance scores $r(\boldsymbol{q}, \boldsymbol{d})$ and adjust them across multiple documents $\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{q}}$ only relying on end task supervision. Since retrieval is simply a means to achieve the downstream task, documents useful for downstream tasks should be promoted by retrieval. Inspired by reader-to-retriever distillation (Izacard and Grave, 2021a; Yang and Seo, 2020), we measure document usefulness based on cross-attention between decoder and encoder, and minimize retrieval attention's discrepancy from it through distillation. In contrast to Izacard and Grave (2021a) that learns two models iteratively and alternatively, we optimize QA and distillation loss in a single model simultaneously.

Minimizing KL-divergence Between Retrieval and Target Attention Specifically, we denote cross-attention before softmax of the first position/token of the last decoder layer as *target attention* $C_{\boldsymbol{a},\boldsymbol{q},\mathcal{D}_{\boldsymbol{q}}} \in \mathbb{R}^{H \times |\mathcal{D}_{\boldsymbol{q}}| \times (|\boldsymbol{d}|+|\boldsymbol{q}|)}$ where $\boldsymbol{a}$ is the answer, $|\mathcal{D}_{\boldsymbol{q}}|$ is the number of sub-sampled documents to be fused by the decoder (§ 2.1), and $|\boldsymbol{d}|$ is document length.[2] To aggregate token-level target attention into document-level distribution $P^{\text{tgt}}(\boldsymbol{a}, \boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}}) \in \mathbb{R}^{|\mathcal{D}_{\boldsymbol{q}}|}$, we first perform softmax over all tokens in all query-document pairs ($|\mathcal{D}_{\boldsymbol{q}}| \times (|\boldsymbol{d}| + |\boldsymbol{q}|)$), sum over tokens of each query-document pair ($|\boldsymbol{d}| + |\boldsymbol{q}|$), then average across multiple heads (H):

$$P^{\text{tgt}}(\boldsymbol{a}, \boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}}) = \text{avg}_0\Big(\text{sum}_2\big(\text{softmax}_{1,2}(C_{\boldsymbol{a},\boldsymbol{q},\mathcal{D}_{\boldsymbol{q}}})\big)\Big).$$

Given relevance scores obtained from retrieval attention, the final cross-document adjustment loss is the KL-divergence between relevance distribution P^{ret} and target distribution P^{tgt}:

$$\begin{aligned} P^{\text{ret}}(\boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}}) &= \text{softmax}\big(r(\boldsymbol{q}, \boldsymbol{d}_1), ..., r(\boldsymbol{q}, \boldsymbol{d}_{|\mathcal{D}_{\boldsymbol{q}}|})\big). \\ \mathcal{L}_{\text{cross-doc}} &= \text{KL}\Big(\overline{P^{\text{tgt}}(\boldsymbol{a}, \boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}})} \Big\| P^{\text{ret}}(\boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}})\Big), \end{aligned} \tag{2}$$

where the overline indicates stop gradient back propagation to target distributions. Our final loss combines QA loss and cross-document adjustment loss with α as combination weight.

$$\mathcal{L} = \mathcal{L}_{\text{QA}} + \alpha \cdot \mathcal{L}_{\text{cross-doc}}. \tag{3}$$

[2] We also attempted other variations of target attention and found performances are similar, consistent with observations in Izacard and Grave (2021a).

Zero Target Attention for Random Documents For a batch with $|\mathcal{Q}|$ queries, we need to compute retrieval attention and target attention between $|\mathcal{Q}| \times |\mathcal{Q}| \times K$ query-document pairs. This is both computation- and memory-intensive when batch size is large, especially for target attention because it requires $L - B$ layers of joint encoding of query-document pairs in the cross-encoder. To alleviate this, we make a simple and effective assumption that in-batch random documents are not relevant to the current query thus having zero target attention: $P^{\text{tgt}}(\boldsymbol{a}, \boldsymbol{q}, \mathcal{D}_{\boldsymbol{q}}^{\text{random}}) \in \mathbb{R}^{|\mathcal{D}_{\boldsymbol{q}}^{\text{random}}|} \leftarrow 0$. As a result, we only need to run cross-encoder and decoder for K close documents of each query, as shown in Fig. 2. In Appendix A we will introduce our efficient implementation to make it possible to run a large batch size over a limited number of GPUs.

