AugTriever: Unsupervised Dense Retrieval by Scalable Data Augmentation

Anonymous ACL submission

Abstract

Dense retrievers have made significant strides in text retrieval and open-domain question answering. However, most of these achievements have relied heavily on extensive human-annotated supervision. In this study, we aim to develop unsupervised methods for improving dense retrieval models. We propose two approaches that enable annotation-free and scalable training by creating pseudo query-document pairs: query extraction and transferred query generation. The query extraction method involves selecting salient spans from the original document to generate pseudo queries. On the other hand, the transferred query generation method utilizes generation models trained for other NLP tasks, such as summarization, to produce pseudo queries. Through extensive experimentation, we have demonstrated that models trained using these augmentation methods can achieve comparable, if not better, performance than multiple strong dense baselines. Moreover, combining these strategies leads to further improvements, resulting in superior performance of unsupervised dense retrieval, unsupervised domain adaptation and supervised fine-tuning, benchmarked on both BEIR and ODQA datasets[1].

1 Introduction

Text retrieval is currently one of the most influential artificial intelligence applications. Through common internet services like web search and product search, billions of users access vast amounts of data on the Internet, benefiting from information retrieval techniques. While traditional lexical retrieval remains a simple yet effective solution, neural network-based models, particularly dense retrievers, have made significant advancements in recent years, showcasing their advantages in scenarios that involve semantic matching.

However, the majority of dense retrievers heavily depend on training with a large volume of annotated data. For example, MS MARCO (Nguyen et al., 2016) and Natural Questions (Kwiatkowski et al., 2019) are the two most widely used datasets, and models trained on these datasets have achieved exceptional performance. Nevertheless, each of these datasets comprises hundreds of thousands of query-document pairs annotated by humans, making the collection process prohibitively expensive, and the models trained on them may not generalize well to unseen domains (Thakur et al., 2021). The challenge of training dense retrieval models without human-annotated data remains unsolved.

Recent efforts have shown promising results in training dense retrievers in an annotation-free manner (Izacard et al., 2021; Ram et al., 2021; Neelakantan et al., 2022). Following the conventional paradigm of self-supervised learning, a pretext task is designed by considering two different views of a single document as a positive pair. Subsequently, a dual-encoder model is trained using contrastive learning, aiming to map the two views of the data to similar hidden representations. However, when directly applied on downstream retrieval tasks, these unsupervised models tend to perform worse than the classic method BM25. Nevertheless, a performance boost is observed when fine-tuning the models with annotated positive pairs. This observation motivates us to investigate the gap between the pretext task and downstream retrieval tasks.

Existing strategies for constructing positive pairs are often heuristic in nature. For example, Contriever (Izacard et al., 2021) randomly samples two text spans from a document to form a positive pair. It is evident that the quality of the positive pairs is poorly controlled, and the resulting pseudo queries bear little resemblance to real-world queries. Consequently, the models are adversely affected by the noisy pseudo pairs, leading to inferior performance on down-stream tasks.

[1] Code and datasets will be publicly available at anonymous.url.

In this study, we propose two novel strategies for constructing pseudo query-document pairs without any retrieval related supervisions. We summarize our contributions as follows:

1. We introduce query extraction (**QEXT**), a novel data augmentation method for training dense retrievers. Given a document, we sample a list of random spans and utilize various techniques to determine their salience. The spans with the highest scores are selected as pseudo queries.

2. We propose transferred query generation (**TQGEN**), where pseudo queries are generated using generation models trained for other NLP tasks, such as summarization, unlike previous studies that heavily rely on human annotated data for training query generation models. To the best of our knowledge, this is the first study demonstrating that the inductive bias from other NLP tasks can be leveraged for training dense retrievers.

3. We contribute two datasets, namely **AUGQ-WIKI** and **AUGQ-CC**, which consist of 22.6M and 52.4M pseudo query-document pairs for unsupervised retrieval training.

4. Extensive experiments show that retrievers using **QEXT** and **TQGEN**, referred to as **AUGTRIEVER**, achieve superior performance and beat strong baselines on BEIR and open-domain QA benchmarks. The results showcase the effectiveness of the proposed augmentation methods as means for retrieval pretraining and domain adaptation, without the need for any human-annotated queries/questions.

2 Background

2.1 Bi-encoder Dense Retriever

We employ a Transformer based bi-encoder architecture and contrastive learning to train our dense retrievers (Karpukhin et al., 2020; Xiong et al., 2020; Izacard et al., 2021). Specifically, we utilize two transformers, denoted as E_q and E_d, to encode queries q and documents d, respectively. These decoders generate low-dimensional vectors by performing average pooling over the output embeddings of the top layer. The similarity score between q and d is computed by taking the inner product of the two vectors. The encoder parameters are initialized with BERT-base (Devlin et al., 2019) and are shared between the encoders.

The model is optimized using a contrastive objective, where other documents in the same batch are treated as negative examples. Alternatively, recent works (Izacard et al., 2021; Yang et al., 2021; Xu et al., 2022) use a momentum encoder and a large vector queue to enable the use of additional negative examples. We refer to the architecture that utilizes negative examples in the same batch as **INBATCH**, while the architecture with a momentum document encoder is referred to as **MOCO**.

2.2 Construction of Pseudo Query-Document Pairs for Unsupervised Text Retrieval

Various methods have been proposed to construct pseudo query-document pairs for training unsupervised dense retrievers. We summarize some of these methods below, while additional related methods can be found in references (Shen et al., 2022; Zhao et al., 2022).

- INVERSE CLOZE TASK (Lee et al., 2019): A sentence is randomly selected from a given document, and a retriever is trained to retrieve the document using the sentence as a query.
- Masked salient span in REALM (Guu et al., 2020): REALM is a retrieval-augmented language model. During its pre-training phase, a retriever and a generator work together to predict a masked named entity.
- Random cropping (RANDOMCROP) in CONTRIEVER (Izacard et al., 2021): For a given document *d*, two random spans (contiguous subsequences) are independently extracted from *d* to create a positive pair.
- SPIDER (Ram et al., 2021): This method selects two passages within a document that contain identical n-grams (recurring spans) as a positive pair. It should be noted that this method may not be as data-efficient, as recurring spans may not be present in all documents, particularly shorter ones.
- CPT (Neelakantan et al., 2022): Positive pairs are constructed by using neighboring text pieces from the same document.
- SPAR Λ (Chen et al., 2021): The dense lexical model Λ is trained with questions or random sentences as queries, paired with the top K passages retrieved by BM25.

3 Method

In this section, we present several data argumentation methods for generating pseudo queries from a given document without the need for annotated queries or questions. We apply those methods on Wikipedia passages and CommonCrawl web documents, resulting in two large augmented datasets

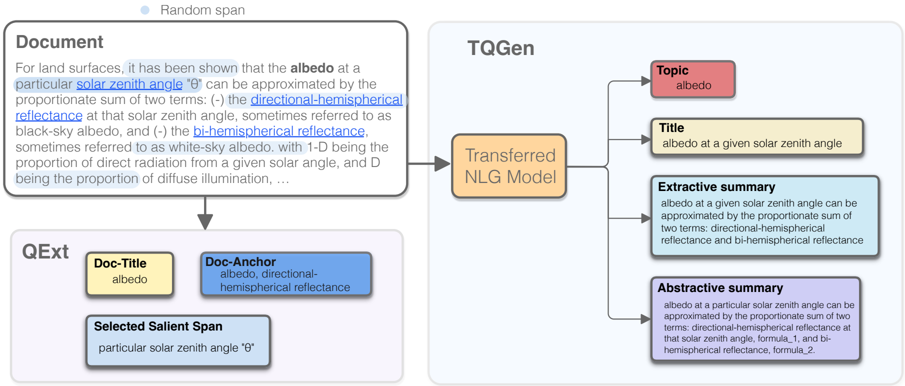
Random span
Document
For land surfaces, it has been shown that the albedo at a particular solar zenith angle "θ" can be approximated by the proportionate sum of two terms: (-) the directional-hemispherical reflectance at that solar zenith angle, sometimes referred to as black-sky albedo, and (-) the bi-hemispherical reflectance, sometimes referred to as white-sky albedo. with 1-D being the proportion of direct radiation from a given solar angle, and D being the proportion of diffuse illumination, …
QExt
Doc-Title
albedo
Doc-Anchor
albedo, directional-hemispherical reflectance
Selected Salient Span
particular solar zenith angle "θ"
TQGen
Transferred NLG Model
Topic
albedo
Title
albedo at a given solar zenith angle
Extractive summary
albedo at a given solar zenith angle can be approximated by the proportionate sum of two terms: directional-hemispherical reflectance and bi-hemispherical reflectance
Abstractive summary
albedo at a particular solar zenith angle can be approximated by the proportionate sum of two terms: directional-hemispherical reflectance at that solar zenith angle, formula_1, and bi-hemispherical reflectance, formula_2.

Figure 1: An overview of proposed augmentation methods for AugTriever.

called AugQ. Subsequently, we train bi-encoder dense retrievers using AugQ and refer to the resulting models AugTriever. These models are trained using either InBatch or MoCo.

3.1 Query Extraction (QExt)

Given a document, we hypothesize that certain parts of it contain more representative information. Therefore, we extract and utilize these parts as pseudo queries to train the retrievers.

3.1.1 Query Extraction Using Document Structural Heuristics

Documents often have rich structures, and extracting information based on these structures for weak supervision has been shown to be effective (Chang et al., 2019; Zhou et al., 2022; Wu et al., 2022). Following this line of research, we propose utilizing the document structure to construct weakly annotated queries for training the dense retriever. Specifically, we consider two types of information, *titles* (Doc-Title) and *anchor texts* (Doc-Anchor). Titles and anchor texts are similar to search queries and are commonly available on the internet. They are considered representative of the core content of the document by the document authors. Extracting titles and anchor texts can be achieved using DOM structures and human-crafted heuristics.

3.1.2 Query Extraction Using Salient Span Selection

The previous method heavily relies on the quality and availability of distant labels embedded in the document structures, which may limit the scalability of training. To address this limitation, we propose an alternative approach that directly extracts informative spans from a document. The hypothesis is that a document can be segmented into multiple spans, and some of these spans are more representative than others. We can then employ various methods to select the most salient spans as pseudo queries. It is important to note that we do not mask the selected spans in the document.

Formally, given a document *d*, we randomly sample a number of text spans $s_1, s_2, ..., s_N$ from it. In this study, we consider 16 random spans, with lengths ranging from 4 to 16 words. We propose three approaches to measure the salience between *d* and each of these spans:

- QExt-Self: This method selects spans by leveraging the model itself. We input each span s_i paired with *d* into the model and use the dot-product as the salience score.
- QExt-BM25: This method selects spans based on lexical models. BM25 is a widely used method for measuring the lexical relevance between queries and documents. Here, we utilize BM25 to select spans based on their lexical statistics.
- QExt-PLM: This method selects spans using pre-trained language models (PLMs). PLMs have shown remarkable performance in text understanding and generation tasks. In our approach, we utilize PLMs to measure the relevance by assessing how likely a span can be generated given a document as the context. Specifically, we feed the document as a prefix to a T5-small LM-Adapted model (Raffel et al., 2020) and use the likelihood $p(s_i|d)$ as the salience score for the span.

3.2 Transferred Query Generation (TQGen)

Previous studies have demonstrated the effectiveness of query generation as a means of augmenting training data (Lewis et al., 2021; Nogueira et al., 2019). However, these approaches typically require a significant amount of annotated data to train a query generator. In our work, we propose a different approach, utilizing text generation models of irrelavant tasks to produce pseudo queries as distant supervision. We hypothesize that the inductive bias of these tasks can effectively bootstrap the training of dense retrievers. Specifically, we leverage models for summarization and keyphrase generation, as these outputs are commonly considered relevant and representative of the source text. Other options include paraphrasing or back-translation.

