Large Dual Encoders Are Generalizable Retrievers

Anonymous ACL submission

Abstract

It has been shown that dual encoders trained on one domain often fail to generalize to other domains for retrieval tasks. One widespread belief is that the bottleneck layer of a dual encoder, where the final score is simply a dot-product between a query vector and a passage vector, is too limited to make dual encoders an effective retrieval model for out-of-domain generalization. In this paper, we challenge this belief by scaling up the size of the dual encoder model *while keeping the bottleneck embedding size fixed.* With multi-stage training, surprisingly, scaling up the model size brings significant improvement on a variety of retrieval tasks, especially for out-of-domain generalization. Experimental results show that our dual encoders, **G**eneralizable **T**5-based dense **R**etrievers (GTR), outperform existing sparse and dense retrievers on the BEIR dataset (Thakur et al., 2021) significantly. Most surprisingly, our ablation study finds that GTR is very data efficient, as it only needs 10% of MS Marco supervised data to achieve the best out-of-domain performance.[1]

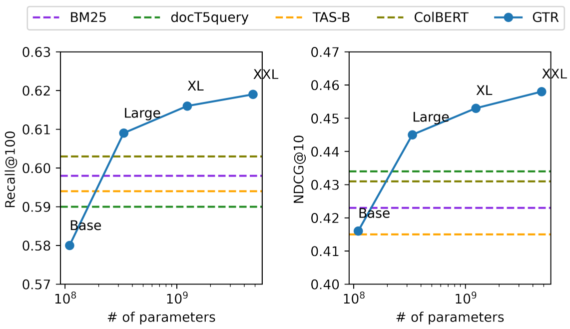

Figure 1: Average Recall@100 and NDCG@100 on all BEIR tasks (excl. MS Marco). Scaling up consistently improves dual encoders' out-of-domain performance.

1 Introduction

Typical neural retrieval models follow a dual encoder paradigm (Gillick et al., 2018; Yang et al., 2020; Karpukhin et al., 2020). In this setup, queries and documents are encoded separately into a shared fixed-dimensional embedding space where relevant queries and documents are represented in each other's proximity. Then, approximated nearest neighbor search (Vanderkam et al., 2013; Johnson et al., 2021) is applied to efficiently retrieve relevant documents given an encoded input query.

While dual encoders are popular neural retrievers, the expressiveness of the model is limited by a bottleneck layer consisting of only a simple dot-product between query embeddings and passage embeddings. Lu et al. (2021); Khattab and Zaharia (2020) have discussed that the dot-product (or cosine similarity) between the embeddings might not be powerful enough to capture semantic relevance. Thakur et al. (2021) studied whether the retriever models can generalize to other domains and conclude that dual encoder models have "issues for out-of-distribution data", and showed that models with more interactions between queries and documents have better generalization ability.

In this paper, we challenge this belief by scaling up the dual encoder model size while keeping the bottleneck embedding size fixed. Note that scaling up a dual encoder is different from scaling up pretrained language models such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) because of the presence of the bottleneck layer. While increasing the model size can greatly increase model capacity, for dual encoders with a fixed embedding size, the interactions between queries and documents are still limited by a simple dot-product.

To test this hypothesis, we take advantage of the existing T5 architecture and checkpoints, which allows us to build encoders of up to 5 billion parameters while keeping the bottleneck embedding dimension of 768 in all configurations, as illustrated in fig. 2. Following Ni et al. (2021), we build dual encoders by taking the encoder part of T5. For

[1] We will release code and models upon publication.

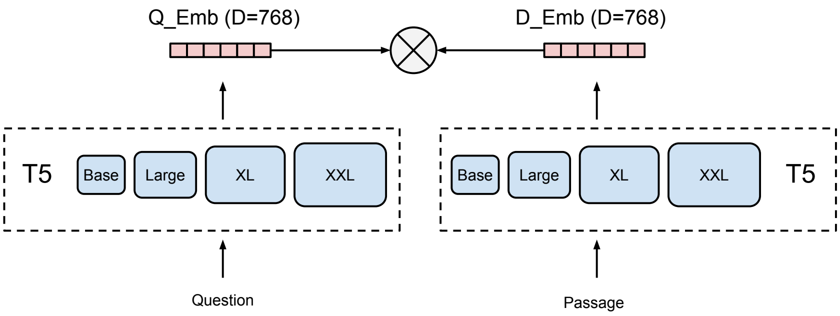

Figure 2: Architecture of **G**eneralizable **T**5-based dense **R**etrievers. The research question we ask is: *can scaling up dual encoder model size improve the retrieval performance while keeping the bottleneck layers **fixed?*** Only the encoder is taken from the pre-trained T5 models, and the two towers of the dual encoder share parameters.

effectively using the power of large models, we collect roughly two billion web question-answer pairs as generic pre-training data. By combining pre-training using generic training data and fine-tuning using MS Marco (Nguyen et al., 2016), we are able to train large-scale dual encoder retrieval models. We call the resulting models **G**eneralizable **T**5-based dense **R**etrievers (**GTR**).

