Improving the Generalizability of the Dense Passage Retriever Using Generated Datasets

Thilina C. Rajapakse(✉) and Maarten de Rijke

University of Amsterdam, Amsterdam, The Netherlands
{t.c.r.rajapakse,m.derijke}@uva.nl

Abstract. Dense retrieval methods have surpassed traditional sparse retrieval methods for open-domain retrieval. While these methods, such as the Dense Passage Retriever (DPR), work well on datasets or domains they have been trained on, there is a noticeable loss in accuracy when tested on out-of-distribution and out-of-domain datasets. We hypothesize that this may be, in large part, due to the mismatch in the information available to the context encoder and the query encoder during training. Most training datasets commonly used for training dense retrieval models contain an overwhelming majority of passages where there is only one query from a passage. We hypothesize that this imbalance encourages dense retrieval models to *overfit* to a single potential query from a given passage leading to worse performance on out-of-distribution and out-of-domain queries. To test this hypothesis, we focus on a prominent dense retrieval method, the dense passage retriever, build generated datasets that have multiple queries for most passages, and compare dense passage retriever models trained on these datasets against models trained on single query per passage datasets. Using the generated datasets, we show that training on passages with multiple queries leads to models that generalize better to out-of-distribution and out-of-domain test datasets.

1 Introduction

Recently, a number of transformer-based dense retrieval models have achieved state-of-the-art results on various benchmark datasets [13,14,28]. The Dense Passage Retriever (DPR) architecture consists of two encoder models, typically BERT models [8], which encode the query and the passages separately. A simple similarity metric, such as the inner product or cosine distance, is then used to compute the relevance of a passage for a query.

An advantage of the DPR architecture is that passage representations can be pre-computed offline and built into an index with relatively small computational cost, making it a preferred model over recent proposals such as, e.g., ColBERT [14] and ANCE [28] with higher computational cost for training and/or retrieval. At runtime, the query encoder is used to compute a dense representation for the query and approximate nearest neighbor methods are used to find the most relevant passage.

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13981, pp. 94–109, 2023.
https://doi.org/10.1007/978-3-031-28238-6_7

A disadvantage of this approach is that a mismatch may exist between the information available to the passage encoder and the information available to the query encoder. As the training objective forces the passage and query encoders to generate representations that are similar, we hypothesize that the passage encoder (which has access to more information) learns to discard information that is not relevant to the query in a given training query-passage pair. The issue is exacerbated by the fact that most retrieval datasets and benchmarks contain far more passages with only one query from a given passage than passages with multiple queries per passage (see Table 1). In such situations, the model is not sufficiently penalized against learning to discard information that is not relevant to the (single) query that is asked from a given passage.

We hypothesize that a DPR model trained on datasets where a given passage typically has one associated query generalizes poorly to other datasets, new types of queries or topics, or both. We investigate this hypothesis by testing the zero-shot performance of the pretrained DPR model (from [13], which is trained on NQ [16]) in both out-of-distribution and out-of-domain settings. Here, we define *out-of-distribution* to be datasets that share the same passage corpus but with queries collected at different times and/or using different methods, and *out-of-domain* to be datasets with their own unique passage collection typically focused on a particular domain (see Sect. 4.1).

Having established that a DPR model trained on datasets where a given passage typically has one associated query, generalizes poorly, we propose a treatment to help improve out-of-distribution and out-of-domain performance. We synthetically generate training datasets where the passages typically have multiple queries from any given passage. The generation pipeline consists of a NER model to tag entities, a sequence-to-sequence model to generate queries, and a question answering model to filter out bad queries (see Sect. 3.1).

Our results show that training on data with multiple queries per passage leads to a DPR model with better generalizability to both out-of-distribution and out-of-domain data. In both settings, our DPR model trained on multiple queries per passage data easily outperforms the baseline DPR model trained on mostly single query per passage data (NQ).

In summary, then, we answer the following research question:

RQ Does training a DPR model on data containing multiple queries per passage improve the generalizability of the model?

In the out-of-distribution setting, the pre-trained DPR model [13], serving as the baseline, and our DPR model trained on generated queries with multiple queries per passage are tested, zero-shot, on six datasets. Our model achieves higher retrieval accuracy on five out of the six datasets demonstrating that training data containing multiple queries per passage does improve the generalizability of dense retrievers to out-of-distribution queries.

The picture becomes even clearer in the out-of-domain setting where our model outperforms the pretrained DPR model on 12 out of 13 datasets. Training DPR models on passages with multiple associated queries prevents the context encoder from (exclusively) focusing on a specific detail or piece of information in the passage, leading to a better generalized retrieval model.

