

# End-to-End Training of Neural Retrievers for Open-Domain Question Answering

Devendra Singh Sachan<sup>1,2,3\*</sup>, Mostofa Patwary<sup>3</sup>, Mohammad Shoeybi<sup>3</sup>, Neel Kant<sup>3</sup>, Wei Ping<sup>3</sup>, William L Hamilton<sup>1,2,4</sup>, Bryan Catanzaro<sup>3</sup>

<sup>1</sup>Mila - Quebec AI Institute; <sup>2</sup>McGill University; <sup>3</sup>NVIDIA; <sup>4</sup>Canada CIFAR AI Chair  
sachande@mila.quebec, mpatwary@nvidia.com

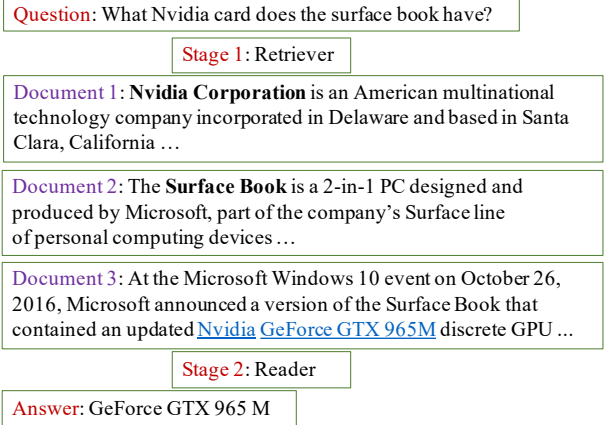
## Abstract

Recent work on training neural retrievers for open-domain question answering (OpenQA) has employed both supervised and unsupervised approaches. However, it remains unclear how unsupervised and supervised methods can be used most effectively for neural retrievers. In this work, we systematically study retriever pre-training. We first propose an approach of unsupervised pre-training with the Inverse Cloze Task and masked salient spans, followed by supervised finetuning using question-context pairs. This approach leads to absolute gains of 2+ points over the previous best result in the top-20 retrieval accuracy on Natural Questions and TriviaQA datasets.

We also explore two approaches for end-to-end supervised training of the reader and retriever components in OpenQA models. In the first approach, the reader considers each retrieved document separately while in the second approach, the reader considers all the retrieved documents together. Our experiments demonstrate the effectiveness of these approaches as we obtain new state-of-the-art results. On the Natural Questions dataset, we obtain a top-20 retrieval accuracy of 84, an improvement of 5 points over the recent DPR model. In addition, we achieve good results on answer extraction, outperforming recent models like REALM and RAG by 3+ points. We further scale up end-to-end training to large models and show consistent gains in performance over smaller models.

## 1 Introduction

The task of open-domain question answering (OpenQA) consists of finding *answers* to information-seeking *questions* using a large knowledge source such as Wikipedia. This knowledge source is also referred to as *evidence* and it typically



**Figure 1:** Open-domain question answering pipeline.

contains millions of documents. OpenQA is often regarded as a central task in Natural Language Processing (NLP). Designing algorithms for OpenQA leverages progress from a number of tasks such as text retrieval, entity recognition, co-reference resolution, and reasoning (Ferrucci et al., 2010). Most approaches for OpenQA consist of a two-stage pipeline (Chen et al., 2017; Chen, 2018). In the first stage, given a question, a *retriever* module identifies the most relevant documents, which is often a very small subset of the evidence known as *context documents*. Traditionally, approaches based on document ranking such as BM25 (Robertson and Zaragoza, 2009) have been used for the retriever. In the second stage, these relevant documents are given as input to the *reader* module, which understands them and extracts the answer for the question (Figure 1).

The main drawback of using the BM25 approach for the retriever is that it is not trainable and hence can't be adapted to different open-retrieval tasks or datasets. Recent work building on top of the advances in learning self-supervised representations in NLP such as BERT (Devlin et al., 2019)

\*This work was done during an internship at NVIDIA. Corresponding authors: Devendra Sachan, Mostofa Patwary.

have attempted to address this limitation by modeling both the retriever and reader components using neural networks, allowing the retriever to be trained using task-specific datasets (Lee et al., 2019; Karpukhin et al., 2020; Guu et al., 2020). These studies model the retriever using a *dual-encoder* architecture (Bromley et al., 1994) where one encoder processes the question and the other processes the context document. Prior work has investigated both unsupervised and supervised approaches to train the retriever. Unsupervised approaches include separately training the retriever with Inverse Cloze Task (ICT) (Lee et al., 2019) or training the retriever and reader jointly by predicting masked salient spans (Guu et al., 2020) while supervised approaches such as Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) train the retriever using human-annotated sets of question and context pairs.

However, there is no study that investigates the comparative advantages of using these two styles of training when the retrieval task is challenging *i.e.* the evidence contains millions of documents. It is unclear if the unsupervised approaches can further help to improve the performance of *strong* supervised approaches, and, if so, then under what conditions? Systematically studying these aspects of retriever training is a central focus of this work. Specifically, we propose a unified approach to train the retriever: unsupervised pre-training followed by supervised finetuning. We also introduce key design decisions—such as relevance score scaling and longer training—and showcase their effectiveness. Our results demonstrate that the proposed approach obtains substantial accuracy gains when evaluated on benchmark OpenQA datasets. Extensive experiments also highlight the relative importance of different pre-training strategies conditioned on the amount of supervised data available to train the retriever.