For implementation, we use a single T0 model (3B parameters) as a meta generator, eliminating the need for selecting models for each generation task. We provide T0 with various prompts to generate outputs for different tasks, including:

- TQGEN-TOPIC: *What is the main topic of the text above?*
- TQGEN-TITLE: *Please write a title of the text above.*
- TQGEN-ABSUM (Abstractive summary): *Please write a short summary of the text above.*
- TQGEN-EXSUM (Extractive summary): *Please use a sentence from the above text to summarize its content.*

We intentionally include a prompt for extractive summaries to encourage the model to use words from the original text and reduce the risk of hallucination. We employ nucleus sampling to generate a single pseudo query for each document, with parameters Top-p=0.9 and Top-K=0. We explore two hybrid settings by combining multiple strategies: *Hybrid-All* which uses all proposed strategies, and *Hybrid-TQGen* which mainly uses TQGEN.

4 Experiments

4.1 Datasets

Training Data: To generate augmented query-document data (AUGQ), we utilize two large text datasets: Wikipedia[2] and CommonCrawl by Pile (Gao et al., 2020) (Pile-CC).

For Wikipedia, we process the original text dump by segmenting articles into paragraphs by line breaks and reserving titles and anchor texts (texts with hyperlinks, italics, or boldface). This results in a total of 22.6 million paragraphs available for training. Pile-CC consists of 52.4 million web documents, but it does not provide structure information, making DOC-ANCHOR unavailable. For DOC-TITLE, we extract the first line of each document as its title, truncating it to a maximum of 64 words. We manually inspected a few hundred examples and found that it correctly extracted titles in approximately 50% of the cases.

Test Data: We use two benchmarks for evaluation: BEIR (Thakur et al., 2021) and six Open-Domain Question Answering (ODQA) datasets. We consider BEIR to be a better benchmark for information retrieval as it covers a broader range of domains and a wide variety of query types. We discuss the scores of MS MARCO (MM) separately since it is one of the most extensively studied IR test sets. On the other hand, all ODQA datasets are based on Wikipedia and primarily designed for evaluating question answering systems. Therefore, they may introduce certain domain and task biases. We utilize these datasets for retrieval evaluation following previous studies (Karpukhin et al., 2020; Ram et al., 2021). We report scores on SQuAD v1.1 (SQ) and EntityQuestions (EQ) separately, as they tend to favor lexical models, while the other four datasets may favor semantic matching approaches.

4.2 Implementation Details

Our models AUGTRIEVER use either **INBATCH** or **MOCO** architecture as the backbone, initialized with BERT-base (Devlin et al., 2019). We adopt most of the settings used by CONTRIEVER (Izacard et al., 2021) for unsupervised training, with a smaller training scale considering the number of model variants we experiment with. Please refer to Sec A.1 for more details.

4.3 Baselines

We consider several unsupervised dense methods discussed in Section 2.2 as baselines. These include BM25 as a lexical baseline, and five dense baselines: CONTRIEVER, SPIDER, SPAR Λ (Wikipedia version), LaPraDor (no BM25) (Xu et al., 2022) and CPT (Neelakantan et al., 2022). We report their scores if publically available (BEIR results of BM25 and CONTRIEVER), or reproduce the results using public code and checkpoints (indicated with †). MOCO+RANDOMCROP can be regarded as our reproduced CONTRIEVER in a smaller scale. We also include baselines with generated queries (using a supervised Doc2Query (Nogueira et al., 2019)) and questions PAQ (Lewis et al., 2021), referred to as QGEN-D2Q and QGEN-PAQ respectively. We rerun most baselines on Touché-2020 (v2) since the data has been updated in BEIR.

[2] enwiki-20211020-pages-articles-multistream.xml.bz2

Group	Model	MM	BEIR14	CPT-subset	QA4	SQ&EQ
Baseline	BM25	**22.8**	**43.0**	**46.1**	70.7†	**71.3†**
	LaPraDor	16.9	30.2	33.6	-	-
	SPAR†	19.3	37.3	41.4	69.1	67.7
	Spider†	15.0	28.1	31.3	73.0	63.6
	Contriever	20.6	36.9	40.7	**73.1†**	63.9†
	CPT-text S	-	-	42.2	-	-
	CPT-text M	-	-	43.2	-	-
	CPT-text L	-	-	44.2	-	-
RandomCrop(Wiki)	InBatch	14.0	25.3	28.0	61.9	44.9
	MoCo	17.5	30.9	34.1	64.6	52.5
RandomCrop(CC)	InBatch	16.3	27.4	30.6	73.2	57.7
	MoCo	19.2	34.0	37.5	71.5	61.9
QGen-D2Q(Wiki)	InBatch	**25.4**	38.5	42.5	**78.6**	67.1
	MoCo	23.7	38.5	42.3	77.4	**67.4**
QGen-D2Q(CC)	InBatch	24.4	39.5	42.8	75.6	63.2
	MoCo	23.2	**39.8**	**43.7**	76.6	65.6
AUGTRIEVER-CC						
Doc-Title	InBatch	19.7	33.2	36.5	73.5	59.7
	MoCo	21.8	38.7	42.7	74.8	64.3
QExt-PLM	InBatch	16.2	27.2	40.4	73.4	57.9
	MoCo	20.6	38.2	42.3	73.0	64.1
TQGen-Topic	InBatch	20.7	39.0	43.0	71.6	60.5
	MoCo	21.2	38.9	43.1	73.3	63.4
TQGen-AbSum	InBatch	18.1	35.3	38.6	72.0	57.4
	MoCo	**23.2**	**39.6**	**43.5**	74.4	**64.9**
TQGen-ExSum	InBatch	18.9	36.3	39.6	72.8	58.7
	MoCo	23.0	39.4	43.4	**74.8**	**64.9**
AUGTRIEVER-CC with Hybrid Strategies						
Hybrid-All	MoCo	23.5	39.4	43.3	74.1	64.4
Hybrid-TQGen	MoCo	23.3	39.0	43.7	74.3	64.4
Hybrid-TQGen+	MoCo	**24.6**	41.1	45.2	76.0	65.9
Hybrid-TQGen++	MoCo	**25.4**	**42.1**	**46.2**	**76.2**	**67.1**
AUGTRIEVER-Wikipedia						
Doc-Anchor	InBatch	15.6	29.8	33.2	64.8	49.9
	MoCo	17.9	35.4	39.2	68.5	57.4
Doc-Title	InBatch	14.7	30.0	33.9	61.9	52.1
	MoCo	18.5	33.7	37.1	68.6	58.4
QExt-PLM	InBatch	15.0	26.3	28.2	61.5	43.8
	MoCo	18.6	34.3	37.8	66.6	55.7
TQGen-Topic	InBatch	21.3	**38.9**	**43.2**	72.4	64.4
	MoCo	21.3	38.3	42.5	73.6	65.3
TQGen-AbSum	InBatch	17.4	36.3	40.2	74.9	65.3
	MoCo	21.2	37.2	41.3	74.5	65.4
TQGen-ExSum	InBatch	18.2	36.7	40.2	**75.6**	65.4
	MoCo	**22.5**	37.9	41.8	75.1	**66.7**

Table 1: Unsupervised retrieval performance (MM/BEIR nDCG@10 and ODQA Recall@20). MM denotes scores on MS MARCO. QA4 denotes averaged scores of NQ, TQA, WebQ and TREC. We highlight the best and second best in each column, and **best** in each group per column.

4.4 Results

4.4.1 Unsupervised Retrieval

We present the main unsupervised results in Table 1 and will discuss certain details in Sec 4.5. Among all unsupervised baselines, BM25 still outperforms the other baselines by a significant margin. For dense retrievers, the lexical-oriented retriever SPAR Λ performs the best on BEIR14 and SQ&EQ, indicating that dense retrievers can achieve robust retrieval performance through a lexical teacher. CONTRIEVER performs comparably with SPAR Λ on BEIR. The supervised augmentation QGEN-D2Q delivers competitive results on both benchmarks, suggesting that query generation trained with MS MARCO can work well both in-domain and out-of-domain.

Regarding AUGTRIEVER[3], we find that multiple variants, e.g. MoCo+QExt-PLM and TQGen models, significantly outperform dense baselines on BEIR, especially when trained with the domain-general data CC. Notably, our best results are achieved by hybrid strategies and longer training (Hybrid-TQGen+/++). Hybrid-TQGen++ outperforms CPT-text L on BEIR by a large margin, despite CPT-text L being 20 times larger than AUGTRIEVER models. These empirical results strongly suggest the effectiveness of the proposed method for unsupervised dense retrieval.

Besides, (1) we observe that TQGEN achieves the overall best performance, indicating that the outputs of transferred NLP tasks, such as keyword and summary generation (Meng et al., 2017; See et al., 2017), can be utilized for training dense retrieval models effectively; (2) It is worth noting that TQGEN-TOPIC generalizes well under all settings, suggesting that keywords can serve as robust surrogate queries. (3) MOCO+QEXT-PLM outperforms all dense baselines on BEIR, indicating that query extraction can be an effective unsupervised method. However, since it scores random spans using an LM on-the-fly, we are unable to scale it up (larger batch size, better scorer) in this study.

4.4.2 Unsupervised Domain-Adaptation

Retrieval models are often applied to data of new domains, making domain adaptation crucial in real-world scenarios. We investigate the effectiveness of the proposed augmentation methods for domain adaptation. We leverage TQGEN-TOPIC method for its simplicity and overall great performance in Table 1. We generate main topics (keywords) for documents in each BEIR domain (test set), and then fine-tune Hybrid-TQGen++ with the in-domain pseudo query-document pairs. We compare our model with various baselines reported by Thakur et al.; Wang et al.; Yu et al.. Note that several baselines use MS MARCO pairs for training models (TAS-B), query generators (QGen and GPL) and rerankers (GPL), while in contrast our models have not used any retrieval related data.

The results are presented in Table 2. We observe an 8% average gain over 14 BEIR datasets, suggesting the importance of adapting models using in-domain documents. Significant improvements (up to 16%) are seen on domains that are specific and distant from the pretraining distribution, such as finance (FiQA) and argument (Tóuche-2020, ArguAna). Furthermore, our model, in spite of

[3] We train InBatch models using augmented query-document pairs only, whereas we train MoCo models using a 50/50 mixed strategy (50% of pairs by RANDOMCROP and 50% by one of the proposed augmentation strategies). Further comparisons between the two settings are discussed in Sec 4.5.

	BEIR14	TREC-COVID	NFCorpus	NQ	HotpotQA	FiQA	ArguAna	Touche	DBPedia	Scidocs	FEVER	Cli-FEVER	Scifact	Quora	CQADup
Unsupervised															
BM25	43.0	65.6	32.5	32.9	60.3	23.6	31.5	36.7	31.3	15.8	75.3	21.3	66.5	78.9	29.9
COCO-DR$_{Base}$	29.5	43.9	23.7	9.9	23.4	17.5	43.1	11.7	15.6	11.2	25.9	8.9	71.1	79.3	27.6
TSDAE	43.6	70.8	31.2	47.1	63.8	29.3	37.5	21.8	35.4	15.4	64.0	16.8	62.8	83.3	31.8
MS MARCO involved															
TAS-B	42.7	48.5	31.9	46.3	58.4	29.8	43.4	16.2	38.4	14.9	69.5	22.1	63.5	83.5	31.5
QGen	42.1	56.0	31.4	35.4	51.4	28.7	52.4	17.1	33.1	15.5	63.8	22.5	63.8	85.0	33.0
TSDAE+QGen	43.4	58.4	33.7	34.6	52.2	31.4	54.7	17.2	33.2	17.1	64.2	22.6	66.7	85.7	35.3
TAS-B+QGen	42.8	56.6	33.4	36.3	52.0	30.1	51.8	17.5	32.7	16.4	63.9	24.4	65.3	85.3	33.7
DocT5-Query	45.3	71.3	32.8	39.9	58.0	29.1	46.9	34.7	33.1	16.2	71.4	20.1	67.5	80.2	32.5
GPL	46.5	71.8	34.2	46.7	56.5	32.8	48.3	23.1	36.1	16.1	77.9	22.7	66.4	83.2	34.5
Ours (Unsupervised)															
Hybrid-TQGen++	42.1	64.0	35.1	33.5	57.6	29.2	39.8	19.8	32.7	16.7	65.3	15.9	68.7	79.9	30.6
+DA	45.4	68.1	36.8	33.7	60.8	34.0	46.2	23.3	34.9	17.2	72.0	18.4	70.3	85.4	34.5
DIFF%	8.0%	6.4%	4.8%	0.7%	5.6%	16.3%	16.3%	17.6%	6.9%	2.4%	10.3%	15.5%	2.4%	6.9%	12.7%

Table 2: Results of domain-adapation on 14 BEIR testsets. We highlight the best and second best scores.

the simple method being used, outperforms BM25 and most neural domain adaptation methods. GPL outruns our method with the advantage of using MS MARCO trained query generator and reranker, whereas our models merely use pseudo keywords for training, presenting a simple yet effective approach for domain adaptation. Here we did not explore how augmentation strategies other than TQGEN-TOPIC would work on DA, and we think the pair filtering used by GPL can also help our models. We leave them for future work.