Our analysis of increasing the size of the set of generated queries with multiple queries per passage as a way to improve the generalizability of dense retrievers indicates a subtle balance. While the model trained on the largest training dataset does achieve higher scores compared to the others, the improvements are relatively minor. But, these relatively minor improvements come at a significantly higher costs in terms of compute and training time. Even the smallest generated dataset with multiple queries per passage performs competitively with larger generated datasets and handily outperforms the pre-trained model trained on mostly single query per passage data.

2 Related Work

Passage Retrieval. Passage retrieval has classically been performed using sparse retrieval methods such as BM25 [25]. Recently, transformer-based dense retrieval methods have garnered interest as the performance of dense retrieval methods surpasses that of traditional sparse methods [13,14,28]. A dense passage retriever indexes a collection of passages in a low-dimensional and continuous space, such that the top-k passages are relevant to a given query [13]. Here, the size of the passage collection is typically very large (21M passages in this work and in [13]) and k is very small (e.g., 20–100). Going beyond *in-distribution* and *in-domain* testing, we focus on generalizability to new data which can be *out-of-distribution* and *out-of-domain*.

Test Collections. The Benchmarking-IR (BEIR) [22] test collection was introduced to facilitate the effectiveness of retrieval models in out-of-domain settings. It provides a collection of 18 datasets (13 of which are readily available) from diverse retrieval tasks and domains. Thakur et al. [22] also highlight considerable room for improvement in the generalization capabilities of dense retrieval models. Our work aims to improve the generalizability of dense retrievers by using synthetic datasets with specially chosen composition of data (multiple queries per passage).

Automatically Generated Collections. Automatically generating training, development and test collections for retrieval has a long history in information retrieval. Examples include test collections for bibliographic systems [21], known-item test collections [2], desktop search [15], web search [1], test collections for academic search [3]. Berendsen et al. [4] focus on test collection generation to improve robustness for tuning and learning. A comprehensive approach to simulated test collection building with considerable attention to privacy preservation is offered in [11]. What we add on top of this is test collection building with a specific focus on generalizability by preventing overfitting.

3 Methodology

We train DPR models on generated query datasets and compare their retrieval performance against the pre-trained model on the test datasets.

3.1 Dataset Generation Process

For our dataset generation process, we follow the steps below:

(1) Identify potential answers to questions to be generated;
(2) Generate queries that are answered by one of the potential answers; and
(3) Filter out bad queries, that is, queries that are unanswerable or do not end with a question mark.

Identifying Potential Answers. We train a token classification model to identify words or phrases from a passage that could serve as potential answers to queries. The trained model is then used to tag potential answers for each passage in a dataset. This process enables us to find all potential answers in a passage, which is critical to ensure that there are sufficient queries from any given passage.

Generating Queries. The passages, along with the tagged answers, are fed to a sequence-to-sequence model that generates a query for each passage-answer pair. Each passage can have multiple associated answers, resulting in multiple queries from the same passage. This ensures that there are queries related to most, if not all, entities found in a given passage.

Filtering Queries. The generated queries are filtered to remove potentially unanswerable queries (from the originating passage). To find such queries, we feed the passages and queries to a question answering (QA) model and discard queries where the QA model answer does not match the original tagged answer. We also discard queries that contain more than one sentence or do not end with a question mark (?). This is to ensure that all the generated queries used for training are reasonable queries (see Sect. 4.2) and provide a good training signal for the model being trained on them.

3.2 Training the Retriever

We build training datasets by generating queries following the procedure given in Sect. 3.1. The generation process ensures that most passages in the training datasets have multiple queries associated with them. We train bi-encoder retrieval models on these training datasets.

4 Experimental Setup

4.1 Datasets

Most popular open-domain retrieval datasets contain a much larger number of passages with only a single query originating from it than passages with multiple queries. Table 1 shows the frequency of passages with a given number of queries originating from the passage for the five datasets used in [13] as well as the five datasets that were generated. The Wikipedia collection and five of the datasets used (NQ, Trivia QA, Curated TREC, Web Questions, and SQuAD) are the same versions provided by [13] available on GitHub.[1]

Table 1. Frequency of passages with a given number of queries originating from the passage.