3.3 Domain Adaptation Methods

One of the major benefits of a single end-to-end trainable model is that given a new corpus from a new domain, possibly without retrieval annotations, we can easily adapt it by end-to-end training. This section describes how we adapt ReAtt under different setups.

We consider adapting ReAtt with (1) QA supervision, (2) information retrieval (IR) supervision, or (3) unsupervised adaptation where we only have access to the document corpus. Although our goal is to learn retrieval through downstream tasks instead of retrieval supervision, being able to consume retrieval annotations is helpful when retrieval supervision is indeed available. To do so, we convert retrieval task with annotations in the form of query-document-relevance triples $\langle \boldsymbol{q}, \boldsymbol{d}, l \rangle$ into a generative task: given a query, the target is to generate *its relevant document and the corresponding relevance* with the following format "relevance: l. $\boldsymbol{d}$". If a query has multiple relevant documents, we follow Izacard and Grave (2021b) to randomly sample one of them. For unsupervised adaptation, with simplicity as our primary goal, we randomly choose one sentence from a document and mask one entity, which is considered as the "query", and have our model generate the masked entity as the "answer", similar to salient span masking (SSM) used in Guu et al. (2020).

4 In-domain Experiments

In this section, we examine if supervisedly training ReAtt end-to-end with *only* QA supervision yields both competitive retrieval and QA performance.

Datasets, Baselines, and Metrics We train our model using the Natural Questions dataset (NQ). We compare retrieval performance with lexical models BM25 (Robertson and Zaragoza, 2009), passage-level dense retrievers DPR, ANCE, coCondenser, FiD-KD, YONO (with and without retrieval pretraining) (Karpukhin et al., 2020; Oguz et al., 2021; Xiong et al., 2021; Gao and Callan, 2022; Izacard and Grave, 2021a; Lee et al., 2021a), and token/phrase-level dense retrievers DensePhrase, ColBERT, ColBERT-NQ (Lee et al., 2021b; Khattab and Zaharia, 2020; Khattab et al., 2020).[3] Among them ColBERT-NQ, FiD-KD and YONO are the most fair-to-compare baselines because of either similar token-level retrieval granularity (ColBERT-NQ) or similar end-to-end training settings (FiD-KD and YONO). We report top-k retrieval accuracy (R@k), the fraction of queries with at least one retrieved document containing answers. We compare QA performance with ORQA, REALM, RAG, FiD, EMDR^2, YONO, UnitedQA, and R2-D2 (Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021b,a; Sachan et al., 2021; Lee et al., 2021a; Cheng et al., 2021; Fajcik et al., 2021) using exact match (EM), among which FiD, EMDR^2, and YONO are the most fair-to-compare baselines because they have similar model sizes and training settings.

5 Implementation Details of ReAtt

ReAtt is based on T5-large with $B = 12$ encoder layers as bi-encoder and temperatures $\tau = 0.001$ to select the best retrieval head. We retrieve $K = 100$ close documents for each query, and use a batch size of $|\mathcal{Q}| = 64$ queries to obtain in-batch random documents. We use $\alpha = 8$ to combine cross-document adjustment loss with QA loss. We use AdamW with a learning rate of 5e-5, 10% steps of warmup, and linear decay. We first warmup cross-attention's ability to distinguish documents by only using the QA loss for 3K steps, then train with the combined losses (Eq. 3) for 4 iterations, where the first iteration uses close documents returned by BM25, and the following 3 iterations use close documents returned by the previous ReAtt model (denoted as ReAtt $_{\text{BM25}}$). Each iteration has 8K update steps and takes $\sim$ 1.5 days on a single node with 8 $\times$ A100 GPUs with 80GB memory. Since DPR (Karpukhin et al., 2020) achieves stronger performance than BM25, training with close documents returned by DPR can potentially reduce training time. We experimented with training on close documents from DPR for a single iteration with 16K steps (denoted as ReAtt $_{\text{DPR}}$). Since both approaches achieve similar performance (Tab. 1 and Tab. 2) and ReAtt $_{\text{DPR}}$ is cheaper to train, we use it in other experimental settings.