Furthermore, motivated by recent work (Guu et al., 2020; Lewis et al., 2020a), we also explore two approaches for end-to-end supervised training of the reader and retriever components. In the first approach, the reader considers each retrieved document separately while in the second approach, the reader takes as input all the retrieved documents together. We compare the effectiveness of these approaches on retrieval accuracy and answer extraction. We show that the first approach leads to an improved retrieval performance, while the

second approach results in an improved answer extraction. With end-to-end training, we outperform previous best models to obtain new state-of-the-art results on retrieval accuracy and answer extraction. We also perform experiments by scaling the model size to large configuration for both retriever and reader and observe that larger models consistently improve the performance compared with smaller models.

In summary, the contributions of this work are as follows:

- We demonstrate that our proposed method of *unsupervised pre-training* of the retriever with ICT followed by *supervised finetuning* leads to absolute gains of more than 2 points in the top-20 retrieval accuracy over the previous best result on Natural Questions and TriviaQA datasets.
- We show that *masked salient spans*-based pre-training of the retriever is more effective than ICT pre-training when the supervised dataset sizes are small.
- Our *end-to-end* training approach obtains new state-of-the-art performance on retrieval accuracy. On Natural Questions, our top-20 accuracy is 84, which is a 5 points gain over DPR results. Similarly, on TriviaQA, we obtain a top-20 accuracy score of 83 which is close to 4 points gain over DPR results.
- We achieve competitive results on *answer extraction* and also outperform recent models like REALM (Guu et al., 2020), and RAG (Lewis et al., 2020c) by more than 3 points.
- We *scale up* end-to-end training to *large models* and show *consistent gains* in performance.

The remainder of this paper is organized as follows. Sec. 2 and Sec. 3 explain the retriever model and end-to-end training of the retriever and reader, respectively. Sec. 4 describes the experimental setup such as datasets, model, and training details. Sec. 5 and Sec. 6 present the results on retriever accuracy and answer extraction, respectively. Sec. 7 reviews the related work. The conclusion, in Sec. 8, summarizes our methodology and results.

## 2 Retriever

In this section, we first describe the retriever architecture and then explain different approaches to train it followed by our proposed approach.

## 2.1 Background

Given a collection of documents in the evidence ( $\mathcal{Z} = \{z_1, \dots, z_m\}$ ) and a question ( $q$ ), the task of the retriever is to select a relevant subset of documents for a question. To do this, the retriever performs a ranking of the evidence documents conditioned on the question and outputs the top-ranked documents.

The retriever model consists of two modules: a question encoder ( $f_Q$ ) and a context encoder ( $f_Z$ ). Such a model is often referred to as a *dual-encoder model* (Bromley et al., 1994). Here, we explain the training methodology of the dual-encoder model given a set of questions and context documents ( $z_i$ ) from  $\mathcal{Z}$ . First, we compute the *relevance score* between the question and context. We define the relevance score to be the dot-product between the question and context representations

$$s(q, z_i; \phi) = f_Q(q)^\top f_Z(z_i) \quad (1)$$

where  $f_Q(q) \in \mathbb{R}^d$  and  $f_Z(z) \in \mathbb{R}^d$  denote the outputs of the question and context encoders, respectively, which are parameterized by  $\phi = [\phi_Q, \phi_Z]$ . We model the  $f_Q$  and  $f_Z$  using BERT-style transformer networks (Devlin et al., 2019; Vaswani et al., 2017). We consider the hidden states of the first token of the sequence ([CLS] token) as the encoder’s output. The probability of a context document  $z_i$  being relevant to the question  $q$  is calculated as

$$p(z_i | q, \mathcal{Z}; \phi) = \frac{\exp(s(q, z_i; \phi)/\tau)}{\sum_{j=1}^{|\mathcal{Z}|} \exp(s(q, z_j; \phi)/\tau)} \quad (2)$$

where  $\tau$  is a scaling factor. The scaling factor helps in better optimization when the model hidden size ( $d$ ) is large. While previous work had used the setting of  $\tau = 1$ , in this work, we set  $\tau = \sqrt{d}$ . We refer to this as *relevance score scaling*.

In practice, as the evidence set consists of millions of documents, the normalization term would be expensive to compute. We approximate the denominator of the above equation by using other context documents in the batch as negative examples, a technique which has shown to perform well in practice (Karpukhin et al., 2020).

## 2.2 Training

In this section, we discuss different approaches to train the retriever model. In all the approaches, we initialize the parameters of both the question and context encoders using BERT weights as implemented in Megatron-LM (Shoeybi et al., 2019).

We also experimented with random initialization but it vastly underperformed BERT initialization.

### 2.2.1 Supervised Training

In this setting, *human-annotated* set of questions, answers, and sometimes context is provided. If the context is not included, then a common approach is to use distant supervision (Mintz et al., 2009) to obtain the context document. Specifically, we select the top-ranked document using BM25 (Robertson and Zaragoza, 2009) from the evidence that contains the answer as the context. We also select other top-ranked documents that do not contain the answer as additional *hard* negative examples. This approach to train neural retriever was popularized by (Karpukhin et al., 2020).