Dataset	Number of queries/passage		
	1	2	≥2
Natural questions	32,155	4,973	3,542
Trivia QA	43,401	5,308	1,793
Curated TREC	990	41	16
Web Questions	2,019	148	46
SQuAD	8,468	6,056	11,790
Generated from NQ train	2,784	3,418	30,120
Wikipedia passages (˜58k) single	58,880	0	0
Wikipedia passages (˜58k) multi	16,634	19,641	985
Wikipedia passages (˜236k)	19,487	18,061	41,308
Wikipedia passages (˜786k)	62,264	60,472	137,266

Out-of-Distribution Test Datasets. To test the models on out-of-distribution data, we use the four datasets available from [13] that were not used in training the baseline model, namely Trivia QA, Curated TREC, Web Questions, and SQuAD. In addition to these four, we include two generated test datasets. The first of these is generated from the NQ *dev* passages and the second is generated from randomly selected Wikipedia passages. This results in a total of six out-of-distribution test datasets. As these datasets use the same passage collection but contain queries collected or generated using different approaches, we consider the datasets to be *out-of-distribution* but *in-domain*.

[1] https://github.com/facebookresearch/DPR.

Out-of-Domain Test Datasets. We use the 13 readily available datasets from [22], each with their own distinct passage collection, to test the models on out-of-domain data. The datasets are as follows: TREC-COVID [24], NFCorpus [6], HotpotQA [29], FiQA-2018 [18], ArguAna [26], Touché-2020 [5], CQADupStack [12], Quora, DBPedia [10], SCIDOCS [7], FEVER [23], Climate-FEVER [9], and SciFact [27]. These datasets cover multiple domains, including bio-medical, Wikipedia/general, finance, news, and scientific domains.

4.2 Generation Pipeline

Named Entity Recognition Model for Tagging Answers. The named entity recognition model is a RoBERTa [17] model trained on the large NER dataset (1 million sentences) from Naman Jaswani on Kaggle,[2] with the tags: *Organization*, *Person*, *Location*, *Date*, *Time*, *Money*, *Percent*, *Facility*, and Geo-Political Entity (GPE). The RoBERTa model, trained on a large NER dataset, ensures that we find all the entities in a passage.

MACAW Model for Query Generation. The pretrained *MACAW* [20] model (3 billion parameters) is used to generate the queries. It is a strong sequence-to-sequence question generation model (among other tasks) based on the T5 model [19]. This model is capable of generating queries for each entity found in the passage such that they are relevant to the context of the passage.

Table 2. Examples of generated queries and answers for a randomly sampled passage.

Passage	Generated query	Generated answer	Related	Answerable
Sirocco (play) Sirocco is a play, in four acts, by Noël Coward. It originally opened at Daly’s Theatre, on November 24, 1927. The production was directed by Basil Dean. Ivor Novello was part of the original cast. The plot told a tale of free love among the wealthy. The London opening of “Sirocco” met with violently unfavorable audience reaction and a very harsh critical reception. Coward was later asked whether he had ever despaired when faced with a failure like “Sirocco”. He replied, “Well, if I’m going to have a flop, I like it to be a rouser. I didn’t	Sirocco was first performed at which theater in London?	Dalys Theatre	Yes	Yes
	When did the first performance of Sirocco take place?	November 24 1927	Yes	Yes
	Which actor played the role of Sirocco in the original production?	Ivor Novello	Yes	No
	Who wrote the play Sirocco?	Noël Coward	Yes	Yes
	Who directed the first production of Sirocco?	Basil Dean	Yes	Yes

[2] https://www.kaggle.com/namanj27/ner-dataset.

Question Answering Model for Query Filtering. A RoBERTa [17] model trained on the SQuAD dataset is used to filter out potential bad queries in the generated datasets. The RoBERTa model is a question answering model that is good at extractive question answering. We can reasonably assume that the questions the model is incapable of answering are most likely flawed.

This generation pipeline results in queries that are typically relevant and answerable from their passages of origin. We found 92% of queries to be relevant, and 86% to be answerable from their passages of origin, based on a randomly sampled set of 50 queries (example shown in Table 2).

4.3 Retrieval Pipeline

The architecture of the retrieval model is identical to [13], i.e., a bi-encoder architecture consisting of two BERT [8] encoders, one for encoding the passages/contexts and the other for encoding the queries. We also use the same hyperparameters as [13] except for the batch size, where we use a batch size of 80 vs. a batch size of 120 due to resource limitations.

We choose the DPR [13] model as our architecture of choice to avoid introducing any confounding factors in our analysis. Other architectures, notably the late interaction based ColBERT [14] architecture, has demonstrated superior retrieval accuracy over the original DPR [13] architecture. However, ColBERT has higher latency and much larger space footprints for indices. As our work is focused on the composition of data, the simpler and more straightforward architecture of DPR is better suited to our analysis. Furthermore, the higher resource demands and complexity of ColBERT makes it a less viable option compared to DPR in any setting with even moderate computational resource constraints.

