Domain-matched Pre-training Tasks for Dense Retrieval

Barlas Oğuz[*], Kushal Lakhotia[*], Anchit Gupta[*],
Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen,
Sebastian Riedel, Wen-tau Yih, Sonal Gupta, Yashar Mehdad

Meta AI

{barlaso,kushall,anchit,plewis,vladk,piktus,
xilun,sriedel,scottyih,sonalgupta,mehdad}@fb.com

Abstract

Pre-training on larger datasets with ever increasing model size is now a proven recipe for increased performance across almost all NLP tasks. A notable exception is information retrieval, where additional pre-training has so far failed to produce convincing results. We show that, with the right pre-training setup, this barrier can be overcome. We demonstrate this by pre-training large bi-encoder models on 1) a recently released set of 65 million synthetically generated questions, and 2) 200 million post-comment pairs from a preexisting dataset of Reddit conversations. We evaluate on a set of information retrieval and dialogue retrieval benchmarks, showing substantial improvements over supervised baselines.

1 Introduction

As a pre-training task, language modeling and its variants (causal (Radford et al., 2018), bi-directional (Peters et al., 2018; Baevski et al., 2019), masked (Devlin et al., 2018), seq2seq (Lewis et al., 2020; Raffel et al., 2019)) have proven to be extremely versatile and shown to transfer well to most, if not all NLP tasks. Nevertheless, in-domain fine tuning remains important, as there is still a gap between the pre-training task and the downstream tasks. Numerous approaches have been proposed to fill this gap, with an additional (intermediate) pre-training stage, mostly based on multi-task learning (Raffel et al., 2019; Aghajanyan et al., 2021). It's been generally accepted that the more similar the end task is to the pre-training task, the larger the gains (e.g., NLI tasks transfer better to other NLI tasks (Phang et al., 2018), QA tasks to QA tasks (Khashabi et al., 2020), *inter alia*).

From this perspective, information retrieval (IR), which is the task of identifying the most relevant document to a given query from a large corpus of candidates, has a unique position. At the surface, IR looks similar to other NLP tasks in standard benchmarks, such as NLI or paraphrase detection. However, the need to accommodate large corpora imposes computational constraints, which leads to important practical differences. Most importantly, indexing needs to happen offline, therefore the candidate representations need to be calculated independently of the query representation. As a result, neural retrieval systems typically use a *bi-encoder* model (Figure 1), trained to minimize the similarity between the document representation and the query representation. This shallow interaction between document and query encoders makes neural IR models architecturally unique, compared to the block cross-attention transformers which are the universal choice for almost every other NLP task.

Researchers have therefore recognized the need to construct intermediate pre-training tasks that are better matched to retrieval. Lee et al. (2019) proposed the *inverse cloze task (ICT)*, which treats sentences as pseudo-queries, and matches them to the paragraph they originate from. Chang et al. (2020) combined this with *body first selection (BFS)* (selecting the first paragraph given a sentence from the same document), and *wiki link prediction*. Guu et al. (2020) pre-trained a retrieval model jointly in an end-to-end system to minimize a language modelling objective.

In each of these cases, pre-training approaches were shown to improve over their respective baselines. However, subsequent work showed that a careful fine-tuning over a vanilla BERT model can outperform all of these approaches (Karpukhin et al., 2020). The findings for model scaling are also similar to those of data scaling. Published results show only modest improvements from larger models for retrieval, and retrieval models which top the most competitive document ranking leaderboards are still based on the relatively small BERT-

[*] Equal contribution

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1524 - 1534
July 10-15, 2022 ©2022 Association for Computational Linguistics

base architecture.[1] This is in sharp contrast to other NLP benchmarks, where data and model scaling has been extremely successful.

We hypothesise that previously proposed pre-training tasks might be still too distant from the target task, which limits useful transfer. We therefore investigate pre-training tasks for retrieval which are as closely matched to the the target task and domain as possible. To this end, we propose using two corpora for retrieval pre-training:

- A corpus of 65 million synthetically generated question-answer pairs from Wikipedia (PAQ, Lewis et al., 2021), which we target for open domain question answering and other passage retrieval tasks.
- A corpus of 220 million post-comment pairs from Reddit, which we use for dialogue retrieval tasks.