### 2.2.2 Unsupervised Training

**Inverse Cloze Task (ICT):** In this setup, we do not consider the human-annotated question-answer pairs. Instead, we simulate the supervised setup in an unsupervised manner. We sample a sentence from the context which is considered as the *pseudo*-query and other sentences as the *pseudo*-context. This approach was first proposed by (Lee et al., 2019).

**Masked salient spans generative training:** Recently, it is shown by (Guu et al., 2020) that the performance of the ICT initialized retriever can be further improved by training it using an objective to predict the masked salient spans such as named entities. In this work, we use a similar approach. However, unlike (Guu et al., 2020) we use a generative language model based on T5 (Raffel et al., 2020) for the reader component instead of using BERT.

## 2.3 Unsupervised Pre-training and Supervised Finetuning

To improve the retriever training, we propose the approach of unsupervised pre-training of retriever followed by supervised finetuning. In this approach, we use the ICT and masked salient spans generative training (Sec. 2.2.2) for the unsupervised pre-training part. We also compare their trade-offs. After warm-starting the retriever weights with unsupervised pre-training, we further finetune it with supervised training as explained above in Sec. 2.2.1.

### 3 End-to-End Retriever and Reader Training

In this section, we describe two *supervised training* approaches to end-to-end train the reader and retriever networks from the task-specific data. In the first approach, the reader considers each retrieved document separately (Sec. 3.1) while in the second approach, the reader takes as input all retrieved documents together (Sec. 3.2). These approaches are designed such that when predicting the answer conditioned on the question, the learning process improves both the reader and retriever. It is worthwhile to mention that (Karpukhin et al., 2020) also performed end-to-end training but didn’t observe performance gains over separately training the retriever and the reader.

**Background and notation:** In end-to-end training, the trainable components consists of the retriever ( $\phi$ ) and reader ( $\theta$ ) parameters. For retriever, we use the dual-encoder architecture and train it as discussed previously in Sec. 2.3. Our reader is a *generative model* designed according to sequence-to-sequence modeling paradigm (Sutskever et al., 2014). Specifically, we use pre-trained T5 as the reader. The inputs to the training process are *questions* ( $q$ ) and its *answers* ( $a$ ), both in string form. Given a question, first the retriever obtains the  $k$  *relevant* context documents ( $\mathcal{K}$ ) from the evidence ( $\mathcal{Z}$ ) as

$$\mathcal{K} = \arg \operatorname{sort} s(q, z_i; \phi)[:, k] \quad (3)$$

The reader then takes as input the question and one or more context documents ( $z_i$ ) to predict the answer, the likelihood of which is defined as

$$p(a | q, z_i; \theta) = \prod_{j=1}^N p(a_j | q, z_i; \theta), \quad (4)$$

where  $N$  is the number of answer tokens. Next, we describe both the proposed approaches. A block diagram illustrating the end-to-end training process is shown in Figure 2.

#### 3.1 Approach 1: Individual Top-k

In this approach, similar to (Guu et al., 2020), the reader’s likelihood is first computed conditioned on the question and each retrieved document. The marginal likelihood is defined as the weighted av-

erage of the *individual* likelihoods as

$$p(a | q; \theta, \phi) = \sum_{z_i \in \mathcal{K}} p(a | q, z_i; \theta) p(z_i | q, \mathcal{Z}; \phi), \quad (5)$$

where  $p(z_i | q, \mathcal{Z}; \phi)$  is computed using Eq. 2. However, the normalization is done over  $\mathcal{K}$  instead of  $\mathcal{Z}$ . The final loss is defined as the negative marginal log-likelihood

$$\mathcal{L}(q, a) = -\log p(a | q; \theta, \phi). \quad (6)$$

We note that the RAG model (Lewis et al., 2020c) also proposes a similar approach, but there are two main differences. The first is that while we update all the parameters of the retriever (both the query and context encoders), RAG just updates the query encoder. The second is that we use T5 model as the reader while RAG uses BART model (Lewis et al., 2020b). These enhancements help us obtain substantial gains over the RAG model, which we will discuss in Sec. 6.

#### 3.2 Approach 2: Joint Top-k

In this approach, similar to (Lewis et al., 2020a), the likelihood is defined as the reader’s likelihood conditioned on the question, *all* the retrieved documents, and the retrieval score