We build five training datasets by generating queries following the procedure given in Sect. 3.1. One dataset is built by generating queries from the same passages used in the NQ train set, while the other four are from randomly selected Wikipedia passages. A bi-encoder DPR model, starting from the pretrained BERT [8] weights, is trained on each of these five datasets.

While positive training examples (matching query and document pairs) are available directly in retrieval datasets, negative training examples must be selected from the set of all documents. The original DPR model is trained using a combination of in-batch negatives (the positive documents of all other queries in the batch used as negatives for a given query) and BM25 selected negatives (highest ranked document retrieved by BM25, which does not contain the answer to the query). In our work, we simply use the in-batch negatives as the negative examples leaving improvements from more complex negative selection strategies for future work as our results demonstrate improved generalizability even without using hard negatives.

4.4 Experiment

We use two models trained on two different datasets to compare the generalizability of DPR models trained on data with multiple queries per passage versus DPR models trained on data with mostly a single query per passage. The pre-trained DPR model from [13], trained on NQ with mostly single query per passage data, is used as the baseline model to be compared against our model trained 58,880 generated queries containing mostly multiple queries per passage data (*58k generated*).

The two models are tested in both the out-of-distribution (6 datasets) and the out-of-domain settings (13 datasets). *Top-100 accuracy* is used as the evaluation metric for the out-of-distribution setting while *recall@100* is used as the evaluation metric for the out-of-domain setting. The decision to use two different metrics is motivated by the fact that the set of all relevant passages is only available for the out-of-domain datasets, which is necessary to calculate recall. Only the true answers are available for the out-of-distribution datasets, so we calculate top-100 accuracy by checking whether the true answer is present in any of the top-100 retrieved documents. In addition to this, we also report MRR@100 (Mean Reciprocal Rank) for all experiments.

5 Results

We report results from the baseline pretrained model trained on NQ (58,880 queries) against our model trained on 58,880 generated queries for the two generalizability settings; out-of-distribution and out-of-domain. Here, the generated query dataset contains mostly passages with multiple queries per passage.

5.1 Out-of-Distribution Generalizability

Table 3 shows the top 100 accuracy scores obtained by the baseline DPR model (trained on NQ) and our DPR model, trained on the 58k generated query dataset with multiple queries per passage (*58k generated*), on the out-of-distribution datasets. We also include the scores on the NQ dataset itself for completeness, but it should be noted that this dataset is an in-distribution dataset for the baseline model.

Table 3. Top 100 accuracy scores for the model trained on *58k generated* and the baseline DPR model trained on NQ for out-of-distribution datasets. The highest score is in **bold** and ‡ indicates in-domain performance. Statistical significance with paired t-test: * indicates $p < 0.05$ and ** indicates $p < 0.01$.

Model	Standard datasets					Generated datasets	
	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev.	Wikipedia
Baseline DPR	**84.9**$^{\ddagger **}$	78.7	**90.7**	77.6	63.5	81.5	56.7
58k generated (ours)	75.0	**80.0****	89.6	**78.3**	**69.4****	**85.3****	**79.2****

The model trained on *58k generated* (our model) outperforms the baseline DPR model on 5 out of 6 out-of-distribution datasets, with the Curated TREC

Table 4. MRR@100 scores for the model trained on *58k generated* and the baseline DPR model trained on NQ for out-of-distribution datasets. Same notational conventions as in Table 3.

Model	Standard datasets					Generated datasets	
	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev.	Wikipedia
Baseline DPR	**0.512**$^{\ddagger **}$	**0.437****	**0.583****	**0.389****	0.234	**0.449****	0.240
58k generated (ours)	0.313	0.426	0.507	0.358	**0.258****	0.426	**0.415****

dataset being the sole exception. However, the difference in accuracy between the two models on Curated TREC and WebQ are not statistically significant. Our model generalizes better in all four datasets (out of six) where the difference is statistically significant. The baseline DPR model does better on the NQ test dataset (in-distribution) compared to the our model trained on generated queries (out-of-distribution).