[3] ColBERT is trained on MS MARCO, ColBERT-NQ is on NQ.

Models	R@1	R@5	R@20	R@100	#Params.
supervised retrievers					
BM25	23.9	45.9	63.8	78.9	-
DPR	45.9	68.1	80.0	85.9	220M
DPR$^{\text{new}}$	52.5	72.2	81.3	87.3	220M
DPR-PAQ	-	74.2	84.0	89.2	220M
ANCE	-	-	81.9	87.5	220M
coCondenser	-	75.8	84.3	89.0	220M
DensePhrase	51.1	69.9	78.7	-	330M
ColBERT	-	-	79.1	-	110M
ColBERT-NQ	54.3	75.7	85.6	90.0	110M
semi/unsupervised retrievers					
FiD-KD	49.4	73.8	84.3	89.3	220M
YONO$_{\text{w/o PT}}$	-	-	72.3	82.2	165M
YONO$_{\text{w/ PT}}$	-	75.3	85.2	90.2	165M
ReAtt $_{\text{DPR}}$	54.6	77.2	**86.1**	**90.7**	165M
ReAtt $_{\text{BM25}}$	**55.8**	**77.4**	86.0	90.4	165M

Table 1: Retrieval performance on NQ. PT is retrieval pretraining. Fair-to-compare baselines are highlighted with background color. Best performance is in bold.

At test-time, we save key vectors of all tokens in the corpus and use exact index from FAISS (i.e., `faiss.IndexFlatIP`) to perform inner-product search. We retrieve $K' = 2048$ document tokens for each query token and return top-100 documents with the highest aggregated scores (Eq. 1) to generate answers. We found compressing index with clustering and quantization proposed by Santhanam et al. (2021) can greatly reduce search latency and index size with a minor retrieval accuracy loss.

5.1 Overall Results

We compare ReAtt with various retrievers and readers in Tab. 1 and Tab. 2. ReAtt achieves both slightly better retrieval performance than the strongest retriever baseline ColBERT-NQ (Khattab et al., 2020) and comparable QA performance than the strong reader baseline FiD-KD (Izacard and Grave, 2021a) on NQ, demonstrating for the first time that fully end-to-end training using QA supervision can produce both competitive retrieval and QA performance. Compared to another single-model architecture YONO (Lee et al., 2021a), ReAtt offers better performance without cumbersome pretraining to warm-up retrieval.

Models	EM	#Params.
ORQA (Lee et al., 2019)	33.3	330M
REALM (Guu et al., 2020)	40.4	330M
RAG (Lewis et al., 2020)	44.5	220M
FiD (Izacard and Grave, 2021b)	51.4	990M
FiD-KD (Izacard and Grave, 2021a)	54.4	990M
EMDR^2 (Sachan et al., 2021)	52.5	440M
YONO$_{\text{w/o PT}}$ (Lee et al., 2021a)	42.4	440M
YONO$_{\text{w/ PT}}$ (Lee et al., 2021a)	53.2	440M
UnitedQA (Cheng et al., 2021)	54.7	1.870B
R2-D2 (Fajcik et al., 2021)	**55.9**	1.290B
ReAtt $_{\text{DPR}}$	54.0	770M
ReAtt $_{\text{BM25}}$	54.7	770M

Table 2: QA performance on NQ. PT is retrieval pre-training. Fair-to-compare baselines are highlighted. Best performance is in bold.

5.2 Ablations

We perform ablation experiments to understand the contribution of each component. Due to resource limitations, all ablations are trained with 2K steps per iteration. We use ReAtt trained with B=12 bi-encoder layers, $|\mathcal{Q}|$=16 batch size, and α=8 cross-document loss weight as the baseline, remove one component or modify one hyperparameter at a time to investigate its effect. As shown in Tab. 3, we found: 1. Only using QA loss without cross-document adjustment (#2) improves retrieval performance over the original T5 (#3), but cross-document adjustment is necessary to achieve further improvement (#1). 2. Iteratively retrieving close documents with the current model is helpful (#5 vs #1). 3. In-batch random documents are beneficial (#4 vs #1), and a larger batch size leads to larger improvements (#8-11). 4. A larger weight on cross-document adjustment loss improves retrieval performance but hurts QA performance, with 4$\sim$8 achieving a good trade-off (#12-15). 5. A small number of bi-encoder layers (#6) significantly hurts retrieval while a large number of layers (#7) significantly hurts QA, suggesting choosing equal numbers of layers in bi-encoder and cross-encoder.