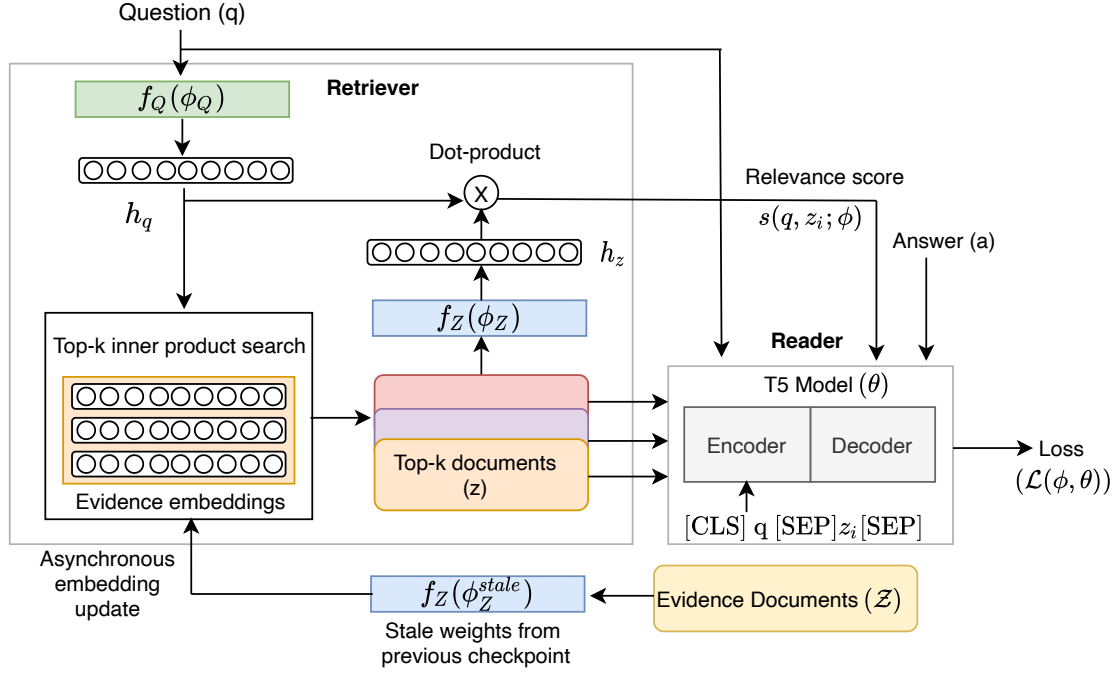
$$p(a | q; \theta, \phi) = p(a | q, z_{1:k}, p(z | q, \mathcal{Z}; \phi); \theta). \quad (7)$$

As the T5 reader consists of separate encoder and decoder modules, it provides the flexibility to customize the input or output of the encoder. We concatenate each retrieved document with the question and feed them as input to the encoder, which computes their hidden representations. Next, we stack the hidden representations of all the retrieved documents, which the decoder *jointly* attends to during the encoder-decoder attention, thus allowing a more powerful form of information aggregation from multiple retrieved documents. We also add retriever similarity score bias to the encoder-decoder attention to help facilitate end-to-end training. The interaction score during the encoder-decoder attention is computed as

$$\operatorname{attn}(q, a, z_{1:k}) \propto Q(a)^\top K(z_{1:k}, q) + \lambda p(z | q; \phi), \quad (8)$$

where  $Q$  denotes the query vector computed from decoder’s input,  $K$  denotes the key vector computed from encoder’s output, and  $\lambda$  is a trainable parameter.





**Figure 2:** A schematic diagram illustrating end-to-end supervised training of the retriever and reader components.

Final loss is defined according to Eq. 6. We further note that a similar approach for OpenQA was proposed in (Izcard and Grave, 2020) but it just optimizes the reader model and didn’t perform end-to-end training of the retriever.

## 4 Experimental Setup

In this section, we describe the datasets, model settings, and training details.

### 4.1 Evidence Dataset

Following (Karpukhin et al., 2020), we make use of their preprocessed version of the English Wikipedia dump from December 2018 as the source of evidence documents. Overall, there are around 21M documents, each 100 words long.

### 4.2 OpenQA Datasets

We perform experiments using two commonly used QA datasets whose details are provided below and their statistics are shown in Table 1.

**Natural Questions (NQ):** This corpus consists of real questions asked from the Google search engine along with their long and short answer annotations from the top-ranked Wikipedia pages (Kwiatkowski et al., 2019). Following prior work (Lee et al., 2019; Karpukhin et al., 2020), we use the same subset of the short answer questions in our experiments, as it is more suited for

Dataset	Train	Filtered Train	Dev	Test
NQ	79,168	58,880	8,757	3,610
TriviaQA	78,785	60,413	8,837	11,313

**Table 1:** OpenQA dataset statistics. The full training set is used for end-to-end training, while the filtered training set is used to train the retriever. Essentially, the filtered set ignores those examples where the context document retrieved from Wikipedia does not align with the provided ground-truth context.

open-domain question answering.

**TriviaQA:** This corpus consists of a collection of trivia questions and their answers scraped from multiple sources in web (Joshi et al., 2017). The training data is constructed using distant supervision, i.e. selecting those paragraphs in Wikipedia that are ranked high in BM25 scores and contain the answer string.

### 4.3 Model Details

We use two models of different sizes, *base* and *large*, for the experiments.

**Base configuration:** It consists of 12 layers, 768-d hidden size, and 12 attention heads. The BERT-base contains 110M parameters while the T5-base contains 220M parameters.

**Large configuration:** It consists of 24 layers, 1024-d hidden size, and 16 attention heads. The

BERT-large contains 330M parameters while the T5-large contains 770M parameters.

#### 4.4 Training Details

We provide the training details of all the experiments below. We use the same setup for both the base and large configurations and use open-source Megatron-LM toolkit (Shoeybi et al., 2019) to implement the models.<sup>1</sup>

##### 4.4.1 Retriever Training

**Supervised:** We use a batch size of 128, learning rate of  $2e-5$ , and set number of epochs to  $\{40, 80\}$ .

**ICT training:** We initialize the parameters of both the question and context encoders using BERT weights trained with Megatron-LM. We use a batch size of 4096, learning rate of  $1e-4$ , and train the model for 100,000 steps using Adam optimizer (Kingma and Ba, 2015). We set the weight decay to 0.01 and the warmup ratio of the optimizer to 0.01. We use Wikipedia paragraphs of maximum tokens 256 to train the model. With a probability of 0.1, we also keep the pseudo-query sentence in the context.