Model	MM	BEIR14
Baseline		
BM25	22.8	43.0
DPR	35.4	36.8
ANCE	38.8	40.5
ColBERT	40.1	44.4
Spider †	24.8	19.1
LaPraDor †	38.9	40.4
SPAR †	38.0	41.5
Condenser(Book&Wiki) †	38.7	40.9
CoCondenser(MSMARCO) †	40.8	42.9
Contriever †	**41.3**	**45.2**
Ours		
QGen-D2Q	39.6	43.9
RandomCrop	38.4	42.4
QExt-PLM	38.8	42.5
TQGen-Topic	38.8	43.2
TQGen-Title	38.8	43.2
TQGen-AbSum	39.1	43.9
TQGen-ExSum	38.8	43.1
Hybrid-All	38.9	43.6
Hybrid-TQGen+	40.3	44.7
Hybrid-TQGen++	**40.9**	**45.8**

Table 3: Retrieval scores after fine-tuning with MS MARCO. AUGTRIEVER models are pre-trained on AUGQ-CC using MoCo. We highlight the best and second best in each column. † indicates results by us, fine-tuned using public checkpoints.

4.4.3 Fine-Tuning with MS MARCO

To assess the effectiveness of the proposed augmentation methods as pretraining measures, we present the fine-tuned results on MS MARCO in Table 3. We use basic fine-tuning settings without employing advanced techniques such as negative mining (Izacard et al., 2021) or asynchronous index refresh (Xiong et al., 2020). We compare with multiple baselines reported in BEIR (Thakur et al., 2021) (BM25, DPR (Karpukhin et al., 2020), ANCE, ColBERT (Khattab and Zaharia, 2020)) and we fine-tune the other pretrained models under the same setting (Spider, LaPraDor, Condenser (Gao and Callan, 2021), CoCondenser (Gao and Callan, 2022), and Contriever).

Among all baselines, ColBERT and Contriever perform best on BEIR overall, indicating the benefit of late-interaction and extensive pre-training. Most AUGTRIEVER models demonstrate equal or better performance to baselines, indicating the effectiveness of using AUGQ for pretraining. Our best model, Hybrid-TQGen++, achieves best score on BEIR (45.8), showing strong generalization performance when zero-shot transferred on 14 different datasets. It falls behind CONTRIEVER on MS MARCO by a small margin (0.4 point), and we think it can be attributed to the fact that CONTRIEVER was pre-trained with both Wikipedia and CommonCrawl data (Hybrid-TQGen++ scores 50.3 on five Wikipedia related datasets, comparing to CONTRIEVER's score 51.1), and its training duration was longer (ours 200k steps vs. CONTRIEVER 500k steps). In most cases, the trends in the fine-tuned scores align with the unsupervised results, providing strong evidence that the inductive bias from various augmentation methods can benefit downstream retrieval tasks.

4.5 Result Analysis

To gain a deeper understanding of how individual augmentation strategies contribute to retrieval tasks and their performance in specific scenarios, we conducted a detailed analysis. We present the results of AUGTRIEVER (trained on AUGQ-WIKI) and base-

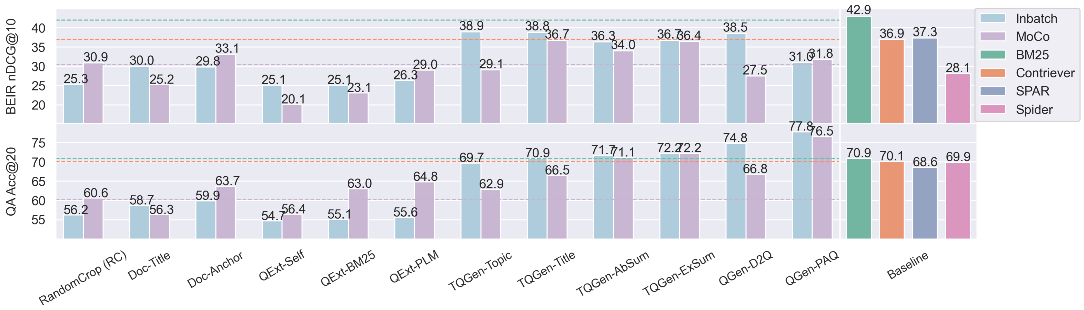

Figure 2: AUGTRIEVER performance with individual augmentation strategies. The upper shows averaged nDCG@10 scores of BEIR benchmark, and the lower shows averaged Recall@20 scores over 6 ODQA datasets. Dashed lines indicate the scores of BM25, CONTRIEVER and MOCO+RC.

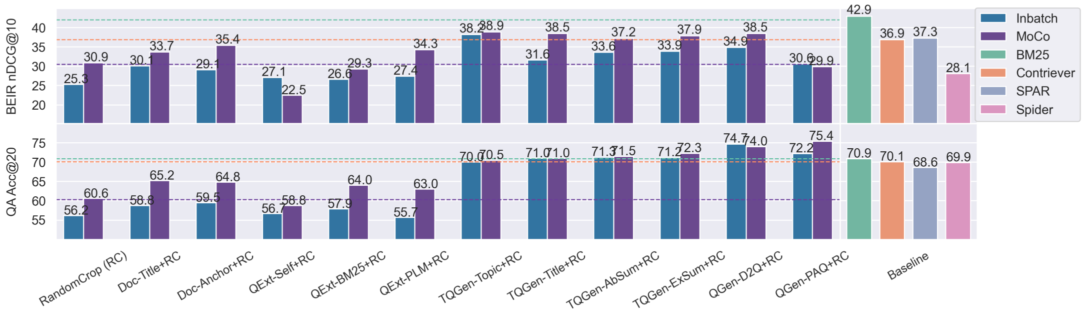

Figure 3: AUGTRIEVER performance with hybrid strategies (50/50 mix of training examples from RANDOMCROP and another augmentation).

line models in Fig 2, averaging the scores across 14 BEIR datasets and 6 ODQA datasets. Notably, while BM25 remains a competitive unsupervised baseline and leads among all models, we made the following observations:

1. BEIR is a more comprehensive benchmark for evaluating retrieval models. In general, the trends observed on BEIR align with those on ODQA. However, ODQA datasets are specifically designed for question-answering using Wikipedia, which introduces certain domain and task biases. For example, QGEN-PAQ, which is trained with 65M generated query-document pairs on Wikipedia, excels on ODQA (in-distribution) but fails to generalize well on BEIR. On the other hand, BEIR covers a wider range of domains and topics, making it a more suitable benchmark for evaluating models' generalization ability. Thus, we consider BEIR to be a more indicative benchmark for text retrieval evaluation and focus our discussion on it.

2. Among all AUGTRIEVER variants, TQGEN achieves the highest scores, significantly outperforming all dense baselines. This finding strongly suggests that the outputs of language generation tasks, such as keyword/title/summary generation, can be directly utilized during the training of dense retrieval models. Interestingly, shorter pseudo queries (TOPIC/TITLE) perform better on BEIR compared to QGEN-D2Q. In contrast, longer ones (ABSUM/EXSUM) are more effective on ODQA, likely due to their resemblance to questions by including more details. EXSUM slightly outperforms ABSUM, possibly because it tends to use original text and has fewer hallucinations.

3. InBatch benefits from queries of higher quality, while MoCo performs well with noisy queries. The RANDOMCROP strategy generates noisy queries, including incomplete sentences and non-informative text. However, MoCo is able to achieve good results with such noisy queries, indicating that a momentum encoder can provide robustness against noisy pairs. Conversely, InBatch performs notably better with "cleaner" queries (TQGEN and QGEN), highlighting the advantages of each architecture.

4. Combining RANDOMCROP with other high-quality queries benefits MoCo, but not InBatch. In Figure 3, we observe that MoCo consistently improves performance by incorporating individual augmentation methods with RANDOMCROP. This demonstrates MoCo's ability to leverage multiple strategies and enhance its generalization capability. However, the mixed strategy provides little benefit or even leads to a performance drop for InBatch, which aligns with our previous argument regarding the characteristics of each architecture. We also find that the mixed strategy helps MoCo achieve decent results with QExt methods (DOC-TITLE, DOC-ANCHOR, and QEXT-PLM), although they still lag behind TQGEN by a significant margin.

5 Related Work

Recent years have seen a flourishing of research works for neural network based information retrieval and question answering. The interested reader may refer to (Lin et al., 2021; Guo et al., 2022; Zhao et al., 2022) for a comprehensive overview. Our study, along with a line of recent studies (Izacard et al., 2021; Ram et al., 2021), falls under the category of self-supervised learning using contrastive learning (Shen et al., 2022), in which a model is trained to maximize the scores of positive pairs and minimize the scores of negative ones. It has demonstrated effective for supervised dense retrieval (Karpukhin et al., 2020; Xiong et al., 2020; Liu et al., 2021) and pretraining (Izacard et al., 2021; Yu et al., 2022). Different from most prior studies, we target at unsupervised models that can be independently applied in retrieval tasks, without any further tuning using annotated data.

Previous works propose different ways to construct query-document pairs to fit the requirement of contrative learning. Lee et al. propose inverse cloze task (ICT), using a random sentence as a pseudo query to predict the surrounding context in a batch. REALM (Guu et al., 2020) pretrains a retriver and generator with a pair of a salient span (named entities) and its context. Spider (Ram et al., 2021) proposes to use recurring spans as pseudo queries. The above studies focus on ODQA tasks and their pseudo queries tend to be entity-like, but results of this study and Izacard et al. show that entity-like queries (e.g. anchor texts) fail to generalize well in a broad range of domains. Some studies propose more generic ideas for training unsupervised models. Specifically, Neelakantan et al.; Ma et al. use neighboring pieces of text as positive pairs. Izacard et al. adopt a random cropping strategy to sample two text spans, encouraging the model to learn lexical matching. Chen et al. use random sentences or real questions as queries and pair them with documents ranked by BM25.

A few research works investigate techniques of data augmentation and domain adaptation for text retrieval and understanding (Tang et al., 2022; Wang et al., 2022; Iida and Okazaki, 2022). Query and question generation have been shown as an effective method for augmenting retrieval training data (Thakur et al., 2021; Nogueira et al., 2019; Lewis et al., 2021; Ma et al., 2021a; Gangi Reddy et al., 2022; Cho et al., 2022; Liang et al., 2020). GPL (Wang et al., 2022) uses cross-encoder to select a good set of synthetic query-document pairs for domain adaptation. InPars and Promptagator (Bonifacio et al., 2022; Dai et al., 2022) propose to generate questions using large language models in a few-shot manner. LaPraDoR (Xu et al., 2022) propose to use Dropout-as-Positive-Instance for pretraining retrievers. CERT (Fang et al., 2020) uses positive pairs generated by back-translation. HyDE (Gao et al., 2022) uses large language models to augment user queries with generated pseudo documents. For hyperlinks, Chang et al. compare three pretraining tasks for retrieval – inverse cloze task, body first selection, and wiki link prediction. Zhou et al.; Wu et al.; Xie et al. utilize hyperlinks to construct pseudo query-document pairs. PROP (Ma et al., 2021b,c) uses a representative words prediction task to optimize the semantic distance between a document and a pair of random word sets, estimated by language models. Recent studies UPR (Sachan et al., 2022a) and ART (Sachan et al., 2022b) use pretrained language models for reranking and Open-domain QA, using the likelihood of question generation to approximate the relevance between questions and documents. DRAGON (Lin et al., 2023) systematically examine supervised training of dense retriever under the framework of data augmentation.