#	Methods	R@1	R@5	R@20	R@100	EM
	ReAtt baseline with B=12, $\|\mathcal{Q}\|$=16, α=8					
1		41.9	68.8	82.5	88.9	46.3
	remove one component					
2	- cross-doc loss	21.7	49.0	71.5	83.5	46.0
3	- QA (=T5)	13.2	33.7	53.6	67.7	3.0
4	- in-batch	38.1	66.0	80.3	87.6	46.7
5	- iterative	41.2	68.3	82.0	88.4	45.0
	different #layers in bi-encoder B					
6	B=6	19.1	42.1	62.4	78.1	40.3
7	B=18	38.2	63.8	79.3	87.4	35.2
	different batch sizes $\mathcal{Q}$					
8	$\|\mathcal{Q}\|$=4	39.4	66.1	80.7	88.1	45.0
9	$\|\mathcal{Q}\|$=8	40.7	67.1	82.1	88.6	45.7
10	$\|\mathcal{Q}\|$=32	43.6	69.4	82.8	89.1	46.4
11	$\|\mathcal{Q}\|$=64	45.5	71.0	83.3	89.4	47.3
	different cross-doc loss weights α					
12	α=1	37.4	65.4	80.9	88.0	47.3
13	α=2	39.7	66.9	81.7	88.4	47.4
14	α=4	40.9	68.0	82.1	88.8	46.9
15	α=16	42.0	68.8	82.5	88.8	45.5

Table 3: Ablations by removing one component or changing one hyperparameter from the ReAtt baseline.

6 Out-of-domain Generalization and Adaptation

In this section, we examine both zero-shot retrieval performance on out-of-domain datasets and ReAtt's end-to-end adaptability in supervised (QA, IR) and unsupervised settings.

6.1 Datasets, Baselines, and Metrics

We choose 7 datasets from BEIR (Thakur et al., 2021), a benchmark covering diverse domains and tasks. On each dataset we compare ReAtt with different types of retrievers including BM25, DPR, and ColBERT. We consider 2 QA datasets (BioASQ and FiQA (Tsatsaronis et al., 2015; Maia et al., 2018)) and one IR dataset (MS MARCO (Nguyen et al., 2016)) to evaluate supervised adaptation capability, and 4 other datasets (CQADupStack, TREC-COVID, SCIDOCS, SciFact (Hoogeveen et al., 2015; Voorhees et al., 2020; Cohan et al., 2020; Wadden et al., 2020)) to evaluate unsupervised adaptation capability. Detailed statistics are listed in Tab. 8. We report nDCG@10 to measure retrieval performance and EM to measure QA performance. We group all baselines into three categories and denote them with different colors in the following tables:

- **Supervised adaptation models** are trained with downstream task supervision, including RAG trained on BioASQ, Contriever fine-tuned on FiQA, and docT5query, ANCE, ColBERT, and Contriever fine-tuned on MS MARCO (Nogueira and Lin, 2019; Xiong et al., 2021; Khattab and Zaharia, 2020; Izacard et al., 2021).
- **Unsupervised adaptation models** are trained on domain corpus in an unsupervised way such as contrastive learning or pseudo query generation, including SimCSE and TSDAE+GPL (Gao et al., 2021c; Wang et al., 2021a,b).
- **Pretraining models** are trained on corpora with-

Tasks	QA		Retrieval
Datasets	BioASQ	FiQA	MS MARCO
zero-shot performance			
BM25	68.1	23.6	22.8
DPR	14.1	11.2	17.7
ColBERT-NQ	65.5	23.8	**32.8**
ReAtt	**71.1**	**30.1**	32.3
additional training			
Contriever	-	32.9	docT5query 33.8
SimCSE	58.1	31.4	ANCE 38.8
TSDAE+GPL	61.6	34.4	ColBERT 40.1
Contriever$_{\text{w/ FT}}$	-	38.1	Contriever **40.7**
ReAtt	$_{+5.8}$**76.9**	$_{+8.5}$**38.6**	ReAtt $_{+7.6}$39.9

Table 4: nDCG@10 of zero-shot and supervised adaptation experiments on two QA and one IR datasets. We use colors to denote categories: pretraining, unsupervised adaptation, and supervised adaptation. Baselines comparable to ReAtt are highlighted with blue background color. We also show the improvement of ReAtt over zero-shot performance in subscript.