**Masked salient spans generative training:** We initialize the retriever with ICT training and pre-train the T5 reader on an aggregated dataset from (Shoeybi et al., 2019). We use pre-trained models provided by the Stanza toolkit (Qi et al., 2020) to segment Wikipedia paragraphs into sentences and extract named entities.<sup>2</sup> We train the model for 100,000 steps with Adam optimizer using a learning rate of  $2e-5$  and a warmup ratio of 0.05. We compute the evidence embeddings asynchronously and update the evidence index every 500 steps.

##### 4.4.2 End-to-End Supervised Training

As the performance of the ICT pre-trained retriever and masked salient spans pre-trained retriever is similar when all the training data is used (Sec. 5.2), we select the retriever finetuned with ICT initialization. As mentioned above, we use a pre-trained T5 reader. We train for 10 epochs using a batch size of 64, learning rate of  $2e-5$ , and weight decay 0.1. During training, we update the evidence embeddings index every 500 steps.

<sup>1</sup><https://github.com/NVIDIA/Megatron-LM>

<sup>2</sup>We use the model trained on OntoNotes (Pradhan et al., 2012) to extract named entities for 10 selected categories.

Model	top-1	top-5	top-20	top-100
<i>Base Configuration</i>				
CLS-pool, 40 epochs	32.6	60.1	76.4	85.9
+ score scaling	34.1	60.9	77.6	85.9
+ 80 epochs	36.7	62.2	77.4	<b>86.0</b>
+ 1 hard negative	<b>48.6</b>	<b>74.5</b>	<b>79.0</b>	85.8

**Table 2:** Effect of different factors on the supervised training of retriever when evaluated on NQ test set.

## 5 Results on Retriever Training

In this section, we discuss retriever accuracy results using different training methods.

### 5.1 Effect of Relevance Score Scaling, Longer Training, and Hard Negatives

In this section, we explore the best training settings for supervised training of the retriever. To do so, we perform a series of experiments on the NQ dataset starting with the training settings from the DPR model (Karpukhin et al., 2020) and then progressively improve it. As the DPR approach was trained for 40 epochs, scaling factor of 1, with [CLS] token pooling performed on the outputs of the dual-encoder, we first report the results of this setting in Table 2. Next, we observe that incorporating relevance score scaling during model training helps to improve the top-1, top-5, and top-20 accuracy scores by 1-1.5 points. We further perform longer training till 80 epochs and it also helps improve top-1 and top-5 accuracy by an additional 1.5-2 points. We next train the retriever with 1 additional hard-negative example for each question-context pair in the batch. Our results, in line with the results of (Karpukhin et al., 2020) show gains of more than 10 points in the top-1 and top-5 accuracy.

From these results, we note that relevance score scaling, longer training, and including 1 hard negative example are important components in improving the retrieval accuracy. Therefore, we use these settings in our next experiments for the supervised training of the retriever.

### 5.2 Effect of Retriever Initialization

We now analyze how retriever weights initialization with ICT and masked salient spans pre-training affects retrieval accuracy. We first discuss the results of unsupervised training, followed by supervised training, and finally our proposed approach. We present the results for NQ and TriviaQA in Table 3.

Model	NQ				TriviaQA			
	<i>top-1</i>	<i>top-5</i>	<i>top-20</i>	<i>top-100</i>	<i>top-1</i>	<i>top-5</i>	<i>top-20</i>	<i>top-100</i>
<i>Base Configuration</i>								
ICT	12.6	32.3	50.6	66.8	19.2	40.2	57.5	73.6
Masked salient spans	20.0	41.7	59.8	74.9	31.7	53.3	68.2	79.4
Supervised	48.6	68.8	79.0	85.8	57.5	72.2	80.0	85.1
ICT + Supervised	48.4	<b>72.1</b>	<b>81.8</b>	<b>88.0</b>	58.4	73.9	81.7	86.3
Masked salient spans + Supervised	<b>50.3</b>	71.9	82.1	87.8	<b>60.6</b>	<b>74.8</b>	<b>81.8</b>	<b>86.6</b>
<i>Large Configuration</i>								
ICT	13.0	31.8	49.3	66.1	20.1	41.6	58.5	74.1
Supervised	51.4	71.0	81.0	87.2	60.4	74.5	81.4	86.0
ICT + Supervised	<b>52.4</b>	<b>72.7</b>	<b>82.6</b>	<b>88.3</b>	<b>61.9</b>	<b>76.2</b>	<b>82.9</b>	<b>87.1</b>

**Table 3:** Effect of unsupervised pre-training on retrieval accuracy when evaluated on NQ and TriviaQA test sets.

Model	NQ						TriviaQA			
	Q	C	<i>top-1</i>	<i>top-5</i>	<i>top-20</i>	<i>top-100</i>	<i>top-1</i>	<i>top-5</i>	<i>top-20</i>	<i>top-100</i>
<i>Base Configuration</i>										
DPR (Karpukhin et al., 2020)			–	67.1	78.4	85.4	–	–	79.4	85.0
ICT + Supervised			48.4	72.1	81.8	88.0	58.4	73.9	81.7	86.3
Individual Top-k	✓		54.5	73.7	83.2	88.6	61.4	75.6	82.1	86.7
Individual Top-k	✓	✓	<b>56.8</b>	<b>75.0</b>	<b>84.0</b>	<b>89.2</b>	<b>63.5</b>	<b>76.8</b>	<b>83.1</b>	<b>87.0</b>
Joint Top-k	✓		51.1	72.1	81.8	87.8	59.1	74.1	81.3	86.3
<i>Large Configuration</i>										
ICT + Supervised			52.4	72.7	82.6	88.3	61.9	76.2	82.9	87.1
Individual Top-k	✓	✓	<b>57.5</b>	<b>76.2</b>	<b>84.8</b>	<b>89.8</b>	<b>66.4</b>	<b>78.7</b>	<b>84.1</b>	<b>87.8</b>
Joint Top-k	✓		53.7	73.3	83.2	88.0	61.2	75.9	82.7	87.0