Interestingly, the baseline DPR model trails our model trained on *58k generated* even on the queries generated from the NQ passages despite being trained on fairly similar data. This indicates that the performance of DPR models trained on data with mostly a single query from each passage deteriorates rapidly when tested on new queries. This observation may be explained by our initial hypothesis. If a model trained on data with a single query per passage learns to discard information, it is logical that the model would struggle when dealing with multiple queries from a passage as this requires the context encoder to encode all information available in the passage in order to correctly match all the queries from that passage. These results indicate that training a model on data with multiple queries per passage results in improved generalizability in the out-of-distribution setting.

The baseline model outperforms the model trained on *58k generated* on 4 out of 6 out-of-distribution datasets when considering MRR@100 scores (Table 4). However, the *58k generated* model performs slightly better on average.

5.2 Out-of-Domain Generalizability

Table 5 shows the recall@100 scores obtained by the baseline DPR model (trained on NQ) and our DPR model trained on *58k generated*. The model trained on *58k generated* outperforms the baseline DPR model achieving higher recall@100 scores in 12 out of 13 out-of-domain datasets. Considering only the statistically significant results ($p < 0.05$), our model trained on multiple query per passage data outperforms the baseline DPR model on all 10 out of 10 datasets.

The MRR@100 scores (Table 5) follow a similar pattern, with the model trained on *58k generated* outperforming the baseline in 9 out of 10 out-of-domain datasets where the results are statistically significant.

The model trained with data containing multiple queries per passage (our model trained on *58k generated*) dominates the baseline DPR model, trained on mostly single query per passage data, in both the out-of-distribution and out-of-domain setting. This clearly superior zero-shot generalization performance when

Table 5. Recall@100 and MRR@100 scores for the baseline DPR model trained on NQ and the model trained on 58k generated queries for out-of-domain datasets. Same notational conventions as in Table 3.

Dataset	Recall@100		MRR@100	
	Baseline DPR	58k generated	Baseline DPR	58k generated
ArguAna	0.480	**0.919****	0.051	**0.213****
Climate FEVER	**0.410**	0.405	**0.258****	0.220
CQA dup stack	0.109	**0.139****	0.041	**0.068****
DBPedia	0.310	**0.335***	0.559	**0.564**
FEVER	0.748	**0.805****	**0.497**	0.492
FiQa	0.313	**0.369****	0.131	**0.195****
HotpotQA	0.493	**0.502**	0.419	**0.559****
NFCorpus	0.170	**0.238**	0.306	**0.377****
Quora	0.566	**0.880****	0.279	**0.590****
SciDocs	0.196	**0.253****	0.136	**0.207****
SciFact	0.581	**0.704****	0.247	**0.372****
Touche	0.276	**0.344****	0.234	**0.386****
TREC-COVID	0.096	**0.177****	0.287	**0.354**

a DPR model is trained on data with multiple queries per passage answers our research question (RQ) demonstrating that training a DPR model on data with multiple queries per passage does result in a better generalized model.

6 Analysis

6.1 Generation Versus Data Composition

We conduct a further analysis to confirm that the improvements in generalizability shown in Sect. 5 is due to the composition of the dataset, specifically the number of queries per passage, rather than any artifact of the query generation process. Here, we compare the generalizability to out-of-distribution and out-of-domain data of two models trained on generated queries. The first model is trained on generated queries with multiple queries per passage (same as in Sect. 5) and the second model is trained on generated queries with only a single query from each passage.

Table 6 shows the top-100 accuracy scores obtained by the two models on the out-of-distribution datasets. The model trained on *58k generated (multi)* outperforms the model trained *58k generated (single)* on 5 out of 7 datasets (one loss and one tie). Four of these results are statistically significant with the model trained on *58k generated (multi)* generalizing better in all four cases. Similarly, the model trained on *58k generated (multi)* outperforms the model trained on *58k generated (single)*, in terms of MRR@100 scores (Table 7), on all six out-of-distribution datasets with four of the results being statistically

Table 6. Top 100 accuracy scores for the models trained on *58k generated (single)* and *58k generated (multi)* for out-of-distribution datasets. Same notational conventions as in Table 3.

Model	Standard datasets					Generated datasets	
	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev	Wikipedia
58k generated (single)	**75.0**	78.4	**90.2**	77.5	67.9	81.9	74.5
58k generated (multi)	**75.0**	**80.0****	89.6	**78.3**	**69.4****	**85.3****	**79.2****

Table 7. MRR@100 scores for the models trained on *58k generated (single)* and *58k generated (multi)* for out-of-distribution datasets. Same notational conventions as in Table 3.