We conduct extensive evaluations on two popular information retrieval tasks, a benchmark composed of 8 knowledge-intensive retrieval tasks, and 3 dialogue retrieval benchmarks. We find that pre-training leads to strong improvements in all cases, and also demonstrate robust generalization. We compare different pre-training tasks, investigating the effect of domain and task similarity, and find both to be important. We also experiment with models of varying sizes, with and without pre-training, showing in some cases that retrieval can indeed benefit from larger models.

2 Dense retrieval

In this section we give an overview of dense retrieval models and how they are trained.

2.1 Bi-encoder architecture

A typical dense retrieval system consists of a query encoder E_Q and a passage encoder E_P, which output a fixed d-dimensional representation for each query and passage respectively. Passages are processed offline, and their representations are indexed using a fast vector similarity search library such as FAISS (Johnson et al., 2017). At runtime, an incoming query is encoded, and the top-k closest passages to its representation in vector distance are returned using the index. Dot-product similarity is most commonly used:

$$\mathrm{sim}(q,p) = E_Q(q)^\intercal E_P(p). \qquad (1)$$

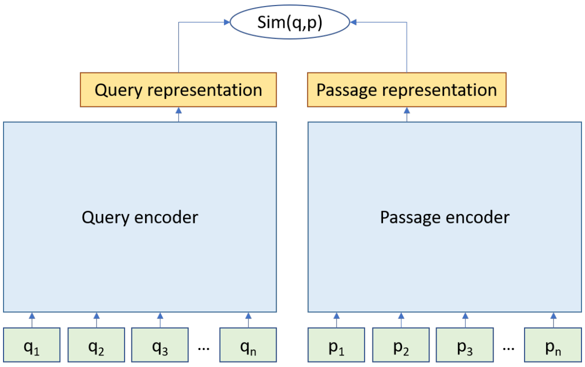

Figure 1: Bi-encoder architecture for retrieval.

The resulting *bi-encoder* architecture is pictured in Figure 1. Crucially, this formulation allows passage representations to be calculated independently from the query encoder, making efficient retrieval possible.

2.2 Training

Given a query, a relevant (positive) passage, and a list of non-relevant (negative) passages, the bi-encoder model is trained to minimize the negative log likelihood of picking the positive passage, where the probability assigned to each passage is proportional to $e^{\mathrm{sim}(q,p)}$. For efficiency reasons, positive passages are recycled as negative passages for queries they are not paired with in the batch, referred to as *in-batch* negatives. In addition, *hard* negatives have been found to be useful, which can either come from a standard retrieval system such as BM25, or an earlier iteration of the dense model (Xiong et al., 2020). We do training in two steps (which we refer to as *iterative training*). In the first step we use a single BM25 negative per query, following best practice from Karpukhin et al. (2020), and in the second step we use hard negatives obtained using the first round model. This procedure approximates the asynchronous model update, which was shown to be helpful in Xiong et al. (2020).

3 Experimental setup

3.1 Pre-training tasks

In this section, we describe the datasets we used to pre-train our retrieval models.

3.1.1 PAQ

For open-domain question answering tasks, we employ the recently-released PAQ dataset (Lewis et al., 2021). This dataset consists of 65 mil-

[1] https://microsoft.github.io/msmarco/

lion synthetic question-answer pairs, generated from Wikipedia passages. PAQ is generated by a pipeline of models trained on Natural Questions (NQ, Kwiatkowski et al., 2019) and TriviaQA (TQA, Joshi et al., 2017). PAQ's main distinguishing features relative to other QA-pair generation techniques are its large size, and the use of a novel *global* consistency filtering. This leads to higher quality, less ambiguous open-domain-style questions than can be achieved using standard consistency filtering with machine-comprehension models (Alberti et al., 2019).