6 Discussion and Conclusion

In this study, we propose a set of scalable augmentation methods to generate surrogate queries for training dense retrievers without the need for annotated query-document pairs. Our approach achieves great performance on widely used benchmarks (BEIR and six ODQA tasks). These results highlight the effectiveness of extracted and transfer generated query-document pairs for training dense retrievers and prompt us to consider low-cost alternatives in place of expensive human annotations.

For future research, an open question remains regarding the differences between synthetic and real query-document pairs. It would be interesting to explore how various augmentation methods contribute to dense retrieval and investigate the salient span selection for query extraction in more depth.

Limitations

While we have demonstrated the effectiveness of various data augmentation methods for training unsupervised dense retrieval models, it is important to acknowledge the following limitations of this study:

- Our experiments with QEXT are limited in scope. We have not extensively explored key factors such as scoring models, span lengths, and the number of candidates. Further investigation into these aspects could provide valuable insights.

- The selection of language generation tasks in TQGEN is manually selected, and alternative tasks such as back-translation and paraphrase generation are not explored. Additionally, the heuristic definition of prompts used to generate pseudo queries may not always yield the desired outputs. Exploring other generation models and considering a wider range of output sequences could be interesting avenues for future research.

- Due to computational constraints, we have only explored a limited number of settings for backbones and hyperparameters, and models were trained for up to 200k steps. It remains unclear whether other configurations could result in significant performance changes. Further exploration of different settings and training durations could provide a more comprehensive understanding of the models' capabilities.

Acknowledging these limitations opens up opportunities for future research to address these gaps and gain a deeper understanding of the proposed augmentation methods in the context of unsupervised dense retrieval.

Ethics Statement

Dataset Biases: The AUGQ datasets used in this study were generated from publicly available web-scale data, specifically Wikipedia and CommonCrawl. It is important to note that these datasets predominantly reflect the cultural perspectives and biases of the English-speaking population. Therefore, models trained on these datasets may inherit and potentially propagate biases related to politics, gender, and other social factors. Moreover, TQGEN models (T0) used in our experiments may also carry biases from the data they were trained on.

Environmental Cost: Our experiments heavily relied on A100 GPUs, with typically eight GPUs used per experiment, and each experiment running for up to four days. The backbone models, such as BERT-base and T0, have a significant number of parameters (110 million and 3 billion, respectively). While our study required extensive experiments to obtain the reported results, we believe that future research and applications can benefit from the insights and resources (checkpoints and data) shared by this work, thereby reducing the need to repeat these resource-intensive comparisons. It is crucial to consider the environmental cost of conducting large-scale experiments and explore ways to optimize computational resources in future research endeavors.

References

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Wei-Cheng Chang, X Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2019. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *arXiv preprint arXiv:2110.06918*.

Sukmin Cho, Soyeong Jeong, Wonsuk Yang, and Jong C Park. 2022. Query generation with external knowledge for dense retrieval. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 22–32.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

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.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Revanth Gangi Reddy, Vikas Yadav, Md Arafat Sultan, Martin Franz, Vittorio Castelli, Heng Ji, and Avirup Sil. 2022. Towards robust neural retrieval with source domain synthetic pre-finetuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1065–1070, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993.

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

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Hiroki Iida and Naoaki Okazaki. 2022. Unsupervised domain adaptation for sparse retrieval by filling vocabulary and word frequency gaps. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 752–765.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv e-prints*, pages arXiv–2112.

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.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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.

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. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and S Yu Philip. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021a. 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.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a discriminative text encoder for dense retrieval via contrastive span prediction. *arXiv preprint arXiv:2204.10641*.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021b. Prop: pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 283–291.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021c. B-prop: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1513–1522.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pretraining. *arXiv preprint arXiv:2201.10005*.

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.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint*, 6.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2021. Learning to retrieve passages without supervision. *arXiv preprint arXiv:2112.07708*.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022a. Improving passage retrieval with zero-shot question generation. *arXiv preprint arXiv:2204.07496*.

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

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Xiaoyu Shen, Svitlana Vakulenko, Marco Del Tredici, Gianni Barlacchi, Bill Byrne, and Adrià de Gispert. 2022. Low-resource dense retrieval for open-domain question answering: A comprehensive survey. *arXiv preprint arXiv:2208.03197*.

Zilu Tang, Muhammed Yusuf Kocyigit, and Derry Tanti Wijaya. 2022. Augcse: Contrastive sentence embedding with diverse augmentations. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 375–398.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States. Association for Computational Linguistics.

Jiawen Wu, Xinyu Zhang, Yutao Zhu, Zheng Liu, Zikai Guo, Zhaoye Fei, Ruofei Lai, Yongkang Wu, Zhao Cao, and Zhicheng Dou. 2022. Pre-training for information retrieval: Are hyperlinks fully explored? *arXiv preprint arXiv:2209.06583*.

Yiqing Xie, Xiao Liu, and Chenyan Xiong. 2023. Unsupervised dense retrieval training with web anchors. *arXiv preprint arXiv:2305.05834*.

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. *arXiv preprint arXiv:2007.00808*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569.

Nan Yang, Furu Wei, Binxing Jiao, Daxing Jiang, and Linjun Yang. 2021. xmoco: Cross momentum contrastive learning for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6120–6129.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. Coco-dr: Combating distribution shifts in zero-shot dense retrieval with contrastive and distributionally robust learning. *arXiv preprint arXiv:2210.15212*.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey. *arXiv preprint arXiv:2211.14876*.

Jiawei Zhou, Xiaoguang Li, Lifeng Shang, Lan Luo, Ke Zhan, Enrui Hu, Xinyu Zhang, Hao Jiang, Zhao Cao, Fan Yu, Xin Jiang, Qun Liu, and Lei Chen. 2022. Hyperlink-induced pre-training for passage retrieval in open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7135–7146, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 Additional Implementation Details

We adopt most of the settings used by CONTRIEVER (Izacard et al., 2021) for unsupervised training, with some modifications considering the number of model variants we experiment with.

A.1.1 Pretraining

Specifically, AUGTRIEVER is trained with the following configurations:

1. Most models are trained for 100k steps.
2. The batch size is set to 1,024 for most models, and 2,048 for AUGQ-CC with TQGEN/QGEN/HYBRID.
3. We use a learning rate (*lr*) of $5e^{-5}$ with 10k steps of linear warmup.
4. The optimizer used is ADAM.
5. For the MoCo architecture, we set the queue size to 2^{14} (CONTRIEVER used a queue size of 2^{17}). We empirically observed that a larger queue size can deteriorate the performance.
6. As to hybrid settings: (1) *Hybrid-All*: We use RANDOMCROP for 20% of the spans, QEXT-PLM for 10%, and a combination of DOC-TITLE and TQGEN for the remaining 70%. (2) *Hybrid-TQGen*: We use RANDOMCROP for 20% of the spans and rely solely on TQGEN for the remaining 80%.
7. We experiment with two larger scale training: (1) *Hybrid-TQGen+* is trained trained for 200k steps using a batch size of 2048 and queue size of 2^{14}; (1) *Hybrid-TQGen++* is trained trained for 200k steps using a batch size of 4096 and queue size of 2^{16}.

Note that our training setting is considerably smaller comparing with CONTRIEVER (500k steps, batch size 2048, queue size 2^{17}). All experiments are conducted on cloud instances equipped with eight NVIDIA A100 GPUs (40GB) and most training jobs were finished within 48 hours.

A.1.2 Domain Adapataion

For domain adaptation, we train a pretrained model (Hybrid-TQGen++) for up to 2k steps with a batch size of 256/32 (MSMARCO&Wikipedia/others) and a learning rate of $1e^{-5}$. We train single models for five Wikipedia datasets (NQ, HotpotQA, DBPedia-Entity, FEVER and Climate-FEVER) and two community QA datasets (Quora and CQADupStack), resulting in 10 different models for target domains.

A.1.3 Fine-tuning

For fine-tuning, we train models using MSMARCO[4] for 10k steps with a batch size of 1,024 plus one hard negative example (1 positive + 2,047 negative), using a learning rate of $1e^{-5}$).

For fair comparison, we fine-tune SPAR Λ using the query-encoder only. For all reproduced fine-tuning, we use pooling and vector normalization in consistence with the way in their pretraining.

A.2 Detailed Analysis on BEIR

We plot the relative performance of different models in comparison with CONTRIEVER on BEIR in Figure 4. The left heatmap shows the relative performance on each BEIR dataset, and the right one presents the averaged scores after grouping from different aspects.

AUGTRIEVER models perform significantly better on QA datasets (e.g. TREC-COVID, FiQA, NQ and HotpotQA), worse on fact checking datasets (e.g. FEVER, Climate-FEVER, Scifact and Quora), and similarly on the rest datasets. CONTRIEVER explicitly blends Wikipedia and CCNet in training, to favor knowledge-rich testsets, but it does not show consistent benefits across all five Wikipedia related testsets. It is enlightening to see that, DOC-TITLE and QEXT-PLM with MoCo, which extract certain parts of original documents for pseudo queries, can deliver comparable or better performance to CONTRIEVER. As for our TQGEN models, both models perform very well on TREC-COVID, which contributes to the major part of the improvement. But they underperform DOC-TITLE and QEXT-PLM considerably on Climate-FEVER and Quora, indicating that each augmentation method may be most beneficial for certain tasks. Training with hybrid strategies does not appear to simply bring the advantage of all. With regard to the effect of pseudo query length, TQGEN-TOPIC indeed performs better on datasets with short and medium length queries (SQ/MQ), and TQGEN-EXSUM shows more strength on medium and long queries.

[4] https://huggingface.co/datasets/sentence-transformers/embedding-training-data/blob/main/msmarco-triplets.jsonl.gz

A.3 Examples of QExt

Two examples of random spans ranked by two proposed QExt methods are shown in Table 4. In the first example, we notice that both BM25 and QEXT-PLM are able to rank informative spans to the higher places and place generic spans to the bottom (e.g. “a few years later” and “over the first three”). In the second example, we observe that BM25 is more influenced by the low-frequent words (e.g. HS, HO, NaH), whereas QEXT-PLM is more resistant to the noise, placing more informative spans to the top.

A.4 Examples of TQGen

Six documents from Pile-CC and the corresponding generated pseudo queries are shown in Table 5. We find that most generated outputs are semantically relevant, in spite of a certain degree of hallucination. In most cases, Topic outputs one or two important phrases and Title outputs one short sentence. Both AbSum and ExSum generate relatively longer sentences as summaries, and ExSum does not necessarily use contents of the original texts. In the last example, four outputs by T0 are almost the same, indicating that the model ignores the specified prompts.