#	Ablations	nDCG@1	@5		EM	
1	RAG	14.6	13.0		1.3	
2	+ reader	14.6	13.0	-	27.5	26.2
3	+ qry enc (e2e)	0.0	0.0	-13.0	25.7	-1.9
4	+ doc/qry enc*	29.4	27.1	14.1	5.0	3.7
5	+ reader (pipe)	29.4	27.1	-	27.8	22.8
6	+ qry enc	23.3	23.2	-4.0	26.2	-1.6
7	T5	49.2	47.7		0.0	
8	+ e2e	75.2	73.5	25.7	44.4	44.4
9	ReAtt	72.8	70.1		17.2	
10	+ e2e	**77.4**	**75.4**	5.3	**47.2**	30.0

Table 5: RAG and ReAtt on BioASQ. Each indent indicates fine-tuning one more component than its parent with performance difference colored with green/red. * denotes fine-tuning conducted sequentially instead of jointly with the current component.

out direct exposure to the target domain, such as Contriever (Izacard et al., 2021) trained with contrastive learning on Wikipedia and CCNet.

We highlight baselines in the same category as ReAtt in the following tables since comparison between them is relatively fair. Details of adaptation of ReAtt can be found in Appendix B.

6.2 Experimental Results

Results of supervised and unsupervised adaptation are listed in Tab. 4, Tab. 5, and Tab. 6 respectively.

Zero-shot Generalization Ability As shown in Tab. 4 and Tab. 6, the zero-shot performance of ReAtt is significantly better than other zero-shot baselines on two QA datasets and one fact checking dataset (+3.0/+6.5/+4.5 on BioASQ/FiQA/SciFact than the second best), and overall comparable on the rest of datasets (-0.5/-0.6/-3.0/-1.0 on MS

Methods	CQA.	TRECC.	SCIDOCS	SciFact
zero-shot performance				
BM25	29.9	**65.6**	**15.8**	66.5
DPR	15.3	33.2	7.7	31.8
ANCE	29.6	65.4	12.2	50.7
ColBERT-NQ	**33.9**	48.9	15.6	65.3
ReAtt	33.3	62.6	14.8	**71.0**
additional training				
Contriever	34.5	59.6	**16.5**	67.7
SimCSE	29.0	68.3	-	55.0
TSDAE+GPL	35.1	74.6	-	68.9
ReAtt	$_{+3.3}$**36.6**	$_{+13.4}$**76.0**	$_{+1.0}$15.8	$_{+0.2}$**71.2**

Table 6: nDCG@10 of zero-shot and unsupervised adaptation on four datasets. Format is similar to Tab. 4

MARCO/CQA./TRECC./SCIDOCS than the best which is usually BM25), demonstrating that our end-to-end training with QA loss on NQ produces a robust retriever. We conjecture that the superior performance on QA datasets can be attributed to our end-to-end training using QA loss which learns retrieval that better aligns with the end task than training with retrieval annotations.

Retrieval Adaptation with QA Supervision As shown in the left-hand side of Tab. 4, end-to-end adaptation with QA supervision significantly improves ReAtt's retrieval performance by 5.8/8.5 on BioASQ/FiQA, achieving similar performance as Contriever fine-tuned on FiQA, and better performance than other unsupervised methods, confirming the end-to-end adaptability of our methods.

End-to-end QA Adaptation We perform end-to-end adaptation on BioASQ and compare with RAG as a baseline, which combines DPR as retriever and BART as reader, and DPR has a query and document encoder. Since updating document encoder requires corpus re-indexing, it is fixed during fine-tuning. We found end-to-end fine-tuning fails on RAG. To understand why, we conduct a rigorous experiment that breaks down each component of RAG to find the failure point in Tab. 5.