PAQ has previously been employed as a semi-structured knowledge base of facts extracted from Wikipedia, and used as data-augmentation for closed-book question answering models (Roberts et al., 2020). However, since QA-pairs in PAQ are generated from Wikipedia passages, we can repurpose PAQ as a source of training data for a passage retrieval task. Here, given a PAQ question, the task is to retrieve the Wikipedia passage that was used to generate said question from a pool of negatives. PAQ's size makes this a suitable large-scale pre-training task and represents a close proxy of the actual downstream open-domain QA retrieval task.

3.1.2 Reddit

For dialogue tasks, we use 200 million post-comment pairs mined from Reddit. This dataset was originally extracted and made available by `pushshift.io` and shown to be useful for dialogue and chit-chat applications previously (Humeau et al., 2019; Roller et al., 2020).

3.2 Evaluation tasks

In this section, we describe our evaluation setup.

3.2.1 Passage retrieval

We evaluate on a mix of standard information retrieval and open-domain question answering benchmarks, and a suite of knowledge-intensive retrieval tasks:

MSMARCO (Nguyen et al., 2016) is a suite of benchmarks created using real user queries to the Bing search engine, with human annotated search results. We evaluate on the passage retrieval task, which is widely reported on in the IR community.

Natural Questions (NQ, Kwiatkowski et al., 2019) is a popular open-domain QA dataset, with questions originating from Google users, and answers annotated from Wikipedia.

KILT (Petroni et al., 2020) is a benchmark consisting of a diverse set of 8 knowledge-intensive tasks, including fact-checking, entity linking, relation extraction, dialogue and question answering. All tasks are grounded in Wikipedia, and we report on the passage selection metrics.

3.2.2 Retrieval for dialogue

We also evaluate on a set of dialogue retrieval benchmarks, to see how far our conclusions generalize to a different domain.

ConvAI2 is based on the PersonaChat dataset (Zhang et al., 2018), and was presented for the NeurIPS ConvAI2 competition (Dinan et al., 2019). The task involves selecting the correct next utterance in a dialogue, out of 20 candidates, given the dialogue history as well as some context about the speakers persona.

Ubuntu v2 (Lowe et al., 2015; Kummerfeld et al., 2019) is a large corpus of 1 million conversations from Ubuntu chat logs, which document users receiving support from other users regarding Ubuntu-related issues.

DSTC7 (Gunasekara et al., 2019) is a challenge set consisting of 100k samples extracted from the Ubuntu dataset described above.

3.3 Implementation

Frameworks We use the Pytorch Lightning (PL) framework (Falcon, 2019) for implementing our models. PL enables effortless scaling to hundreds of GPUs, with memory and speed optimizations such as half-precision training, and sharded gradients during distributed training. We add memory-mapped data loaders, which allow us to scale to datasets with hundreds of millions of query-passage pairs. We use pre-trained encoders provided by the Huggingface transformers library (Wolf et al., 2020).

In-batch negatives Following (Karpukhin et al., 2020) we implement in batch negatives by using the differentiable all gather primitive provided by PL. Unlike the original implementation in (Karpukhin et al., 2020) this lets us gather negatives across all nodes leading to higher training efficiency.

Validation Metrics Evaluating neural retrieval models requires embedding tens of millions of passages for indexing. This is a one-time, manageable cost for deployment systems, however for research iteration and model selection purposes, it is prohibitively expensive. One option is to use proxy-metrics such as validation cross-entropy loss, or in-batch accuracy to do model selection. Unfortunately such metrics often do not correlate well with end-to-end retrieval accuracy. As a middle-ground, we implement distributed in-memory validation using the all gather primitive. This allows us to use a fairly large proxy corpus of up to 300k passages, including up to 50 hard negative examples for each test query. We find that using mean reciprocal rank on this corpus as a model selection metric correlates well with full evaluation metrics.