Model	Standard datasets					Generated datasets	
	NQ	TriviaQA	TREC	WebQ	SQuAD	NQ dev	Wikipedia
58k generated (single)	0.309	0.397	0.489	0.350	0.247	0.394	0.366
58k generated (multi)	**0.313**	**0.426****	**0.507**	**0.358**	**0.258****	**0.426****	**0.415****

significant. These results clearly show that having multiple queries per passage in the training data helps the model generalize better to out-of-distribution queries, as the only difference between the two models is the composition of the training data.

Table 8 shows the recall@100 scores obtained by the two models on the out-of-domain datasets. Again, the model trained with multiple queries per passage outperforms the model trained on single query per passage data and generalizes

Table 8. Recall@100 and MRR@100 scores for the models trained on *58k generated (single)* and *58k generated (multi)* for out-of-domain datasets. Same notational conventions as in Table 3.

Dataset	Recall@100		MRR@100	
	58k generated (single)	58k generated (multi)	58k generated (single)	58k generated (multi)
ArguAna	0.885	**0.919****	0.208	**0.213**
Climate FEVER	0.378	**0.405****	0.188	**0.220****
CQA Dup Stack	0.134	**0.139****	**0.068**	**0.068**
DBPedia	0.312	**0.335****	0.545	**0.564**
FEVER	0.722	**0.805****	0.415	**0.492****
FiQa	0.358	**0.369**	0.189	**0.195**
HotpotQA	0.430	**0.502****	0.460	**0.559****
NFCorpus	0.185	**0.238**	0.376	**0.377**
Quora	**0.909****	0.880	**0.658****	0.590
SciDocs	0.246	**0.253**	0.202	**0.207**
SciFact	0.685	**0.704**	0.346	**0.372**
Touche	**0.371**	0.344	0.343	**0.386**
TREC-COVID	**0.181**	0.177	0.300	**0.354**

better to 10 out of 13 out-of-domain datasets. Looking at the statistically significant results, the model trained on *58k generated (multi)* does better on 6 out of 7 datasets. The results on the remaining six datasets are likely not statistically significant as they contain a very small number of queries.

Overall, the model trained on *58k generated (multi)* generalizes better, in both out-of-distribution and out-of-domain settings, compared to the model trained on *58k generated (single)* when all other factors are kept constant. This confirms that the composition of training data, specifically the number of queries per passage, is an important factor to consider when training dense retrieval models and that training on data with multiple queries per passage leads to a model that is capable of generalizing better to out-of-distribution and out-of-domain queries.

6.2 Effect of Dataset Size

We also investigate the effect of the total number of generated queries in a training dataset on the generalizability of DPR models. For this analysis we compare three DPR models trained on three generated query datasets, where each dataset contains 58,880 (*58k generated*), 236,444 (*236k generated*), and 786,312 (*786k generated*) queries respectively. Note that all three of these datasets contain data with multiple queries per passage. Again, we report zero-shot scores in both the out-of-distribution and out-of-domain settings.

Table 9 shows the top-100 accuracy scores obtained by each model on the out-of-distribution datasets. The model trained on *786k generated* generalizes better to all seven datasets, with five of the results being statistically significant. In terms of MRR@100 (Table 10), the model trained on *786k generated* obtains higher scores on 5 out of 6 datasets, with four being statistically significant. These results indicate that training on larger datasets, containing data with multiple queries per passage, does yield better results on out-of-distribution datasets in a zero-shot setting.

Table 9. Top 100 accuracy scores for the models trained on the three generated query datasets *58k*, *236k*, and *786k* for out-of-distribution datasets. Same notational conventions as in Table 3.

Model	Standard datasets					Generated datasets	
	NQ	TriviaQA	TREC	WQ	SQuAD	NQ dev	Wikipedia
58k Generated	75.0	80.0	89.6	78.3	69.4	85.3	79.2
236k Generated	79.5	82.5	91.7	80.6	71.6	90.1	85.4
786k Generated	**80.5***	**83.2****	**92.2**	**80.7**	**72.9****	**92.4****	**89.4****

Table 11 shows the recall@100 scores obtained by each model on the out-of-domain datasets. Overall, the model trained on the largest dataset, *786k generated*, does marginally better than the other two models, obtaining the highest recall@100 score for seven out of thirteen out-of-domain datasets. The other

Table 10. MRR@100 scores for the models trained on the three generated query datasets *58k*, *236k*, and *786k* for out-of-distribution datasets. Same notational conventions as in Table 3.