A.5 BEIR dataset groups

The groups disussed in Figure 4 are defined as follows:

- SQ (Short Query): 3 datasets of which the average query length is *shorter than 7 words*, including: NFCorpus, Touchè-2020, DBPedia-Entity.
- MQ (Medium Query): 8 datasets of which the average query length is *between 8 to 13 words*, including: TREC-COVID, NQ, FiQA-2018, SCIDOCS, FEVER, SciFact, Quora, CQADupStack.
- LQ (Long Query): 3 datasets of which the average query length is *longer than 17 words*, including: HotpotQA, ArguAna, Climate-FEVER.
- SD (Short Document): 3 datasets of which the average document length is *shorter than 70 words*, including: NFCorpus, HotpotQA, DBPedia-Entity.
- MD (Medium Document): 6 datasets of which the average document length is *between 75 and 150 words*, including: NQ, FiQA-2018, SCIDOCS, FEVER, Climate-FEVER, CQADupStack.
- LD (Long Document): 6 datasets of which the average document length is *longer than 160 words*, including: TREC-COVID, NFCorpus, ArguAna, Touchè-2020, SCIDOCS, SciFact.
- Phrase: 2 datasets whose queries are phrases: DBPedia-Entity, ArguAna, Climate-FEVER.
- Question: 7 datasets whose queries are questions: TREC-COVID, NQ, HotpotQA, FiQA-2018, Touchè-2020, Quora, CQADupStack.
- Sentence: 5 datasets whose queries are sentences: ArguAna, SCIDOCS, FEVER,

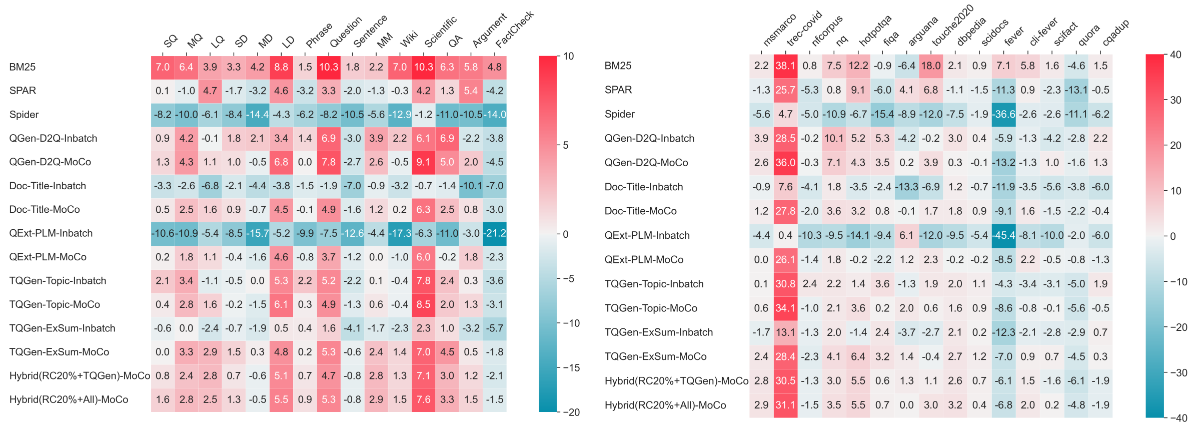

Figure 4: Two heatmaps show the relative performance gain/loss of different models against Contriever. The left heatmap shows nDCG@10 difference on each BEIR dataset, and in the right figure we group BEIR datasets in different ways. *SQ/MQ/LQ*: datasets with short/medium/long queries. *SD/MD/LD*: datasets with short/medium/long documents. *Phrase/Question/Sentence* denotes datasets that use *this* form of queries. And the rest are categorized by text domains as defined in (Thakur et al., 2021). Refer to A.5 for specific grouping of datasets.

Climate-FEVER, SciFact.
- MM: MS MARCO.
- WIKI: 5 Wikipedia-related datasets: NQ, HotpotQA, DBPedia-Entity, FEVER, Climate-FEVER.
- SCIENTIFIC: 4 datasets: TREC-COVID, Touchè-2020, SCIDOCS, SciFact.
- QA: 3 datasets: NQ, HotpotQA, FiQA-2018.
- ARGUMENT: 2 datasets: ArguAna, Touchè-2020.
- FACTCHECK: 3 datasets: SciFact, FEVER, Climate-FEVER.

A.6 Complete Results

Document	**BM25**	**QEXT-PLM** (T5-Small)
ASD can sometimes be diagnosed by age 14 months, although diagnosis becomes increasingly stable over the first three years of life: for example, a one-year-old who meets diagnostic criteria for ASD is less likely than a three-year-old to continue to do so a few years later. In the UK the National Autism Plan for Children recommends at most 30 weeks from first concern to completed diagnosis and assessment, though few cases are handled that quickly in practice. Although the symptoms of autism and ASD begin early in childhood, they are sometimes missed; years later, adults may seek diagnoses to help them or their friends and family understand themselves, to help their employers make adjustments, or in some locations to claim disability living allowances or other benefits.	[14.09] diagnosis and assessment, though few cases are handled	[79.73] adjustments, or in some locations to claim disability
	[14.05] partly because autistic symptoms overlap with those	[85.62] completed diagnosis and assessment, though few cases are
	[13.55] completed diagnosis and assessment, though few cases are	[90.46] some locations to claim disability living allowances or
	[12.06] adjustments, or in some locations to claim disability	[102.39] diagnosis and assessment, though few cases are handled
	[11.05] some locations to claim disability living allowances or	[105.74] partly because autistic symptoms overlap with those
	[10.48] the challenge of obtaining payment can	[112.83] and assessment, though few cases are handled
	[9.78] Conversely, the cost of screening	[115.09] ASD begin early in childhood,
	[9.67] ASD begin early in childhood,	[120.57] family understand themselves, to help their
	[9.42] in some locations to claim disability	[127.31] Conversely, the cost of screening
	[9.38] and assessment, though few cases are handled	[135.09] the challenge of obtaining payment can
	[8.45] overlap with those of common blindness	[139.24] overlap with those of common blindness
	[8.15] family understand themselves, to help their	[139.38] friends and family understand themselves,
	[7.03] facial expressions and eye	[141.17] in some locations to claim disability
	[6.77] friends and family understand themselves,	[147.92] a few years later.
	[5.56] a few years later.	[159.65] facial expressions and eye
	[4.32] over the first three	[180.59] over the first three
Reaction with oxygen Upon reacting with oxygen, alkali metals form oxides, peroxides, superoxides and suboxides. However, the first three are more common. The table below shows the types of compounds formed in reaction with oxygen. The compound in brackets represents the minor product of combustion. The alkali metal peroxides are ionic compounds that are unstable in water. The peroxide anion is weakly bound to the cation, and it is hydrolysed, forming stronger covalent bonds. NaO + 2HO → 2NaOH + HO The other oxygen compounds are also unstable in water. 2KO + 2HO → 2KOH + HO + O LiO + HO → 2LiOH Reaction with sulphur With sulphur, they form sulphides and polysulphides. 2Na + 1/8S → NaS + 1/8S → NaS...NaS Because alkali metal sulphides are essentially salts of a weak acid and a strong base, they form basic solutions. S + HO → HS + HO HS + HO → HS + HO Reaction with nitrogen Lithium is the only metal that combines directly with nitrogen at room temperature.	[26.35] HS + HO HS + HO →	[207.94] 2NaCl Alkali metals in liquid ammonia Alkali metals
	[25.02] NaH + HO → NaOH + H Reaction	[227.13] they form sulphides and polysulphides. 2Na + 1/8S
	[23.74] 2NaCl Alkali metals in liquid ammonia Alkali metals	[256.10] give dilithium acetylide. Na and K can react
	[22.44] + HO → NaOH + H Reaction	[298.67] Reaction with sulphur With sulphur,
	[20.70] Na + xNH → Na +	[304.82] dilithium acetylide. Na and K
	[18.01] + 1/3N → LiN LiN	[309.33] 4NaCl + Ti Reaction with organohalide
	[18.00] (at 150C) Na + NaCH →	[312.97] Because alkali metal sulphides are essentially salts of
	[17.89] + 2HO → 2KOH + HO +	[315.61] peroxides are ionic compounds that are unstable in
	[17.36] high temperatures) NaH + HO →	[316.52] HS + HO HS + HO →
	[17.30] reaction with oxygen. The compound in brackets	[320.61] NaH + HO → NaOH + H Reaction
	[16.25] give dilithium acetylide. Na and K can react	[321.21] + 2HO → 2KOH + HO +
	[15.73] they form sulphides and polysulphides. 2Na + 1/8S	[327.07] give dilithium acetylide. Na
	[15.60] Reaction with sulphur With sulphur,	[336.63] Na + e(NH) Due to the presence
	[15.39] Alkali metals dissolve in liquid ammonia or	[337.68] the case of Rb and Cs. Na
	[15.30] Because alkali metal sulphides are essentially salts of	[347.56] (at 150C) Na + NaCH →
	[14.90] the case of Rb and Cs. Na	[348.45] 1/2H (at 150C) Na + NaCH

Table 4: Examples of random spans ranked by BM25 and a Pretrained Language Model (T5-Small). The numbers in brackets are scores of BM25 or T5-Small (negative log likelihood).

Document	Output
Topic: reinvent midnight madness\n\nAmazon announced a new service at the AWS re:Invent Midnight Madness event. Amazon Sumerian is a solution that aims to make it easier for developers to build virtual reality, augmented reality, and 3D applications. It features a user friendly editor, which can be used to drag and drop 3D objects and characters into scenes.	**Topic:** Sumerian **Title:** aws announces sumerian 3d editor **AbSum:** Developing immersive experiences for virtual and augmented reality (VR) and 3D gaming has become an increasing challenge for tech companies. **ExSum:** a new service called amazon sumerian is launched **QGen:** what is amazon sumerian 3d?
Running\n\nStat\n\nDinner with people is always better than eating alone, especially when the food is good. Good food tastes even better when enjoyed with people. Tonight Amy came over to try my second attempt at the Brussels Sprouts Veggie Soup to which I have made some changes (see recipe below in previous post) for a better result, I believe.\n\nWe were at the store earlier and saw some nice looking haricot verts and heirloom tomatoes, so we decide to assemble a simple salad from those. Of course while I\u2019m at the market, I can\u2019t not get some five peppercorn salami. Our simple dinner of soup, salami, bread, cheese, salad, and wine was on the table in 15 minutes.	**Topic:** Simple dinner **Title:** An easy recipe for a delicious winter soup enjoyed by friends is always a winner. **AbSum:** If you try making a Brussels Sprouts Veggie Soup this weekend, please let me know how it turns out. **ExSum:** If you try out my Brussels Sprouts Veggie Soup recipe it should turn out even better, and it was certainly better tonight. **QGen:** are you better at dinner with someone or alone?
Gordhan asks for fresh thinking\n\nBusiness News / 9 July 2012, 4:48pm\n\nSAPA\n\nCape Town 141010\nFinance Minister, Pravin Gordhan briefing parliment on the annal business report.South African Finance Minister Pravin Gordhan said on Thursday the world was heading towards a \"currency war\" unless developed nations gave ground in negotiations at the Group of 20 (G20).\npicture : neil baynes\n\nGordhan told the 16th World Economic History Congress in Stellenbosch there was an imbalance between the locus of production and that of growth, and between political beliefs and the predominant reality...	**Topic:** Finance Minister Pravin Gordhan said there was an imbalance between the loc **Title:** Finance Minister Pravin Gordhan has warned that the world is heading towards a \"currency war\" unless developed nations give ground in negotiations **AbSum:** Finance Minister Pravin Gordhan has warned that the world is heading towards a \"currency war\" unless developed nations give ground in negotiations at the G20. In a speech at the World Economic History Congress in Stellenbosch on Thursday, Gordhan said that he was encouraged **ExSum:** Pravin Gordhan has criticised South Africa's cabinet. **QGen:** who is the finance minister of south africa
The Difference Between Botox and Dermal Fillers\n\nWritten by CG Cosmetic on February 19, 2015\n\nCG Cosmetic understands the difficulties that come with aging. Everyone has days where they look in the mirror and are concerned with what they see. Whether or not you see deep lines and wrinkles, or fine lines, aging is inevitable. The question then, is what can you do about these signs of aging? Perhaps the most common solution people have heard of is Botox. However, Botox is not the only option. While CG Cosmetic offers Botox procedures, we also offer Dermal Fillers.\n\nBotox\n\nMost men and women have heard about celebrities using Botox to rejuvenate their skin and appear more youthful, but Botox isn\u2019t just for celebrities. Botox has provided amazing and effective results that are also safe and convenient for many individuals. Facial lines and wrinkles often occur because of the way your muscles work underneath your skin...	**Topic:** Botox vs. Dermal Fillers **Title:** Botox is popular for many reasons, including it helping treat aging lines and wrinkles. Dermal fillers, meanwhile, add volume to **AbSum:** Botox relaxes muscles under the skin and helps treat aging lines and wrinkles. Dermal Fillers, meanwhile, add volume to treat sagging skin. CG Cosmetic offers both treatments to help patients maintain a youthful appearance. **ExSum:** Botox relaxes muscles under the skin and helps treat aging lines and wrinkles. Dermal Fillers aims to add volume to facial features such as lips, chin, and jaw line. **QGen:** is botox used for aging?
Cryopreserved (frozen) Donor vials are available to individual Client(s)/Recipient(s), within the United States and Worldwide to achieve Assisted Reproduction.\nBioGenetics Corporation was established in 1980 to become the first commercial sperm bank in the United States located in New Jersey.\n\nBioGenetics respects the current medical standards and ethics set forth by the:\n\nAmerican Society for Reproductive Medicine (ASRM)\nAmerican Society of Andrology (ASA)\nThe American Association of Tissue Banks (AATB)\nThe American Urological Association (AUA)\nand all related scientific associations as well as regulatory government agencies that\nmay impact Reproductive Cell and Tissue Banking...	**Topic:** Cell banking **Title:** BioGenetics is a Commercial Sperm Bank and Tissue Bank **AbSum:** A commercial sperm bank for patients seeking fertility treatment and assisted reproduction **ExSum:** BioGenetics is a Commercial Sperm Bank and Tissue Bank providing services to all Assisted Reproductive Technologies. **QGen:** where is biogenetics located?