Training details Pre-training on PAQ and Reddit are run for up to 10 epochs on 64 Nvidia V100 32GB GPUs, with the ADAM optimizer and triangular learning rate schedule. Learning rate and batch size vary for each model, and are presented in the appendix. We fine-tune for up to 40 epochs on the end task on 8 GPUs. For BERT and DeBERTa models, we use the [CLS] token directly as the representation, whereas for RoBERTa we add a linear projection of the same size and an additional layer normalization. BERT models use seperate encoders for query and passage, where RoBERTa and DeBERTa models use shared encoders.

Data preparation For PAQ pre-training, we mined negative examples using a publicly available DPR checkpoint[2]. For Reddit, the engineering effort to setup an index of 200M documents was too large; therefore we pre-train it without negatives. For MSMARCO and KILT, we use standard pre-processing and splits, and for NaturalQuestions we follow (Karpukhin et al., 2020). For the dialogue tasks, we use the dataset-provided negative examples when available. We concatenate all dialogue context (including persona for ConvAI2) to form the query, and truncate from the beginning if it is longer than 256 tokens.

Our code, pre-trained checkpoints, and pre-processed data files are publicly available[3].

4 Main results

In this section, we summarize our main results, before we dive into some analysis in the next section.

4.1 Passage retrieval results

Our main results for passage retrieval are presented in two tables, Table 1 for MSMARCO and NQ, and Table 2 for KILT. PAQ-based pre-training results in strong gains on almost all passage retrieval tasks. For NaturalQuestions, pre-training improves +3.2 points over our non-pretrained baseline on top-20 accuracy, without using iterative training (Figure 2). Our setup with iterative training is most similar to (Xiong et al., 2020), on which pre-training improves by *additional* 2.1 points (81.9 vs. 84.0). We advance the best published results (Qu et al., 2021) by +1.7 points on both top-20 and top-100 accuracy. We note that the main contribuition of (Qu et al., 2021) is using a large cross-encoder model to pre-filter training data - an approach which is orthogonal to pre-training and could provide additional gains. On MSMARCO, we see similar gains, improving +3.8 points over our best non-pretrained baseline.

On KILT, we advance passage retrieval SoTA on all tasks by 6.7 points of R-precision on average. This result shows that PAQ-based pre-training generalizes well across a wide variety of tasks.

4.2 Dialogue retrieval results

To further verify the matched-domain hypothesis, we conduct experiments in the dialogue retrieval domain, using Reddit chat threads as pre-training data. We see clear gains on all datasets over vanilla BERT baselines, affirming the usefulness of additional pre-training for retrieval. However, the gains are less pronounced for UbuntuV2, which has a much larger training dataset. Nevertheless, our best model (RoBERTa$_{large}$) still outperformes the previous SoTA by a comfortable margin on two tasks. For DSTC7, the results also support our conclusions, however we were not able to reproduce previous baselines on this dataset, and our numbers are generally lower.

5 Pre-training retrieval models

In this section we cover our findings regarding how to best pre-train bi-encoder models for retrieval. We compare our pre-training approach with previous approaches, and emphasize the importance of picking the right pre-training task. We discuss the effects of data and model size for pre-training retrieval models.

[2] https://github.com/facebookresearch/dpr, with id checkpoint.retriever.single.nq.bert-base-encoder