Starting from the initial model trained on NQ (#1), we first fine-tune the reader while fixing the query encoder (#2), and as expected QA performance improves. However fine-tuning both query encoder and reader (end-to-end #3) makes the retriever collapse with zero relevant documents returned, indicating end-to-end fine-tuning does not work for RAG on new domains. In order to improve both retrieval and QA, we need to fine-tune RAG in a pipeline manner: first fine-tune the re-

triever (both query and doc encoder) similarly to DPR using retrieval annotations (#4), then fine-tune the reader (#5). With the DPR-like fine-tuned retriever, end-to-end fine-tuning of query encoder and reader still fails (#6), although the retriever does not completely collapse.

End-to-end fine-tuning of ReAtt improves retrieval and QA simultaneously. Fine-tuning starting from ReAtt trained on NQ is better than starting from T5, indicating the capability learned in NQ could be transferred to BioASQ. Comparing RAG and ReAtt, we identify several keys that enable end-to-end adaptation. (1) ReAtt relying on token-level attention has a strong initial performance, (2) cross-document adjustment over both close and random documents in ReAtt provides a better gradient estimation than only using retrieved documents in RAG, (3) distillation-based loss in ReAtt might be more effective than multiplying the retrieval probability into the final generation probability.

Leveraging Retrieval Annotations As shown on the right-hand side of Tab. 4, ReAtt is able to consume retrieval supervision in a generative format and achieve competitive performance as other supervised dense retrievers.

Unsupervised Adaptation with SSM As shown in Tab. 6, adaptation by simply masking salient entities from sentences as input and generating masked entities using ReAtt improves the retrieval performance on 4 datasets, some by a large margin, achieving comparable or superior performance than strong retrieval adaptation methods such as TSDAE+GPL that relies on query generation. This indicates that our end-to-end trainable model also works well in unsupervised settings without involving too many engineering heuristics.

7 Related Work

Retrieval-augmented question answering utilizes evidence retrieved from an external knowledge source to facilitate question answering. There have been several attempts to learn retrievers and readers jointly. ORQA, REALM, RAG, EMDR^2, YONO, and Atlas (Lee et al., 2019; Guu et al., 2020; Sachan et al., 2021; Lee et al., 2021a; Izacard et al., 2022) first warm-up retrievers using unsupervised pre-training methods such as inverse cloze task (ICT), salient span masking (SSM), and large-scale contrastive learning, or initialize from supervised retrievers, then fine-tune both retriever and reader on downstream tasks. They either use fixed index (Lee et al., 2019; Lewis et al., 2020) or asynchronously update the index during training (Guu et al., 2020; Sachan et al., 2021; Lee et al., 2021a; Izacard et al., 2022). Recently, retrieval-augmented models are scaled up to very large corpora such as the web (Piktus et al., 2021; Borgeaud et al., 2021), making them capable of handling information out of the scope of Wikipedia. Atlas (Izacard et al., 2022) scales up retrieval-augmented models with T5-11B as the reader and Contriever (Izacard et al., 2021) as the retriever and achieves strong few-shot performance on multiple benchmarks. Detailed comparisons of these models can be found in Tab. 7. More related works on dense retrieval, unsupervised retrieval learning, and retrieval augmentation for language modeling can be found in Appendix C.

8 Conclusion

We propose retrieval as attention (ReAtt), a single Transformer model that can be learned in an end-to-end fashion only using end task loss. We demonstrated on NQ dataset that ReAtt can achieve both competitive retrieval and QA performance. We further show that ReAtt is easy to adapt to other domains in both supervised and unsupervised settings, achieving both boosted retrieval and end task performance. Future directions include better end-to-end training objectives, efficient training and inference, and transferring our solution to large-scale pretraining.

Limitations

ReAtt is based on token-level representations, and belongs to the same category as token-level dense retrievers such as ColBERT (Khattab and Zaharia, 2020). Comparing to passage-level dense retrievers such as DPR (Karpukhin et al., 2020), token-level retrievers usually offer better performance (shown in Tab. 1, Tab. 4, and Tab. 6) but require more space to store the index and longer query time. Our methods have the same limitation. We found ColBERT's practice in index compression and approximate search (Khattab and Zaharia, 2020; Santhanam et al., 2021, 2022) also works for our model, making this issue less of a concern.