Model	Standard Datasets					Generated Datasets	
	NQ	TriviaQA	TREC	WQ	SQuAD	NQ dev	Wikipedia
58k Generated	0.313	0.426	0.507	0.358	0.258	0.426	0.415
236k Generated	0.339	0.467	0.515	**0.381**	0.274	0.493	0.488
786k Generated	**0.360****	**0.492****	**0.526**	0.379	**0.283****	**0.522****	**0.542****

two models, trained on *236k generated* and *58k generated*, achieve the highest scores in four out of thirteen and two out of thirteen, respectively. Only three of these results are statistically significant with the model trained on *786k generated* doing better on two and the model trained on *58k generated* performing better on the other. The MRR@100 scores (Table 11) are even more mixed, with the model trained on *236k genrated* performing better in 2 out of 4 statistically significant results while the other two models perform better on one each.

Table 11. Recall@100 and MRR@100 scores for the model trained on the three generated query datasets *58k generated*, *236k generated*, and *786k generated* for the out-of-domain datasets. Same notational conventions as in Table 3.

Dataset	Recall@100			MRR@100		
	58k *generated*	236k *generated*	786k *generated*	58k *generated*	236k *generated*	786k *generated*
ArguAna	0.919	0.939	**0.940**	**0.213**	0.209	0.202
Climate FEVER	0.405	**0.406**	0.371	0.220	**0.224**	0.198
CQA Dup Stack	0.139	**0.154**	0.153	0.068	**0.072****	0.069
DBPedia	0.335	0.362	**0.364**	**0.564**	**0.564**	**0.564**
FEVER	0.805	0.853	**0.856**	0.492	**0.508****	0.476
FiQa	0.369	**0.385**	0.377	**0.195**	0.190	0.171
HotpotQA	0.502	0.557	**0.572****	0.559	0.598	**0.603**
NFCorpus	**0.238**	0.216	0.216	0.377	**0.387**	0.382
Quora	0.880	0.897	**0.929****	0.590	0.613	**0.636****
SciDocs	0.253	0.253	**0.261**	0.207	**0.212**	0.198
SciFact	0.704	0.737	**0.790**	0.372	0.373	**0.374**
Touche	0.344	**0.366**	0.325	**0.386**	0.325	0.314
TREC-COVID	**0.177****	0.124	0.119	**0.354***	0.219	0.166

While larger training datasets help with zero-shot performance on out-of-distribution datasets, the benefit of more generated data is less clear with regard to zero-shot performance on out-of-domain datasets. Although the model trained on *786k generated* generalizes better than the other two models, the increase in recall scores are marginal, especially compared to the increased cost of training which increases linearly with dataset size. Overall, training DPR models on more generated queries with multiple queries per passage can improve the generalizability of the model, but with sharply diminishing gains. This is likely due to the fact that increasing the size of the training dataset does not necessarily increase the diversity of the training data.

7 Conclusion and Future Work

We have shown that the generalizability of dense passage retrievers may suffer from learning to discard information from passages during training. This problem can be mitigated by using training data containing a sufficient number of passages with multiple associated queries. By exposing the dense retriever to multiple facets of information contained in the same passage, we ensure that the model does not learn to discard potentially useful information, leading to improved retrieval accuracy for out-of-domain topics and queries and a better-generalized model overall.

As a general lesson, when training a dense retrieval model, it is important to consider the number of queries per passage, or more generally, how much of the information contained in a given passage is covered by the queries. Training datasets with a large number of queries per passage can be automatically generated for training dense retrievers resulting in a better generalized model.

As to limitations, we did not use hard negative mining [28] or late interaction [14], which are known to improve the generalizability of dense retrievers. We leave their integration to future work but note that our method is trivially compatible with such techniques and is also independent of the actual dense retriever architecture that is used.

Finally, it would be interesting to use our proposed dataset generation method on a full collection of Wikipedia passages to train a DPR model. While our analysis of the effect of dataset size (Sect. 6.2) did not demonstrate meaningful gains in generalizability, a sufficiently large query collection (a generated query dataset of the full Wikipedia collection would be several orders of magnitude larger) containing diverse topics may generalize very well to most domains.