[3] https://github.com/facebookresearch/dpr-scale

	Methods	Base model	MSMARCO	Natural Questions		
			MRR@10	R@5	R@20	R@100
1	BM25 (anserini) (Yang et al., 2017)	-	18.7	-	59.1	73.7
2	DPR (single) (Karpukhin et al., 2020)	BERT$_{\text{base}}$	-	65.8	78.4	85.4
3	GAR (Mao et al., 2020)	-	-	-	74.4	85.3
4	ANCE (single) (Xiong et al., 2020)	RoBERTa$_{\text{base}}$	33.0	-	81.9	87.5
5	RocketQA (Qu et al., 2021)	ERNIE$_{\text{base}}$	37.0	74.0	82.7	88.5
6	DPR(ours)	BERT$_{\text{base}}$	29.0	65.5	78.3	85.6
7	DPR(ours)	BERT$_{\text{large}}$	28.8	69.14	80.19	86.73
8	DPR(ours)	RoBERTa$_{\text{base}}$	29.5	67.00	79.03	85.42
9	DPR(ours)	RoBERTa$_{\text{large}}$	30.2	69.67	81.27	87.01
10	DPR(ours)	DeBERTa$_{\text{xlarge-v2}}$	-	72.66	82.38	87.56
11	DPR-PAQ	BERT$_{\text{base}}$	31.4	74.5	83.7	88.6
12	DPR-PAQ	BERT$_{\text{large}}$	31.1	75.3	84.4	88.9
13	DPR-PAQ	RoBERTa$_{\text{base}}$	32.3	74.15	84.01	89.2
14	DPR-PAQ	RoBERTa$_{\text{large}}$	34.0	76.93	84.68	89.22
15	DPR-PAQ	DeBERTa$_{\text{xlarge-v2}}$	-	73.38	83	88.61

Table 1: Passage retrieval results for MSMARCO development set and NaturalQuestions test set.

Methods	Base model	FEV	T-REx	zsRE	NQ	HoPo	TQA	WoW	Avg.
BM25	-	40.1	51.6	53.0	14.2	38.4	16.2	18.4	33.1
Multi-task DPR	BERT$_{\text{base}}$	52.1	61.4	54.1	40.1	41.0	34.2	24.6	43.9
DPR-PAQ	BERT$_{\text{base}}$	61.4	68.4	73.28	44.1	44.6	38.9	26.5	50.6
DPR-PAQ	BERT$_{\text{large}}$	62.8	66.58	66.9	42.6	42.1	37.9	23.4	48.9

Table 2: Paragraph-level R-Precision on the KILT benchmark.

5.1 Picking the pre-training task

As pointed out earlier, previous attempts at pre-training dense retrieval models have largely been ineffective. In Table 4, we confirm this conclusion. We see that BFS and ICT do result in non-trivial zero-shot retrieval performance on the NQ dataset. However, after fine-tuning these gains disappear, and they do not outperform a vanilla BERT model. The performance of PAQ-pretrained retrieval is exceptionally strong even before fine-tuning. This is expected to an extent, since PAQ has been trained on NQ, and many NQ training questions might already appear verbatim in the PAQ generated questions. Nevertheless, pre-training with PAQ results in robust gains, which persist after fine-tuning. Note that both BFS and ICT were pre-trained on more data than PAQ (200 million pairs vs. 65 million). We conclude that PAQ pairs are higher quality, and better matched to the end task than previously proposed artificial pre-training tasks, resulting in better performance.

For the dialogue experiments, we compare against (Humeau et al., 2019), who also pre-trains on the same Reddit corpus, but using a cross-encoder with masked-language-modeling and next-sentence-prediction objectives *a la* BERT (Devlin et al., 2019). This allows us to compare bi-encoder pre-training, with cross-encoder pre-training on the same dataset. Looking at Table 3, we see that bi-encoder pre-training (DPR-Reddit, BERT$_{base}$) performs significantly better than cross-encoder pre-training on the ConvAI2 dataset (rows 5&8). However, the same conclusion does not hold for the larger and more domain-mismatched Ubuntu corpus. (Our RoBERTa-large bi-encoder does improve over (Humeau et al., 2019), but we don't have a corresponding cross-encoder pre-trained baseline for this model.) We conclude that transfer is somewhat fragile for dense retrieval pre-training, and is sensitive to domain and task mismatch.