Acknowledgments

This work was supported by a gift from Bosch Research. We would like to thank Chunting Zhou, Uri Alon, Omar Khattab, and Patrick Lewis for their insightful feedback and help with experiments.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2021. Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3080–3090. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2270–2282. Association for Computational Linguistics.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-D2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 854–870. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 981–993. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2843–2853. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3030–3042. Association for Computational Linguistics.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021b. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 316–321, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021c. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Doris Hoogeveen, Karin M. Verspoor, and Timothy Baldwin. 2015. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium, ADCS 2015, Parramatta, NSW, Australia, December 8-9, 2015*, pages 3:1–3:8. ACM.

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. 2021a. 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 and Edouard Grave. 2021b. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

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. *CoRR*, abs/2208.03299.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *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.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2020. Relevance-guided supervision for openqa with colbert. *CoRR*, abs/2007.00814.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher D. Manning, and Kyoung-Gu Woo. 2021a. You only need one model for open-domain question answering. *CoRR*, abs/2112.07381.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021b. Phrase retrieval learns passage retrieval, too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3661–3672. Association for Computational Linguistics.

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.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith B. Hall, and Ryan T. McDonald. 2021. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1075–1088. Association for Computational Linguistics.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1941–1942. ACM.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to doctttttquery.

Barlas Oguz, Kushal Lakhotia, Anchit Gupta, Patrick S. H. Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched pre-training tasks for dense retrieval. *CoRR*, abs/2107.13602.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oguz, Edouard Grave, Wen-tau Yih, and Sebastian Riedel. 2021. The web is your oyster - knowledge-intensive NLP against a very large web corpus. *CoRR*, abs/2112.09924.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. 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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Devendra Singh Sachan, Siva Reddy, William L. 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 34: Annual Conference on Neural Information Processing*

Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 25968–25981.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. PLAID: an efficient engine for late interaction retrieval. *CoRR*, abs/2205.09707.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *CoRR*, abs/2112.01488.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artières, Axel-Cyrille Ngonga Ngomo, Norman Heino, Éric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinform.*, 16:138:1–138:28.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ellen M. Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2020. TREC-COVID: constructing a pandemic information retrieval test collection. *SIGIR Forum*, 54(1):1:1–1:12.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7534–7550. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021a. TSDAE: using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 671–688. Association for Computational Linguistics.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021b. GPL: generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *CoRR*, abs/2112.07577.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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 *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Sohee Yang and Minjoon Seo. 2020. Is retriever merely an approximator of reader? *CoRR*, abs/2010.10999.

Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training language models with memory augmentation. *CoRR*, abs/2205.12674.

Model	Architecture			Retriever training			Granu.
	Retriever	Reader	Single	Init.	Warm-up	End-to-end loss	
ORQA (Lee et al., 2019)	BERT	BERT	✗	BERT	ICT	Prob. marginalization	Passage
REALM (Guu et al., 2020)	BERT	BERT	✗	BERT	ICT, SSM	Prob. marginalization	Passage
RAG (Lewis et al., 2020)	BERT	BART	✗	DPR	-	Prob. marginalization	Passage
EMDR^2 (Sachan et al., 2021)	BERT	T5	✗	BERT	ICT, SSM	Expectation maximization	Passage
YONO (Lee et al., 2021a)	T5	T5	✓	T5	SSM	Attention distillation	Passage
Atlas (Izacard et al., 2022)	BERT	T5	✗	Contriever	MLM	Perplexity distillation	Passage
ReAtt	T5	T5	✓	T5	-	Attention distillation	Token

Table 7: Detailed comparison between end-to-end retriever-reader models. ICT is inverse cloze task, SSM is salient span masking, and MLM is masked language modeling. Granu. is retrieval granularity.

Dataset	Domain	Task	Train		Test	
			#Queries	#Annotations	#Queries	|Corpus|
In-domain						
NQ	Wiki	Question answering	79K	133K	3,610	21.015M
Out-of-domain supervised adaptation						
BioASQ	Biomed	Question answering	3K	32K	500	1.000M
FiQA	Finance	Question answering	6K	14K	648	58K
MS MARCO	Misc	Information retrieval	503K	533K	6,980	8.842M
Out-of-domain unsupervised adaptation						
CQADupStack	StackExchange	Duplicate question retrieval	-	-	13,145	457K
TREC-COVID	Biomed	Information retrieval	-	-	50	171K
SCIDOCS	Science	Citation prediction	-	-	1,000	26K
SciFact	Science	Fact checking	1K	1K	300	5K

Table 8: Statistics of 8 datasets categorized by experimental settings, including the number of training/test queries, retrieval annotations (query-document pairs), and documents in the corpus.