Setting	Model	MS MARCO	TREC-COVID	NFCorpus	NQ	HotpotQA	FiQA-2018	ArguAna	Tóuche-2020	DBPedia	SCIDOCS	FEVER	Climate-FEVER	SciFact	Quora	CQADupStack	BEIR Avg (14)	NQ			TriviaQA			WQ			CuratedTREC			SQuAD v1.1			EQ (Macro-Avg)			Average		
																		R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100
Baselines																																						
	BM25	**22.8**	**65.6**	32.5	**32.9**	**60.3**	23.6	31.5	**36.7**	**31.3**	15.8	**75.3**	**21.3**	66.5	78.9	**29.9**	**43.0**	43.8	63.0	78.2	66.3	76.4	83.1	41.8	62.3	75.5	64.6	81.1	90.4	57.5	71.2	82.0	61.0	71.4	80.1	55.8	70.9	81.5
	Contriever †	20.6	27.5	31.7	25.4	48.1	24.5	37.9	16.7	29.2	14.9	**68.2**	15.5	64.9	**83.5**	28.4	36.9	47.2	67.2	81.3	59.5	74.2	83.2	44.0	65.8	79.8	67.7	85.5	93.1	44.2	63.1	78.4	50.3	64.7	76.3	52.1	70.1	82.0
	SPAR †	19.3	53.1	26.5	26.2	57.2	18.5	42.0	23.4	28.1	13.4	56.9	16.4	62.7	70.4	27.9	37.3	43.9	61.8	76.8	64.4	75.5	83.1	40.7	59.9	74.7	62.5	79.1	90.4	52.1	67.1	79.2	57.4	68.2	77.7	53.5	68.6	80.3
	Spider †	15.0	32.2	26.7	14.5	41.3	9.1	29.0	4.7	21.6	13.0	31.6	12.9	62.3	72.4	22.2	28.1	48.3	67.6	80.8	63.4	75.4	83.4	45.5	65.8	80.1	64.7	83.4	93.1	43.9	61.8	77.0	53.0	65.3	76.9	53.1	69.9	81.9
	CPT-text S		52.9	32.0		51.5	34.1	38.7	21.0	27.2		57.1	15.8	65.4	68.1																							
	CPT-text M		44.3	34.5		53.0	37.3	41.2	23.3	29.6		58.2	15.6	68.3	70.3																							
	CPT-text L		42.7	**36.9**		54.3	**39.7**	39.2	22.8	31.2		63.8	16.1	**71.2**	68.7																							
	Condenser																	13.0	25.5	43.4	4.5	9.6	18.5	20.3	35.8	51.9	9.9	20.2	34.4	6.1	13.2	25.3	1.0	2.7	7.6	9.1	17.8	30.2
	CoCondenser																	28.9	46.8	63.5	7.5	13.8	24.3	30.2	50.7	68.7	11.7	22.5	39.3	8.5	16.5	28.8	0.5	1.4	8.7	14.6	25.3	38.9
Pile-CC																																						
RC	Inbatch	16.3	28.7	20.8	15.5	34.4	15.2	43.9	4.6	20.5	9.6	22.6	7.9	56.7	81.4	22.3	27.4	45.4	65.2	78.5	57.5	72.2	81.9	52.0	70.1	81.3	66.4	85.0	92.9	42.0	60.6	75.2	40.7	54.8	69.3	50.7	68.0	79.8
RC	MoCo	19.2	39.5	28.0	22.3	43.0	21.2	38.1	12.6	22.2	13.8	48.1	14.6	63.5	81.8	27.1	34.0	44.7	64.2	78.1	59.2	73.8	82.5	45.3	65.6	80.6	63.4	82.4	93.2	43.0	62.5	77.2	47.6	61.3	73.8	50.5	68.3	80.9
QGen-D2Q	Inbatch	24.4	55.9	31.6	**35.5**	53.2	**29.8**	33.8	16.5	32.1	15.3	62.3	14.2	60.8	80.7	**30.6**	39.5	**51.8**	**72.1**	**84.4**	60.2	74.6	83.5	46.7	68.9	81.6	70.0	87.0	**94.5**	39.2	58.2	73.5	55.7	68.3	77.9	53.9	71.5	82.6
QGen-D2Q+RC	MoCo	23.2	**63.4**	31.5	32.5	52.4	28.0	38.1	**20.5**	29.4	14.8	55.1	14.2	65.9	81.9	29.7	39.8	51.7	70.6	82.8	63.6	76.3	83.8	**52.9**	**71.5**	**82.4**	**70.8**	**87.9**	94.0	43.8	63.3	77.9	55.3	67.9	78.3	**56.3**	**72.9**	83.2
Hybrid-All	MoCo	23.3	58.0	30.5	28.4	53.5	25.1	39.2	17.8	31.8	15.6	62.1	17.0	63.3	77.4	26.5	39.4	46.0	66.4	80.7	63.0	75.6	83.7	48.7	69.4	82.2	68.0	85.2	93.2	43.1	62.4	77.7	53.5	66.5	77.3	53.7	70.9	82.5
Hybrid-TQGen	MoCo	23.5	58.5	30.3	28.9	53.6	25.2	38.0	19.6	32.3	15.3	61.4	17.5	65.2	78.7	26.5	39.0	45.6	66.1	80.4	62.8	75.7	83.6	49.0	69.7	81.9	67.0	85.7	92.9	42.5	62.3	77.3	53.6	66.5	77.8	53.4	71.0	82.3
Hybrid-TQGen+	MoCo	**24.6**	59.5	32.7	32.7	57.1	26.9	41.2	17.8	**32.6**	**16.8**	**65.5**	16.4	67.8	79.3	29.4	41.1	49.5	69.7	82.7	**64.1**	**76.6**	**84.3**	51.1	70.7	82.3	69.9	86.9	93.4	43.9	63.4	78.0	**56.6**	**68.5**	**78.7**	55.8	72.6	**83.3**
Hybrid-TQGen++	MoCo	**25.4**	**64.0**	**35.1**	33.5	**57.6**	29.2	39.8	19.8	**32.7**	**16.7**	65.3	15.9	**68.7**	79.9	**30.6**	**42.1**	50.2	70.0	83.1	**65.4**	**77.2**	**84.5**	50.5	70.6	82.3	**72.8**	86.9	93.8	44.7	**64.2**	78.4	**58.2**	**69.9**	**79.2**	**57.0**	**73.1**	**83.6**
Doc-Title	Inbatch	19.7	35.1	27.6	27.2	44.5	22.1	24.6	9.8	30.3	14.2	56.4	12.0	59.4	79.7	22.4	33.2	46.7	66.4	80.8	58.7	72.8	82.4	47.3	68.9	81.3	67.0	86.0	93.1	38.6	57.3	73.1	47.9	62.1	74.4	51.0	68.9	80.8
Doc-Title+RC	MoCo	21.8	55.3	29.7	28.9	51.2	25.3	37.9	18.4	31.0	15.8	59.1	17.1	63.5	81.3	28.0	38.7	46.9	67.5	80.1	62.3	75.9	83.7	50.7	70.5	81.9	70.5	85.2	93.5	**44.9**	**64.0**	77.9	51.0	64.5	76.0	54.4	71.3	82.2
QExt-BM25	Inbatch	16.3	27.2	21.6	17.1	34.3	16.6	40.5	5.3	20.1	10.3	26.0	8.1	56.4	81.4	24.2	27.8	46.5	66.1	79.5	58.1	72.5	82.1	51.1	70.0	81.6	68.4	85.6	93.2	41.8	60.7	75.8	42.1	56.0	70.3	51.3	68.5	80.4
QExt-BM25+RC	MoCo	20.2	46.8	28.4	24.3	47.5	22.6	37.2	13.3	25.2	14.6	53.0	16.1	63.0	**82.7**	27.0	35.8	44.5	65.0	80.3	60.3	74.1	82.8	44.4	68.0	80.9	65.0	85.2	92.8	43.8	63.5	77.7	50.6	63.9	75.4	51.4	69.9	81.6
QExt-PLM	Inbatch	16.2	27.9	21.4	15.9	33.9	15.1	**44.0**	4.7	19.7	9.5	22.8	7.4	54.9	81.5	22.4	27.2	46.4	65.6	79.6	58.1	72.7	82.1	51.2	69.9	81.2	67.0	85.6	93.7	42.2	60.9	75.7	40.9	54.9	68.9	51.0	68.3	80.2
QExt-PLM+RC	MoCo	20.6	53.6	30.3	27.2	47.9	22.3	39.1	19.0	29.0	14.7	59.7	17.7	64.5	**82.7**	27.1	38.2	45.2	65.2	79.9	61.3	74.8	83.3	45.7	68.1	81.4	65.6	84.0	93.1	44.7	63.7	**78.7**	50.9	64.4	76.3	52.2	70.0	82.1
TQGen-Topic	Inbatch	20.7	58.2	**34.1**	27.6	49.4	28.1	36.7	18.6	31.2	16.0	63.9	12.1	61.8	78.5	30.3	39.0	44.2	65.0	80.3	56.9	72.3	82.3	41.0	65.2	80.6	63.5	83.7	92.7	37.1	57.1	73.8	49.4	64.0	75.7	48.7	67.9	80.9
TQGen-Topic+RC	MoCo	21.2	61.6	30.7	27.4	51.7	24.7	40.0	17.3	30.7	15.8	59.6	14.7	64.8	77.9	27.9	38.9	45.6	65.7	80.9	62.0	75.4	83.7	45.7	67.3	81.1	67.2	84.9	93.5	42.2	62.3	77.6	50.7	64.5	76.4	52.2	70.0	82.2
TQGen-Title	Inbatch	20.0	46.5	32.1	28.9	49.6	27.0	32.9	14.3	30.9	16.4	64.2	14.1	62.7	80.1	29.6	37.8	44.7	65.7	80.5	57.5	73.2	82.7	42.2	66.7	80.9	64.7	84.3	92.8	36.2	56.0	72.5	49.8	64.0	75.7	49.2	68.3	80.8
TQGen-Title+RC	MoCo	21.8	60.0	31.0	29.0	51.8	26.3	39.3	17.0	30.0	15.8	60.7	16.1	65.0	79.4	28.7	39.3	46.8	66.6	80.4	62.7	75.8	83.9	48.2	69.0	82.0	66.0	85.3	92.9	42.4	62.2	77.1	52.8	66.2	77.2	53.1	70.8	82.3
TQGen-AbSum	Inbatch	18.1	39.8	29.5	26.7	45.5	25.7	33.5	13.6	30.0	15.6	54.1	12.9	60.0	79.6	28.2	35.3	43.5	64.5	79.7	55.9	72.4	82.1	45.1	67.1	80.3	64.1	84.0	92.8	35.3	54.8	71.8	45.7	60.0	72.9	48.3	67.1	79.9
TQGen-AbSum+RC	MoCo	23.3	52.7	29.9	29.8	54.0	26.8	38.7	20.2	31.1	16.6	63.6	**17.8**	65.6	78.1	29.3	39.6	48.3	67.1	81.0	62.3	75.6	83.9	48.7	69.7	81.4	68.3	85.3	92.8	43.6	62.3	77.0	55.0	67.4	77.8	54.4	71.2	82.3
TQGen-ExSum	Inbatch	18.9	40.5	30.4	27.4	46.7	26.9	34.3	13.9	31.3	15.1	55.9	13.4	62.2	80.6	29.1	36.3	46.0	66.5	81.0	56.5	73.1	82.4	45.4	67.5	79.7	64.8	84.2	92.1	36.4	55.8	72.4	47.3	61.5	73.8	49.4	68.1	80.2
TQGen-ExSum+RC	MoCo	23.0	55.8	29.4	29.5	54.4	27.6	39.4	16.3	31.8	16.1	61.2	16.4	65.6	79.0	28.7	39.4	48.8	67.9	80.8	63.2	76.1	83.8	49.4	69.8	81.5	67.2	85.3	92.5	43.6	63.0	77.7	54.2	66.8	77.5	54.4	71.5	82.3
Wikipedia																																						
RC	Inbatch	14.0	23.8	13.7	20.2	32.0	9.4	43.2	2.9	19.2	8.1	23.3	10.8	52.4	76.8	18.1	25.3	40.7	60.3	75.7	50.9	66.9	79.1	30.2	50.7	68.7	48.3	69.6	86.9	29.4	49.6	67.9	26.4	40.3	58.4	37.6	56.2	72.8
RC	MoCo	17.5	35.5	22.8	25.1	42.1	13.8	38.5	6.9	22.0	12.1	40.5	15.7	60.9	76.7	20.8	30.9	43.8	62.9	77.5	55.8	70.7	80.8	30.2	52.2	70.9	48.6	72.6	88.6	34.0	53.4	71.4	36.6	51.7	67.5	41.5	60.6	76.1
QGen-D2Q	Inbatch	**25.4**	58.3	26.1	39.8	55.3	**23.6**	28.9	17.2	31.0	11.7	67.9	**21.7**	59.2	77.7	20.4	38.5	58.2	75.5	86.1	65.0	76.7	83.6	56.7	**73.8**	82.7	75.7	88.6	93.8	44.3	62.5	76.8	**61.7**	**71.7**	**79.8**	60.3	74.8	83.8
QGen-D2Q+RC	Inbatch	23.5	53.1	22.7	39.6	54.5	22.5	41.0	14.7	27.3	10.9	44.0	13.1	60.0	62.9	22.8	34.9	57.8	74.9	85.5	65.6	77.0	84.1	56.0	73.5	82.3	75.9	88.0	94.0	46.7	63.9	77.5	60.3	70.7	79.4	60.4	74.7	83.8
QGen-D2Q	MoCo	16.3	42.5	19.7	19.8	38.0	12.3	32.5	8.3	25.1	9.7	26.6	10.9	46.4	75.3	17.7	27.5	39.4	59.0	74.9	57.6	71.3	80.8	48.9	67.5	79.9	70.2	86.2	93.1	38.6	57.3	73.9	45.5	59.7	73.0	50.0	66.8	79.3
QGen-D2Q+RC	MoCo	23.7	**64.9**	26.2	35.9	55.3	22.1	36.8	21.1	28.1	12.8	50.3	17.9	62.3	79.9	25.4	38.5	53.8	71.8	83.4	66.0	77.4	84.1	54.7	71.6	82.5	76.1	88.8	94.5	47.1	65.8	79.4	56.4	69.0	78.9	59.0	74.0	83.8
QGen-PAQ	Inbatch	22.3	49.3	19.3	**42.6**	**55.9**	16.4	2.0	12.8	24.1	10.3	42.2	10.5	55.0	74.0	20.1	31.0	**65.3**	**79.7**	**87.4**	**72.5**	**80.9**	**86.0**	**57.6**	**73.8**	**83.3**	**82.1**	**90.6**	95.7	55.2	71.2	82.4	59.8	70.5	79.4	**65.4**	**77.8**	**85.7**