5.2 Effect of data size

In Figure 2 we investigate the effect of pre-training data size on retrieval performance. We randomly downsample the PAQ pre-training dataset, and plot

	Methods	Base model	ConvAI2	DSTC7		Ubuntu v2	
			R@1	R@1	MRR	R@1	MRR
1	(Wolf et al., 2019)	BERT$_{\text{base}}$	82.1	-	-	-	-
2	(Chen and Wang, 2019)	BERT$_{\text{base}}$	-	64.5	73.5	-	-
3	(Dong and Huang, 2018)	BERT$_{\text{base}}$	-	-	-	75.9	84.8
4	(Humeau et al., 2019)	BERT$_{\text{base}}$	83.3	66.8	74.6	80.6	88.0
5	(Humeau et al., 2019) (Reddit)	BERT$_{\text{base}}$	86.9	70.9	78.1	83.6	90.1
6	DPR (ours)	BERT$_{\text{base}}$	82.4	53.1	62.6	80.6	87.9
7	DPR (ours)	RoBERTa$_{\text{base}}$	84.6	58.4	68.2	84.2	90.4
8	DPR-Reddit	BERT$_{\text{base}}$	88.5	61.5	70.2	82.0	88.8
9	DPR-Reddit	BERT$_{\text{large}}$	88.2	62.0	70.9	81.8	88.7
10	DPR-Reddit	RoBERTa$_{\text{base}}$	88.4	66.5	75.1	85.1	90.9
11	DPR-Reddit	RoBERTa$_{\text{large}}$	90.7	68.2	76.4	86.3	91.7

Table 3: Dialogue retrieval results.

Pre-training data	w/o FT	w/ FT
None	-	78.4
BFS	37.0	75.7
ICT	25.5	77.0
PAQ	78.1	81.6

Table 4: Comparison of different pre-training data, with and without fine-tuning (FT). Metric is top-20 accuracy on NaturalQuestions test set. Baseline is vanilla BERT-base model.

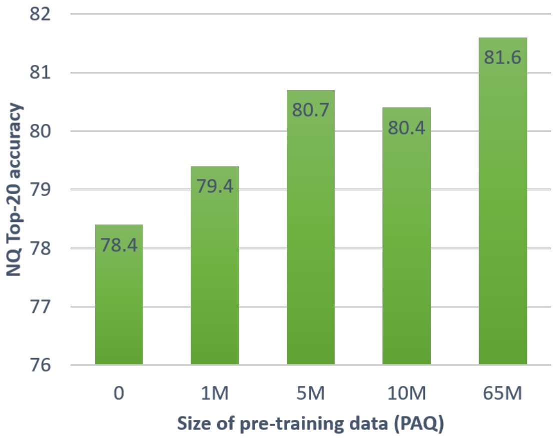

Figure 2: Effect of different sizes of PAQ data for pre-training. Results show top-20 accuracy on NQ after fine-tuning. No iterative pre-training is used.

top-20 accuracy on NQ after fine-tuning on the full NQ training set. We see that as little as 1 million pre-training examples can improve performance, with larger pre-training data resulting in more gains as expected. This suggests that expanding PAQ with even more questions could potentially be beneficial (though this could be contingent on the quality of additional generated questions).

It is interesting to note that additional MLM training is not generally helpful for retrieval on open-domain QA. RoBERTa (Liu et al., 2019) was trained on an order of magnitude more data for much longer compared to BERT, yet fine-tuning on RoBERTa results in little, if any improvement over BERT, in the absence of retrieval-specific pre-training (Table 1, rows 6&8). For dialogue retrieval, better MLM training does help, as was shown previously (Humeau et al., 2019).

5.3 Effect of model size

We experimented with pre-trained models of varying sizes, including BERT(base/large) (Devlin et al., 2019), RoBERTa(base/large) (Liu et al., 2019), and DeBERTa(xlarge-v2) (He et al., 2020). In terms of how better and larger pre-trained models interact with retrieval-specific pre-training, we get mixed results. For instance, for passage retrieval, DeBERTa-xlarge-v2 model does outperform the BERT-base baseline significantly in the fully supervised setting (Table 1, rows 6&10), yet this gain disappears after additional pre-training with PAQ (rows 11&15). The opposite is true when comparing RoBERTa vs. BERT, as we see RoBERTa performing better after intermediate pre-training, both for passage retrieval (rows 12&14) and for dialogue tasks (Table 3, rows 9&11). In contrast to the clear-cut conclusions for other NLP tasks, it is hard to conclude whether larger and better language models actually make better retrieval models.