A Efficient Implementation

Under typical optimization setups where the loss is point-wise with respect to each training data, like training classifiers or readers, scaling batch size can be easily achieved with gradient accumulation. However, due to the use of in-batch negatives, our systems, like others (Karpukhin et al., 2020; Qu et al., 2021), require having all examples in a batch to reside in GPUs simultaneously when trained directly. Larger batches therefore need proportionally more GPU memory.

In order to accommodate large batches with our limited memory hardware, we adopt the gradient cache approach (Gao et al., 2021b) decouple instances in the same batch. In particular, we run an extra forward pass over the large batch in inference mode and record (1) representations for all query and document tokens ($Q_{\boldsymbol{q}}^{B+1,h}$ and $K_{\boldsymbol{d}}^{B+1,h}$) and (2) decoder-encoder target attention values ($C_{\boldsymbol{a},\boldsymbol{q},\mathcal{D}_{\boldsymbol{q}}}$). Note that we *do not* store model internal activation nor perform gradient computation with respect to model parameters in this step. With (1) we can compute the retrieval attention, and with (2) we can compute cross-document adjustment loss (Eq. 2). We then compute and cache gradient vectors of all query and document vectors with respect to Eq. 2. We finally optimize the model with a sufficiently small batch size to fit in GPU memory and use cached gradient in the backward pass of the Eq. 2.

B Details of Adaptation Experiments

For supervised adaptation, we train on BioASQ, FiQA, and MS MARCO separately using all training queries. For CQADupStack, we merge the document corpora of 12 sub-domains into a single corpus to sample masked sentences for salient span masking training. For each of the 4 unsupervised domain adaptation datasets (CQADupStack, TREC-COVID, SCIDOCS, SciFact), we sample 20∼100K sentences and mask one entity, which is approximately proportional to the size of the corpus with a larger sampling rate for small corpora. We reuse the same hyperparameters as NQ (§ 5), except that we train each model for a single iteration using close documents from BM25 with 4K update steps and a batch size of 16. Since MS MARCO has a large number of annotations, we train for 12K update steps.

C Related Work

Dense Retrieval Models Dense retrieval models can be categorized into two groups, passage-level retrievers (Karpukhin et al., 2020; Oguz et al., 2021; Xiong et al., 2021; Gao and Callan, 2022) and token/phrase-level retrievers (Khattab and Zaharia, 2020; Khattab et al., 2020; Gao et al., 2021a; Lee et al., 2021b). Passage-level retrievers encode queries and documents into a single vector, while token/phrase-level retrievers directly use token/phrase representations, resulting in multi-vector representations. Passage-level retrievers are usually more efficient but less expressive than token-level retrievers.

Unsupervised Retrieval Learning Unsupervised retrieval learning methods can be categorized into two types: pretraining-based (Lee et al., 2019; Gao et al., 2021c; Wang et al., 2021a; Gao and Callan, 2021; Izacard et al., 2021) and question generation-based (Ma et al., 2021; Wang et al., 2021b). SimCSE (Gao et al., 2021c) obtains representations of the same input by passing through the model twice with different dropout masks and minimizes their distance. Contriever (Izacard et al., 2021) is trained by large-scale contrastive learning with random cropping of text spans sampled from Wikipedia and CCNet. GPL (Wang et al., 2021b) leverages query generators to obtain pseudo queries, and collect positive and negative documents by pseudo labeling using a cross-encoder.

Retrieval Augmentation for Language Modeling Retrieval from external datastore to improve language modeling perplexity has been explored by many works, where additional tokens are retrieved during generation based on contextual representations (Khandelwal et al., 2020; Borgeaud et al., 2021; Wu et al., 2022; Zhong et al., 2022). They differ in whether retrieval is fixed or learnable, retrieval frequency, and contextual representations used to perform nearest neighbors search.