**Table 4:** Effect of end-to-end training on retrieval accuracy when evaluated on NQ and TriviaQA test sets. **Q** and **C** signify if the query encoder and the context encoder, respectively are updated during training or not.

We observe that unsupervised training with ICT is quite effective in providing a non-trivial zero-shot retrieval accuracy on both the datasets. Moreover, masked salient spans-based generative pre-training further improves zero-shot retrieval accuracy by more than 8 points over ICT initialization. Supervised training shows an impressive performance for all the top-k values and can be considered as a very strong baseline.

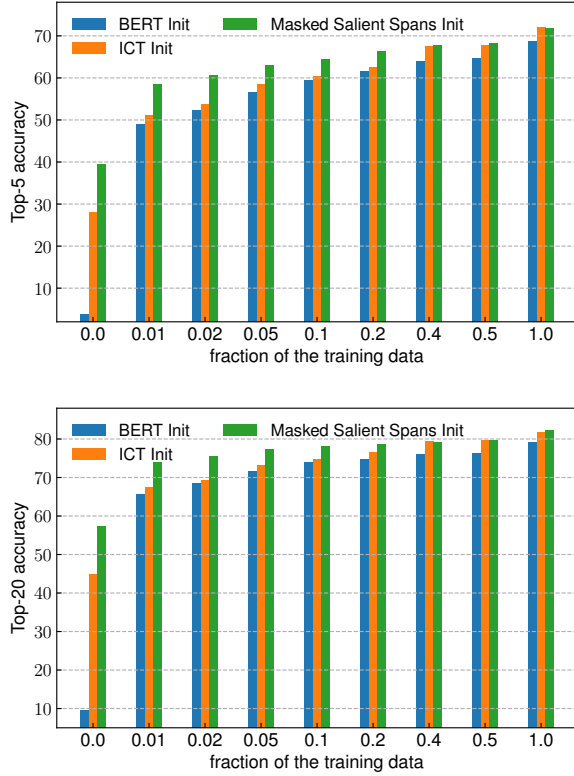
Further, we showcase that our proposed approach of unsupervised pre-training with both ICT and masked salient spans followed by supervised finetuning provides improvements of 2-3 points over the strong results that were achieved with supervised training. We observe that our gains are consistent across both NQ and TriviaQA. Comparing ICT and masked salient spans initialization, we note that their accuracy gains roughly similar. Lastly, we show more accuracy gains when we scale up the model size from the base to the large configuration, with the large configuration achieving state-of-the-art performance results.

### 5.3 Effect of Amount of Training Data

In this section, we study the effect on retrieval accuracy when the retriever is pre-trained with either BERT, ICT, or masked salient spans and the amount of training data is varied. We train the retriever with 1%, 2%, 5%, 10-50%, of NQ’s training data and plot the top-5 and top-20 accuracy in Figure 3. We see that at smaller fractions of the training data, masked salient spans pre-training is much more effective than ICT pre-training, consistently leading to large gains. However, as the fraction of training data increases to 40% or higher, ICT and salient spans pre-training perform almost similarly. We further note that compared to BERT initialization, ICT and masked salient spans pre-training leads to more optimal performance.

### 5.4 Effect of End-to-End Training

We next describe the effect of end-to-end training on retrieval accuracy. The focus here is to analyze how much retrieval accuracy improves if we update retriever weights during end-to-end training.



**Figure 3:** Effect of amount of training data on retrieval accuracy when evaluated on NQ test set.

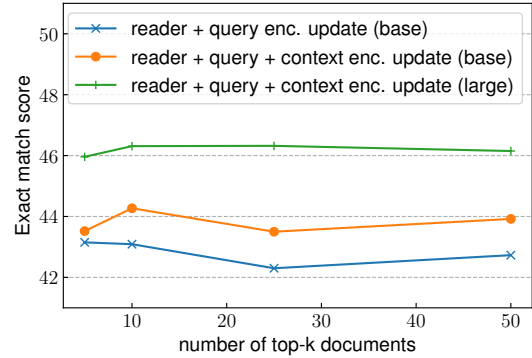
From the results in Table 4, we see that when the retriever is finetuned using *Individual Top-k* approach and when the query encoder is updated while the context encoder is kept fixed, it further improves the performance of top-1 by 6 points, top-5 and top-20 by more than 1.5 points on NQ. Furthermore, we see that when the context encoder is also updated with the evidence embeddings index being refreshed every 500 steps asynchronously, the retrieval accuracy improves to 75% at top-5 and 89.2% at top-100 leading to substantial improvements over previous state-of-the-art DPR retriever.