DPR-PAQ DPR	R@20 ✓	R@20 ✗
R@20 ✓	2.5	3.2
R@20 ✗	3.3	3.1

Table 5: Mean Levenshtein distance to most similar question in PAQ, for DPR-PAQ and a DPR baseline for NQ test questions, stratified by whether the model achieves Recall@20

5.4 Effects of PAQ on Retrieval

In section 4.1, we established that pretraining on PAQ is beneficial for passage retrieval for QA. However, it is worth considering where the source of this improvement lies. Lewis et al. (2021) note that QA-pairs in PAQ have substantial overlap with the test sets of NQ — indeed, this is intentional, given their aim of preempting a large number of probable questions for use as a cache for question answering models. In fact, ~9% of the NQ test questions appear verbatim in PAQ.

It is worth investigating then, whether the gains we observe are due to simply memorizing the relevant passages for PAQ questions which overlap with test questions, or, whether they are due to learning a more robust, generalizable model behavior.

To investigate, we compare the predictions of the DPR-PAQ retriever with an otherwise equal baseline DPR model, without PAQ-pretraining. On the subset of the NQ test set that overlaps verbatim with PAQ questions, we find that DPR-PAQ achieves 95.5% R@20, whereas the baseline achieves 94.8% These are both remarkably high scores, indicating that these verbatim questions are very easy for models to solve, regardless of pretraining. Due to the very similar performance on this subset, the difference in overall performance cannot be attributed to simply memorising verbatim-overlapping questions.

Another analysis we conduct is to check whether the questions that DPR-PAQ does well are those that *look like* the questions in PAQ. Specifically, for each test question, we find the question in PAQ that has the smallest Levenshtein distance to it[4] and record the distance value. For each retrieval model of DPR and DPR-PAQ, we split the test questions into two disjoint sets, based on whether the top 20 retrieved results of each question contain the relevant document (i.e., R@20 = 1). As shown in Table 5, questions that both retrieval models do well indeed look similar to PAQ questions, with a small mean minimum edit distance of 2.5 words. Questions where only one model performs well have higher edit distance. However, there is no big quantitative difference between DPR and DPR-PAQ, with edit distance 3.2 and 3.3 words, respectively. This suggests that the improvement of DPR-PAQ cannot be explained by simply memorizing PAQ questions. Otherwise, questions that DPR-PAQ does well should have a lower edit distance to PAQ questions.

6 Related Work

6.1 Dense retrieval

Lee et al. (2019) was first to show that dense pretrained representations can outperform BM25 for end-to-end retrieval in the context of open-domain QA. This work also proposed the ICT pre-training task for retrieval, and demonstrated its usefulness. Guu et al. (2020) improved on this work, by end-to-end pre-training of retriever and reader using a language modeling loss. It was subsequently shown (Karpukhin et al., 2020) that these sophisticated end-to-end pre-training methods are not necessary, and a fully-supervised fine-tuning of the retriever can produce superior results. The performance of fully-supervised models were improved even further in (Xiong et al., 2020) and (Qu et al., 2021) by iteratively updating negative candidates, using cross-encoder models for increasing the quality of negative candidates, and hyperparameter optimizations. Encoding each passage with multiple vectors, based on dense phrase representations, has also been proposed and shown good retrieval accuracy (Lee et al., 2021).

Pretraining for retrieval Chang et al. (2020) investigate several artificial tasks for training dense retrieval models, including ICT and BFS, showing improvements over no pre-training. However, their setting is not fully open, and they report on a smaller set of 1 million passages. These results have also been superseded by better supervised fine-tuning.

Concurrent work (Sachan et al., 2021) combined ICT pre-training with masked-salient-span pre-training, as well as an end-to-end fine-tuning

[4] To avoid calculating Levenshtein distance over all questions in PAQ, we use the RePAQ question retriever from Lewis et al. (2021), and calculate the minimum distance over the top 100 candidates.

using a T5-large model, obtaining results comparable or slightly better than what is presented here. The major improvements in this work are attributed to end-to-end training, which amounts to a type of distillation from the powerful T5 model into the retrieval model. It is interesting to compare this to more direct distillation methods (Izacard and Grave, 2020; Yang and Seo, 2020), which also reported similar gains. Our method also relies on a reader model indirectly, through the global filtering stage of generated questions in PAQ. However, this is different and more general than mere distillation on a supervised dataset, as it also involves data augmentation at large scale, and generalizes well to other datasets, as shown in section 4.1.