QGen-PAQ+RC	Inbatch	19.1	25.9	25.0	26.0	46.8	15.2	39.5	12.6	26.7	11.1	35.0	10.8	52.9	81.1	19.9	30.6	51.9	71.3	83.6	66.0	77.3	84.2	48.5	68.9	81.5	73.9	87.5	94.7	46.3	63.7	78.2	50.7	64.5	76.7	56.2	72.2	83.1
QGen-PAQ	MoCo	21.4	37.2	24.9	36.6	52.3	16.4	10.4	16.5	28.8	10.2	45.5	8.8	59.7	74.9	23.7	31.8	63.4	78.6	87.1	70.2	80.3	85.9	54.6	72.2	82.6	79.1	89.9	95.1	**55.4**	**71.9**	82.9	52.7	65.9	77.4	62.6	76.5	85.2
QGen-PAQ+RC	MoCo	22.6	28.4	23.1	26.7	49.8	14.7	32.0	10.7	26.6	9.7	31.4	5.7	53.9	**82.3**	23.7	29.9	56.9	74.2	85.2	70.0	79.7	85.4	54.3	71.4	81.9	77.0	89.3	**96.1**	55.1	71.8	**83.0**	52.6	66.0	77.7	61.0	75.4	84.9
Doc-Title	Inbatch	14.7	37.7	27.5	22.6	41.6	13.6	24.7	17.2	28.8	10.9	60.9	13.4	45.4	61.6	14.1	30.0	35.9	54.6	71.2	49.6	64.7	76.9	32.2	55.3	72.8	51.2	73.2	87.6	25.5	42.7	61.9	49.2	61.6	73.0	40.6	58.7	73.9
Doc-Title+RC	Inbatch	16.0	25.3	27.1	25.4	43.4	12.9	36.0	14.8	27.1	10.8	46.7	11.7	56.7	65.2	18.0	30.1	21.3	45.6	54.8	63.1	77.3	32.0	55.2	63.2	69.8	80.7	12.9	32.7	44.1	53.0	71.4	23.3	51.3	62.3	71.0	87.3	12.4
Doc-Title	MoCo	12.0	42.3	23.0	14.5	34.7	7.1	31.9	8.0	22.9	10.1	19.7	4.6	47.7	72.4	14.3	25.2	32.3	51.6	68.9	48.3	63.9	76.3	34.3	53.3	72.2	51.7	71.3	85.7	26.1	44.3	63.1	39.6	53.2	67.0	38.7	56.3	72.2
Doc-Title+RC	MoCo	18.5	40.4	26.8	27.8	47.4	15.8	38.0	10.7	28.0	12.9	49.8	13.8	60.3	76.7	22.9	33.7	45.6	66.5	79.9	59.5	72.9	82.3	36.1	57.0	74.0	57.1	77.8	89.5	36.9	56.2	72.9	47.2	60.6	73.0	47.1	65.2	78.6
Doc-Anchor	Inbatch	15.6	36.3	27.1	22.3	44.6	12.5	26.3	13.7	26.9	11.6	42.7	10.1	50.0	74.5	18.3	29.8	36.4	56.7	74.3	55.7	70.6	80.8	32.7	55.6	74.4	51.6	76.5	89.3	28.9	47.9	67.6	35.9	51.8	68.2	40.2	59.9	75.8
Doc-Anchor+RC	Inbatch	16.5	32.3	27.7	22.4	38.9	14.1	34.1	9.5	25.7	12.6	21.2	11.0	60.7	76.7	20.8	29.1	41.7	61.9	77.6	57.3	71.3	81.9	31.9	51.9	72.6	52.0	75.4	88.3	31.6	50.6	69.7	31.4	46.1	63.1	41.0	59.5	75.5
Doc-Anchor	MoCo	16.7	49.5	25.6	23.5	49.6	14.1	38.0	14.6	29.1	12.8	40.5	13.6	54.2	76.8	20.9	33.1	39.4	58.7	75.9	59.5	73.1	82.1	36.2	57.2	76.0	60.1	79.1	91.6	37.5	55.8	72.7	44.1	58.4	72.2	46.1	63.7	78.4
Doc-Anchor+RC	MoCo	17.9	51.8	26.8	27.8	48.1	16.4	36.7	18.3	27.3	13.0	50.9	17.0	60.3	77.8	23.3	35.4	45.1	64.7	79.1	59.1	72.7	82.6	37.9	58.2	76.2	56.2	78.4	90.8	36.0	55.4	72.6	45.6	59.3	72.1	46.7	64.8	78.9
QExt-Self	Inbatch	11.4	30.2	16.4	19.1	39.0	11.8	8.8	2.3	12.6	9.5	37.2	14.1	59.6	71.2	19.9	25.1	39.3	56.8	72.7	48.5	64.2	77.2	28.7	47.6	66.5	46.8	68.2	83.4	31.0	48.7	66.3	29.1	42.9	59.7	37.2	54.7	71.0
QExt-Self+RC	Inbatch	14.6	25.9	15.6	22.1	36.0	11.3	36.7	3.5	20.8	9.0	32.5	13.2	54.7	77.2	20.2	27.1	41.5	60.3	75.8	51.5	67.1	79.4	29.8	50.6	68.1	47.6	71.2	86.2	31.5	50.7	69.6	26.7	40.3	58.8	38.1	56.7	73.0
QExt-Self	MoCo	9.9	29.1	16.1	14.0	28.1	9.1	2.2	4.4	18.7	8.0	18.7	10.8	38.4	69.3	14.9	20.1	34.7	51.4	67.6	48.7	64.1	76.7	34.6	54.7	71.7	54.3	75.2	88.2	32.0	50.0	67.6	29.9	43.0	58.5	39.0	56.4	71.7
QExt-Self+RC	MoCo	12.2	27.8	14.3	17.0	31.2	8.7	29.7	4.5	19.4	8.5	20.1	10.8	40.4	68.6	14.1	22.5	37.7	55.3	70.0	51.9	67.5	78.7	34.6	56.4	73.0	55.2	76.4	88.0	33.8	52.1	69.4	32.7	45.0	60.0	41.0	58.8	73.2
QExt-BM25	Inbatch	11.4	30.2	16.4	19.1	39.0	11.8	8.8	2.3	12.6	9.5	37.2	14.1	59.6	71.2	19.9	25.1	39.2	58.0	73.5	49.9	66.2	78.4	27.8	47.7	65.9	44.7	67.0	83.7	28.1	46.6	65.7	30.6	45.0	62.1	36.7	55.1	71.5
QExt-BM25+RC	Inbatch	15.3	28.2	18.0	22.3	35.3	12.8	27.8	3.4	18.1	9.7	27.6	12.4	58.0	76.9	21.4	26.6	42.7	61.6	77.2	52.5	67.8	79.9	31.4	52.2	70.0	50.3	71.2	87.0	31.0	50.9	69.6	29.3	43.9	61.6	39.5	57.9	74.2
QExt-BM25	MoCo	15.8	19.5	21.5	18.3	36.6	9.7	16.2	3.5	19.6	10.1	21.3	7.1	52.6	66.0	21.8	23.1	40.2	58.6	74.2	57.4	70.9	81.0	39.6	59.6	75.0	57.9	77.5	90.1	41.2	59.1	74.5	38.0	52.2	67.4	45.7	63.0	77.0
QExt-BM25+RC	MoCo	17.1	29.2	25.3	24.3	40.9	15.0	35.6	9.0	17.5	12.6	26.9	16.7	58.5	75.9	22.6	29.3	45.8	63.7	78.2	59.2	73.0	82.2	34.0	57.4	74.2	57.4	75.9	89.8	38.5	57.5	74.4	42.8	56.7	70.4	46.3	64.0	78.2
QExt-PLM	Inbatch	15.0	28.4	15.1	26.8	31.2	14.1	3.9	8.9	16.9	10.6	45.8	12.4	57.0	76.4	21.2	26.3	39.8	59.1	74.7	48.9	65.7	78.3	31.1	50.9	69.4	46.8	70.5	86.3	30.0	49.1	68.9	24.4	38.5	56.8	36.8	55.6	72.4
QExt-PLM+RC	Inbatch	15.1	28.0	21.8	25.0	34.8	12.8	11.1	8.1	21.6	10.4	39.7	13.8	57.1	77.1	21.8	27.4	40.9	59.9	75.3	49.4	66.1	79.1	29.5	49.9	69.2	47.0	70.9	86.6	29.6	49.7	69.1	23.1	37.6	56.9	36.6	55.7	72.7
QExt-PLM	MoCo	16.7	26.0	24.1	23.1	34.5	12.8	**44.5**	11.9	22.3	11.5	26.7	10.6	57.4	78.6	21.3	29.0	43.6	62.5	77.5	58.8	72.6	81.8	37.1	59.2	76.7	59.4	79.7	91.6	38.2	57.6	74.5	42.9	57.3	71.6	46.6	64.8	79.0
QExt-PLM+RC	MoCo	18.6	44.0	26.5	28.6	44.8	15.5	32.3	15.0	24.6	12.1	53.0	20.8	60.9	78.6	23.0	34.3	44.2	63.2	78.7	57.7	72.2	82.1	34.2	55.2	74.8	53.3	75.9	90.1	35.5	55.4	73.3	41.5	56.1	70.3	44.4	63.0	78.2
TQGen-Topic	Inbatch	21.3	62.4	**30.3**	29.7	54.3	21.2	32.8	21.4	**31.6**	13.5	67.7	15.9	60.9	76.9	25.8	**38.9**	44.2	64.6	79.1	61.8	74.7	82.9	45.6	66.8	79.9	67.0	83.3	92.4	41.9	60.2	75.5	57.0	68.5	78.3	52.9	69.7	81.3
TQGen-Topic+RC	Inbatch	21.5	56.2	30.0	31.2	52.2	22.9	42.0	17.1	30.7	13.9	56.3	14.2	63.6	78.5	26.5	38.2	48.3	67.0	80.3	62.6	75.4	83.4	45.1	66.5	80.4	66.1	82.7	92.1	43.7	62.3	76.9	53.1	66.3	77.0	53.2	70.0	81.7
TQGen-Topic	MoCo	15.4	51.1	24.8	16.0	39.4	13.8	32.9	9.9	22.6	11.9	30.3	12.2	49.8	76.0	17.1	29.1	34.6	54.6	72.3	54.7	69.8	80.4	42.6	61.4	77.4	56.1	78.8	89.9	35.3	54.3	71.9	43.6	58.5	72.3	44.5	62.9	77.4
TQGen-Topic+RC	MoCo	21.3	62.1	27.6	29.9	53.9	20.8	39.2	18.9	29.8	13.8	55.7	15.1	64.5	80.2	25.5	38.3	46.4	66.5	80.5	63.8	76.0	83.6	47.3	67.7	80.8	68.6	84.2	93.2	44.5	63.6	78.6	53.8	66.9	77.9	54.1	70.8	82.4
TQGen-Title	Inbatch	20.8	58.6	30.1	31.0	52.0	22.6	32.0	20.4	30.4	14.2	**69.7**	16.5	61.6	78.2	25.8	38.8	46.5	67.2	80.7	61.5	75.4	83.5	46.7	68.4	80.8	66.7	85.2	92.1	40.7	59.7	75.3	57.4	69.6	79.2	53.2	70.9	81.9
TQGen-Title+RC	Inbatch	20.9	49.3	28.8	30.7	45.1	22.7	39.7	13.7	28.2	**14.4**	58.6	15.0	64.2	77.3	25.8	36.7	48.5	68.0	81.0	63.0	75.8	83.6	45.9	68.3	80.9	68.3	85.5	91.9	42.6	61.6	76.6	53.7	67.1	77.7	53.6	71.0	82.0
TQGen-Title	MoCo	18.5	61.1	24.7	21.4	46.2	14.8	27.8	13.4	23.3	12.4	35.3	12.8	54.6	74.4	19.8	31.6	38.5	58.8	74.4	58.3	72.1	82.3	44.4	65.6	80.6	61.7	81.3	91.5	38.6	57.7	74.4	49.6	63.3	75.6	48.5	66.5	79.8
TQGen-Title+RC	MoCo	21.6	64.5	27.8	29.9	52.1	20.3	37.7	**23.7**	28.7	13.6	56.9	17.8	63.4	77.7	25.1	38.5	46.1	66.5	80.0	63.1	76.2	83.9	47.7	68.0	81.9	68.2	85.5	92.8	43.4	63.2	77.9	53.6	66.6	77.4	53.7	71.0	82.3
TQGen-AbSum	Inbatch	17.4	51.2	27.7	27.2	46.8	20.8	35.3	15.5	26.5	14.1	64.0	16.1	60.5	77.8	25.3	36.3	47.2	67.1	81.1	63.4	76.4	83.9	48.8	69.6	81.6	68.9	86.3	93.2	42.5	60.9	76.6	57.6	69.7	79.5	54.7	71.7	82.6
TQGen-AbSum+RC	Inbatch	17.0	40.1	26.2	25.6	41.7	18.7	40.7	13.1	24.5	13.7	53.5	15.2	60.6	77.3	25.0	34.0	48.0	66.9	80.9	63.2	76.0	83.8	49.3	68.8	81.5	72.1	86.6	92.9	44.4	62.7	77.3	53.3	66.7	77.6	55.0	71.3	82.3
TQGen-AbSum	MoCo	21.2	53.3	25.3	25.9	45.6	17.3	41.2	15.7	24.7	13.6	40.5	15.7	59.8	71.0	21.2	33.6	46.1	65.9	79.4	63.8	75.9	83.9	50.5	69.6	81.5	70.9	85.7	92.9	44.3	63.3	78.3	52.5	66.3	78.1	54.7	71.1	82.4
TQGen-AbSum+RC	MoCo	21.2	61.3	25.5	31.0	52.2	19.7	38.2	20.1	27.4	13.4	56.0	19.1	61.5	73.0	22.4	37.2	47.5	67.2	80.7	64.5	76.5	84.0	49.5	68.5	81.0	69.5	85.7	93.1	44.2	63.3	78.1	54.7	67.6	78.2	55.0	71.5	82.5
TQGen-ExSum	Inbatch	18.2	46.4	27.9	30.7	46.7	20.9	34.3	13.8	27.3	14.0	67.8	16.4	61.8	79.4	**26.6**	36.7	50.6	69.8	81.9	63.6	76.5	83.9	48.8	68.9	81.6	69.2	87.3	92.8	43.4	61.9	77.3	57.0	68.9	78.6	55.4	72.2	82.7
TQGen-ExSum+RC	Inbatch	17.6	38.0	26.5	26.1	39.1	20.1	40.0	13.0	25.0	13.3	53.9	13.7	60.9	79.2	25.6	33.9	50.7	68.6	81.1	63.5	76.0	84.0	48.9	68.5	80.6	71.0	87.2	92.7	44.8	63.5	77.7	49.8	63.5	75.5	54.8	71.2	81.9
TQGen-ExSum	MoCo	22.8	60.8	26.7	31.1	51.9	18.7	36.5	16.2	28.5	13.6	47.9	15.8	**64.5**	73.6	23.2	36.4	48.9	67.5	81.0	65.8	77.1	84.3	51.6	69.8	81.8	70.5	86.5	92.8	47.2	65.2	79.4	53.9	67.1	78.4	56.3	72.2	82.9
TQGen-ExSum+RC	MoCo	22.5	57.9	27.0	32.6	54.9	19.4	39.9	17.6	30.4	13.4	57.6	17.5	62.7	75.0	24.3	37.9	49.5	67.3	81.2	65.9	77.1	84.3	51.1	70.1	82.1	72.2	85.9	92.5	46.2	64.9	79.3	56.1	68.5	78.8	56.8	72.3	83.0