We also find that larger retriever models help further improve the top-5 score by 1.2% and top-100 score by 0.6%, thus obtaining new state-of-the-art results. Overall, end-to-end training of the retriever with *Individual Top-k* surpasses the previous best approach of DPR by more than 9 points in top-5, 6 points in top-20, and 4 points in top-100 on NQ. Similarly for TriviaQA, we obtain new state-of-the-art results in retrieval accuracy for both the base and large configurations.

The second approach of *Joint Top-k*, when updating the query encoder improves the top-1 by close to 2.5 points on NQ but doesn’t lead to accuracy

Model	NQ	TriviaQA
<i>Base Configuration</i>		
ORQA (Lee et al., 2019)	33.3	45.0
REALM (Guu et al., 2020)	40.4	–
DPR (Karpukhin et al., 2020)	41.5	56.8
Individual Top-k	<b>45.9</b>	56.3
<i>Large Configuration</i>		
RAG (Lewis et al., 2020c)	44.5	56.8
Individual Top-k	<b>48.1</b>	<b>59.6</b>

**Table 5:** Answer extraction results using *Individual Top-k* approach. The grouping under base and large configurations is based on the size of the reader model.



**Figure 4:** Effect of increasing top-k documents on answer generation for *Individual Top-k* approach evaluated on NQ dev set.

gains for higher top-k values. Similarly, for the large configuration, we see a gain of around 1 point in top-1 accuracy but relatively smaller gains for higher top-k values.<sup>3</sup> Almost a similar trend follows for TriviaQA. From these results, we conclude that the *Joint Top-k* approach doesn’t improve the performance of the retriever when the retriever is already well-initialized. As we will show next, that the utility of this method lies in answer extraction.

## 6 Results on Answer Extraction

We next examine the utility of both *Individual Top-k* and *Joint Top-k* training approaches for answer extraction. We report the results using the Exact Match (EM) metric.

### 6.1 Individual Top-k

We compare our results with previous approaches such as ORQA (Lee et al., 2019), REALM (Guu et al., 2020), and DPR which similarly use one doc-

<sup>3</sup>We don’t update the context encoder for *Joint Top-k* as it didn’t improve the performance during our initial experiments.



Model	NQ	TriviaQA
<i>Base Configuration</i>		
FiD (Izcard and Grave, 2020)	48.2	65.0
Joint Top-k	<b>49.2</b>	64.8
<i>Large Configuration</i>		
FiD (Izcard and Grave, 2020)	51.4	67.6
Joint Top-k	<b>51.4</b>	<b>68.3</b>

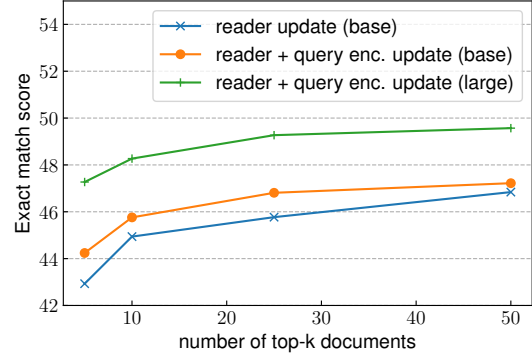
**Table 6:** Results on answer extraction using *Joint Top-k* model.

ument for answer extraction. The notable difference is that while these are span-based approaches that select the best answer span from the document, ours is a generative approach that generates the answer conditioned on a document.

During inference, the reader model first greedily generates an answer for each retrieved document. Next, we score each generated answer using Eq. 5 and finally select the answer with the highest probability score. We present the results in Table 5. We showcase that for the base configuration on NQ, our model outperforms both REALM and DPR approaches by more than 4 points.

For the large configuration, we compare with the RAG approach (Lewis et al., 2020c), which is similar to ours but uses BART (Lewis et al., 2020b) pre-trained generative model as the reader and DPR as the retriever. RAG also performs joint training but without updating the retriever’s context encoder. Our approach outperforms the RAG model by more than 3.5 points. We believe that our better results are due to a more accurate initial retriever, stronger reader model, and end-to-end training which includes updating the context encoder.

In TriviaQA, for the base configuration, our model’s performance is close to the DPR results while for the large configuration, it outperforms RAG results by 2.8 points. In Figure 4, we show the effect of context encoder updates and different number of top-k documents on EM scores. It can be seen that updating the context encoder in joint training is very helpful for both the base and large configurations as it consistently improves the results for different top-k’s. We also note that the performance of *Individual Top-k* approach is sensitive to the number of top-k documents and can also decrease with an increase in top-k documents.



**Figure 5:** Effect of increasing top-k documents on answer generation for *Joint Top-k* approach evaluated on NQ dev set.

## 6.2 Joint Top-k

We compare our results with a similar approach from (Izcard and Grave, 2020) called Fusion-in-Decoder (FiD), where the authors use the retrieved documents from the DPR model and use the T5 model as reader initialized with the open-source weights.<sup>4</sup> The difference is that unlike our approach, they just finetune the reader weights and don’t perform end-to-end training. From the results in Table 6, we see that for the base configuration, our approach outperforms the FiD model by 1 point on NQ and performs almost similarly on TriviaQA. For the large configuration, our results are identical to the FiD model on NQ and obtain a gain of 0.7 points on TriviaQA.