Acknowledgements. This research was supported by the Dreams Lab, a collaboration between Huawei, the University of Amsterdam, and the Vrije Universiteit Amsterdam, and by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

1. Asadi, N., Metzler, D., Elsayed, T., Lin, J.: Pseudo test collections for learning web search ranking functions. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1073–1082. ACM (2011)

2. Azzopardi, L., de Rijke, M.: Automatic construction of known-item finding test beds. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, pp. 603–604. ACM (2006)
3. Berendsen, R., Tsagkias, M., de Rijke, M., Meij, E.: Generating pseudo test collections for learning to rank scientific articles. In: Catarci, T., Forner, P., Hiemstra, D., Peñas, A., Santucci, G. (eds.) CLEF 2012. LNCS, vol. 7488, pp. 42–53. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33247-0_6
4. Berendsen, R., Tsagkias, M., Weerkamp, W., de Rijke, M.: Pseudo test collections for training and tuning microblog rankers. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 53–62. ACM (2013)
5. Bondarenko, A., et al.: Overview of touché 2022: argument retrieval. In: International Conference of the Cross-Language Evaluation Forum for European Languages, pp. 311–336. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-13643-6_21
6. Boteva, V., Gholipour, D., Sokolov, A., Riezler, S.: A full-text learning to rank dataset for medical information retrieval. In: Ferro, N., et al. (eds.) ECIR 2016. LNCS, vol. 9626, pp. 716–722. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30671-1_58
7. Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D.: SPECTER: Document-level representation learning using citation-informed transformers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 2270–2282. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.207, https://aclanthology.org/2020.acl-main.207
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding (2018). arXiv preprint arXiv:1810.04805
9. Diggelmann, T., Boyd-Graber, J., Bulian, J., Ciaramita, M., Leippold, M.: Climate-fever: a dataset for verification of real-world climate claims (2020). https://doi.org/10.48550/ARXIV.2012.00614, https://arxiv.org/abs/2012.00614
10. Hasibi, F., et al.:Dbpedia-entity v2: a test collection for entity search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, pp. 1265–1268. Association for Computing Machinery, New York (2017) . https://doi.org/10.1145/3077136.3080751
11. Hawking, D., Billerbeck, B., Thomas, P., Craswell, N.: Simulating Information Retrieval Test Collections. Morgan & Claypool (2020)
12. Hoogeveen, D., Verspoor, K.M., Baldwin, T.: Cqadupstack: a benchmark data set for community question-answering research. In: Proceedings of the 20th Australasian Document Computing Symposium, pp. 1–8 (2015)
13. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781. Association for Computational Linguistics (2020)
14. Khattab, O., Zaharia, M.: 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, pp. 39–48 (2020)
15. Kim, J., Croft, W.B.: Retrieval experiments using pseudo-desktop collections. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1297–1306. ACM (2009)

16. Kwiatkowski, T., et al.: Natural questions: a benchmark for question answering research. Trans. Assoc. Comput. Linguist. **7**, 453–466 (2019)
17. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
18. Maia, M., et al.: Www'18 open challenge: financial opinion mining and question answering. In: Companion Proceedings of the The Web Conference 2018, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, WWW 2018, pp. 1941–1942 (2018). https://doi.org/10.1145/3184558.3192301
19. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
20. Tafjord, O., Clark, P.: General-purpose question-answering with macaw. arXiv preprint arXiv:2109.02593 (2021)
21. Tague, J., Nelson, M., Wu, H.: Problems in the simulation of bibliographic retrieval systems. In: Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval, Butterworth & Co., pp. 236–255 (1980)
22. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: a heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021). https://openreview.net/forum?id=wCu6T5xFjeJ
23. Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A.: FEVER: a large-scale dataset for fact extraction and VERification. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long Papers), pp. 809—819. Association for Computational Linguistics, New Orleans (2018). https://doi.org/10.18653/v1/N18-1074, https://aclanthology.org/N18-1074
24. Voorhees, E., et al.: Trec-covid: constructing a pandemic information retrieval test collection. SIGIR Forum **54**(1) (2021). https://doi.org/10.1145/3451964.3451965
25. Voorhees, E.M.: The trec-8 question answering track report. In: Proceedings of TREC-8, pp. 77–82 (1999)
26. Wachsmuth, H., Syed, S., Stein, B.: Retrieval of the best counterargument without prior topic knowledge. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 241–251. Association for Computational Linguistics, Melbourne (2018). https://doi.org/10.18653/v1/P18-1023, https://aclanthology.org/P18-1023
27. Wadden, D., et al.: Fact or fiction: verifying scientific claims. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7534–7550. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.609, https://aclanthology.org/2020.emnlp-main.609
28. Xiong, L., et al.: Approximate nearest neighbor negative contrastive learning for dense text retrieval (2020). arXiv preprint arXiv:2007.00808
29. Yang, Z., et al.: HotpotQA: a dataset for diverse, explainable multi-hop question answering. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2369–2380. Association for Computational Linguistics, Brussels (2018). https://doi.org/10.18653/v1/D18-1259, https://aclanthology.org/D18-1259