Table 6: Scores for all of the experiments in this paper. For MS MARCO and BEIR datasets, we report the nDCG@10. For six open-domain QA datasets, we report the Recall@5/20/100. We highlight the best and second best in each column, and **best** in each group per column.

	MM	BEIR Avg (14)	trec-covid	nfcorpus	nq	hotpotqa	fiqa	arguana	webis-touche2020	dbpedia-entity	scidocs	fever	climate-fever	scifact	quora	cqadupstack
BM25	22.8	43.0	65.6	32.5	32.9	60.3	23.6	31.5	36.7	31.3	15.8	75.3	21.3	66.5	78.9	29.9
DPR	35.4	36.8	57.5	21.0	39.8	37.1	27.4	41.4	20.8	23.6	10.7	58.9	17.6	47.5	84.2	28.1
ANCE	38.8	40.5	65.4	23.7	44.6	45.6	29.5	41.5	24.0	28.1	12.2	66.9	19.8	50.7	85.2	29.6
ColBERT	40.1	44.4	67.7	30.5	52.4	59.3	31.7	23.3	20.2	39.2	14.5	77.1	18.4	67.1	85.4	35.0
SPAR†	38.0	41.5	14.8	16.5	15.0	35.5	9.4	15.0	5.2	21.7	7.7	14.4	1.5	29.3	64.2	17.7
Spider†	24.8	19.1	53.3	31.1	46.1	50.5	29.3	47.5	16.6	34.3	16.2	56.7	18.8	51.2	85.3	29.2
LaPraDor†	38.9	40.4	59.9	28.6	43.0	59.4	25.5	32.4	24.1	34.7	14.2	71.6	19.3	60.8	80.0	27.5
Condenser (Book&Wiki)†	38.7	40.9	56.9	28.6	46.3	50.0	25.4	43.0	17.1	35.0	13.8	68.6	22.7	54.1	84.8	26.1
CoCondenser (MSMARCO)†	40.8	42.9	66.3	31.8	46.4	50.9	29.8	43.6	20.4	35.9	14.2	66.2	23.1	56.7	85.6	29.1
Contriever†	41.3	45.2	65.4	33.1	52.0	64.6	32.8	26.1	18.9	42.0	15.9	74.8	21.9	67.6	85.7	32.6
QGen-D2Q	39.6	43.9	68.3	31.6	49.8	58.2	29.7	27.0	19.8	38.7	14.6	70.0	23.6	65.9	85.1	31.7
RandomCrop	38.4	42.4	67.9	31.1	47.3	57.1	29.4	23.5	16.0	38.0	13.8	70.2	20.7	63.8	84.5	30.4
QExt-PLM	38.8	42.5	67.7	31.8	47.2	57.8	28.0	22.2	17.9	38.4	14.4	69.7	20.6	63.9	84.6	31.1
TQGen-Topic	38.8	43.2	70.0	31.0	48.1	58.1	28.5	24.4	19.7	38.0	14.6	71.2	21.5	63.5	84.9	30.7
TQGen-Title	38.8	43.2	70.0	31.0	48.1	58.1	28.5	24.4	19.7	38.0	14.6	71.2	21.5	63.5	84.9	30.7
TQGen-AbSum	39.1	43.9	71.7	31.8	50.0	59.6	30.1	22.1	18.3	39.4	15.1	73.5	21.9	65.4	84.8	31.1
TQGen-ExSum	38.8	43.1	72.3	31.2	47.2	57.6	29.7	25.7	18.0	39.0	14.2	69.8	19.0	64.0	84.9	30.7
Hybrid-All	38.9	43.6	68.3	31.6	48.2	58.9	28.4	24.9	20.3	39.1	14.7	72.8	22.1	65.0	85.1	30.4
Hybrid-TQGen+	40.3	44.7	69.2	32.9	50.8	61.3	30.9	26.2	19.7	39.8	15.8	72.1	22.1	66.5	85.8	32.5
Hybrid-TQGen++	40.9	45.8	74.0	35.2	51.4	63.3	31.9	24.9	20.1	40.7	15.6	72.4	23.5	68.7	86.2	33.3

Table 7: Detailed fine-tuning scores.