Question generation Lewis et al. (2021), used generated questions as a cache to build a fast lookup-based QA system. Using the same question bank for pre-training, we have shown that we can get additional value and generalisation from this resource. Ma et al. (2021) and Jia et al. (2021) also investigate training on generated QA pairs, but the former only considers application to domain transfer and the latter to other NLP tasks.

7 Conclusion

We have investigated domain-matched pre-training tasks for bi-encoder dense retrieval models. We found that the proposed approach is more effective than previously proposed artificial pre-training tasks. We demonstrated the generality of our conclusions, by evaluating on a large and varied set of passage retrieval and dialogue retrieval benchmarks.

Our work should be considered as a new statement in the ongoing dialogue of how to best train dense retrieval models. We believe we have addressed some important open questions, such as whether and when pre-training can be useful. However we have also raised new questions, in addition to the many which remain open. For instance, many different ways of leveraging reader models for better retrieval have been recently proposed, including end-to-end training, distillation, data filtering and data augmentation. What is the relationship between these approaches? Are they complementary? Which ones are more efficient, and more performant? We believe these questions deserve a more thorough investigation.

We have focused mostly on dense retrieval when full supervision is available, and showed that for $k = 100$ retrieval candidates, the performance is already approaching a ceiling. There is more room for improvement for smaller k. In this regime, however, re-ranking models also become feasible and separable architecture is not a strict requirement. Therefore, further improvements to retrieval will likely need to be discussed with more emphasis on the computation-accuracy trade-off. Few-shot and zero-shot retrieval will also be of increasing importance, and there are already works investigating this direction (Maillard et al., 2021; Thakur et al., 2021).

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*.

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*.

Qian Chen and Wen Wang. 2019. Sequential attention-based network for noetic end-to-end response selection. *arXiv preprint arXiv:1901.02609*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan

Lowe, et al. 2019. The second conversational intelligence challenge (convai2). *arXiv preprint arXiv:1902.00098*.

Jianxiong Dong and Jim Huang. 2018. Enhance word representation for out-of-vocabulary on ubuntu dialogue corpus. *arXiv preprint arXiv:1802.02614*.

et al. Falcon, WA. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

Chulaka Gunasekara, Jonathan K Kummerfeld, Lazaros Polymenakos, and Walter Lasecki. 2019. Dstc7 task 1: Noetic end-to-end response selection. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 60–67.

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*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

Robin Jia, Mike Lewis, and Luke Zettlemoyer. 2021. Question answering infused pre-training of general-purpose contextualized representations. *arXiv preprint arXiv:2106.08190*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

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)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick 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)*, pages 6769–6781, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy. Association for Computational Linguistics.

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

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. Phrase retrieval learns passage retrieval, too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661–3672, Online and Punta Cana, Dominican Republic. 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 Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *arXiv preprint arXiv:2102.07033*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan 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*, pages 1075–1088.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oğuz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. *arXiv preprint arXiv:2101.00117*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

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 *CoCo@ NIPS*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. *arXiv preprint arXiv:2101.00408*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256.

Sohee Yang and Minjoon Seo. 2020. Is retriever merely an approximator of reader? *arXiv preprint arXiv:2010.10999*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

A Appendices

A.1 Pre-training hyperparameters

Encoder	**lr**	**bs**
BERT$_{base}$	2.5e-5	32
BERT$_{large}$	1e-5	12
RoBERTa$_{base}$	2e-5	40
RoBERTa$_{large}$	1e-5	12
DeBERTa$_{xlarge}$	1e-5	12

Table 6: Learning rate and batch size for pre-training.