In our analysis presented in Figure 5, we see that for both the base and large configurations, the performance of *Joint Top-k* increases as the number of top-k documents is increased. This highlights that in contrast to the *Individual Top-k* approach, the *Joint Top-k* has the potential to better leverage the information contained in multiple context documents to generate the final answer. We also observe that with the increase of top-k documents, the utility of the query encoder update tends to get diminished, thus explaining the relatively low gains observed in retrieval performance when jointly training the model.

## 6.3 Overall Comparison

From the discussions in Sec. 5.4 and Sec. 6, we showcase that end-to-end training using the two approaches has a complementary effect on the re-

<sup>4</sup><https://github.com/google-research/text-to-text-transfer-transformer>

trieval and answer generation. While the *Individual Top-k* approach helps to improve the retrieval performance, the *Joint Top-k* approach is more useful for answer extraction. Therefore, we believe that future OpenQA systems would leverage both these approaches in sequence — first, *Individual Top-k* can be used to train an accurate retriever followed by approaches similar in spirit to *Joint Top-k* which can better aggregate the information contained in the retrieved documents to extract the answer.

## 7 Related Work

(Yih et al., 2011) proposed one of the early successful approaches for learning dense representations of query and evidence using discriminative retriever models. However, this approach was data-hungry and not trivial to scale. Recently, (Lee et al., 2019; Karpukhin et al., 2020) have attempted to address this limitation by leveraging pre-trained models like BERT (Devlin et al., 2019). This makes it possible to train dual-encoder retriever models using small amounts of question-context pairs. In particular, (Lee et al., 2019) first pre-train the retriever in an unsupervised manner with the ICT task and then jointly train the retriever and reader for OpenQA. On the other hand, (Karpukhin et al., 2020) perform supervised training of the retriever by using hard-negative examples along with in-batch negatives, yielding impressive results on several OpenQA retrieval benchmarks.

To improve the retrieval accuracy of the BERT-initialized dual-encoder retriever model, (Chang et al., 2020) explore three paragraph-level pre-training strategies: ICT, body-first selection, and Wikipedia link prediction. They demonstrated that with these pre-training tasks, the dual-encoder retriever provides improved performance over sparse-retrieval approaches such as BM-25. In their work, the evidence set consists of the training set context which was further increased to 1M documents for open-domain retrieval experiments. Our work differs from them in several ways. First, our OpenQA setup is more challenging as the evidence set consists of 21M documents. Second, we pre-train with two strategies consisting of ICT and masked salient-spans and finetune using strong supervised methods, which leads to much improved results. Third, we further update the retriever with end-to-end supervised training leveraging question-answer pairs, which substantially improves the retrieval accuracy leading to state-of-the-art results.

A new line of work investigates task-specific end-to-end unsupervised pre-training approaches. For example, (Guu et al., 2020) predicts masked salient spans consisting of named entities to pre-train the reader and retriever components for OpenQA tasks. Similarly, (Lewis et al., 2020a) perform cross-lingual pre-training where the objective is to reconstruct a sequence using its paraphrases in different languages, demonstrating improved zero-shot performance in document translation tasks.

There is also an emerging line of work on performing OpenQA using the knowledge stored in the parameters of a large pre-trained language model (Petroni et al., 2019; Jiang et al., 2020). Impressive performance has been shown for both zero-shot OpenQA (Brown et al., 2020) and when using a small amount of data to finetune the language model (Roberts et al., 2020). The advantage of these methods is that an additional retriever component and text storage is not required while a limitation is that these models would have to be re-trained to introduce new knowledge in them, which is expensive. Recently, (Verga et al., 2020) have proposed approaches to update the knowledge stored in a pre-trained language model that doesn't require re-training from scratch.

## 8 Conclusion

In this work, we propose approaches to improve the accuracy of the dual-encoder retriever model for the task of open-domain question answering (OpenQA). We demonstrate that unsupervised pre-training of the retriever with Inverse Cloze Task (ICT) followed by supervised training using question-context pairs helps to improve the retrieval accuracy over the previous best results. Our empirical analysis further reveals that pre-training with masked salient spans is a more effective approach than ICT when the training dataset sizes are relatively smaller. We also propose two approaches to perform end-to-end supervised training of the OpenQA systems using question-answer pairs. The first approach where the reader considers each retrieved document separately excels in improving the retrieval accuracy. The second approach where the reader considers all the retrieved documents together excels in answer extraction. With end-to-end training, we outperform previous best models and achieve new state-of-the-art results on both retrieval accuracy and answer extraction on benchmark datasets.

## References

- Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. [Signature verification using a "siamese" time delay neural network](#). In *Advances in Neural Information Processing Systems*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. [Pre-training tasks for embedding-based large-scale retrieval](#). In *International Conference on Learning Representations*.
- Danqi Chen. 2018. *Neural Reading Comprehension and Beyond*. Ph.D. thesis, Stanford University.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.
- 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 and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wenteau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *The 2015 International Conference for Learning Representations*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of 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 Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.
- Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. *arXiv preprint arXiv:2006.15020*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. 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*.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, V. Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020c. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.
- 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)*.

- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. [CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes](#). In *Joint Conference on EMNLP and CoNLL - Shared Task*.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A Python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- 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)*.
- S. Robertson and H. Zaragoza. 2009. *The Probabilistic Relevance Framework*.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. *arXiv preprint arXiv:1909.08053*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.
- Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *arXiv preprint arXiv:2007.00849*.
- Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*.