We evaluate the zero-shot performance of GTR on the BEIR benchmark (Thakur et al., 2021), which consists of 18 information retrieval tasks across 9 domains. Scaling up leads to better generalization despite the fixed bottleneck embedding dimension. Second, pre-training on community question-answer pairs and fine-tuning on human curated data are both important to fully utilize the power of the scaled up model. In addition, with scaling and pre-training, we found GTR to be highly data efficient in terms of human annotated queries, as it only needs to use 10% of MS Marco to match the overall out-of-domain performance.

2 Background

2.1 Dual Encoder and dense retrieval

Classic retrieval models such as BM25 (Robertson and Zaragoza, 2009) relies on lexical overlap: term frequency, inverse document frequency and document length. To allow semantic matching between queries and documents, dense retrieval models such as dual encoders (Yih et al., 2011; Gillick et al., 2019; Karpukhin et al., 2020) are introduced, where both queries and documents are embedded into low-dimensional dense representations.

One critical challenge for dual encoder models is that the performance is possibly bounded by the dot-product similarity function. As such, there is growing interest in applying lightweight interaction layers to replace the single dot-product function. Luan et al. (2020) proposes a multi-vector encoding model, which represents each document as a fixed-size set of multiple vectors, and calculate the relevance scores as the maximum inner product over this set. ColBERT (Khattab and Zaharia, 2020) proposes to learn embeddings for each token and then use a "MaxSim" operation to select the best candidate. While these models can achieve significant improvement, dual encoder is still the most popular one in practice due to its simpleness and ability to scale. In this paper, we take a step back and show that performance of single dot-product based methods can be improved significantly.

2.2 BEIR generalization task

For evaluation in this paper we use BEIR, a heterogenous benchmark for zero-shot evaluation of information retrieval models. The BEIR zero-short evaluation suit contains 18 information retrieval datasets[2] across 9 domains, including *Bio-Medical*, *Finance*, *News*, *Twitter*, *Wikipedia*, *StackExchange*, *Quora*, *Scientific*, and *Misc*. The majority of the datasets have binary relevancy labels indicating whether a document is relevant to a given query or not. A small part of the datasets have 3-level or 5-level relevancy judgements. We refer readers BEIR (Thakur et al., 2021) for more details.

3 Generalizable T5 Retriever

3.1 T5 dual encoder

We use the dual encoder framework to train dense retrieval models and follow prior work (Xiong et al., 2020; Hofstätter et al., 2021) to initialize dual encoders from pre-trained language models. In this

[2]MS Marco is excluded from the zero-shot comparison as many baseline model used it as training data.

work, we found convenient to use the pre-trained T5 model family as our backbone encoder because the T5 model family provides off-the-shelf pre-trained models (e.g. T5, mT5, byT5) with a wide range of model capacity from millions to billions of parameters (Raffel et al., 2020; Xue et al., 2020, 2021). The architectures of our models are illustrated in fig. 2.

Let paired examples $\mathcal{D} = \{(q_i, p_i^+)\}$ be the training set, where q_i is an input question and p_i^+ is a related passage (e.g., a semantically relevant passage to the question). Following Ni et al. (2021), we encode the question q_i and passage p_i^+ into embeddings by feeding them to the T5 encoder and taking the mean pooling of the encoder as output. In all outr experiments, we fix the output embeddings to be of size 768.

We train the model using an in-batch sampled softmax loss (Henderson et al., 2017):

$$\mathcal{L} = \frac{e^{\mathrm{sim}(q_i, p_i^+)/\tau}}{\sum_{j \in \mathcal{B}} e^{\mathrm{sim}(q_i, p_j^+)/\tau}}, \tag{1}$$

where the similarity scoring function *sim* is the cosine similarity between the embeddings of q_i and p_i^+. $\mathcal{B}$ is a mini-batch of examples and τ is the softmax temperature.

Additional negatives p_j^- can be given for input question q. The loss is computed by including them in the denominator:

$$\mathcal{L} = \frac{e^{\mathrm{sim}(q_i, p_i^+)/\tau}}{\sum_{j \in \mathcal{B}} e^{\mathrm{sim}(q_i, p_j^+)/\tau} + e^{\mathrm{sim}(q_i, p_j^-)/\tau}}. \tag{2}$$

We also apply a bi-directional in-batch sampled softmax loss (Yang et al., 2019), where we compute losses for both question to document matching and document to question matching.

3.2 Multi-stage training

As shown in fig. 3, we use a multi-stage dual encoder training approach to achieve generalizable retrieval models.

The training process includes a pre-training stage on a web-mined corpus and a fine-tuning stage on search datasets. The web-mined corpus provides a large amount of semi-structured data pairs (such as question-answer pairs and conversations), which can provide rich semantic relevance information. It is easy to collect but it is often not well annotated, if at all. The search datasets are often annotated by humans, and the queries and documents are also authored by humans. These datasets are of high quality but costly to collect.

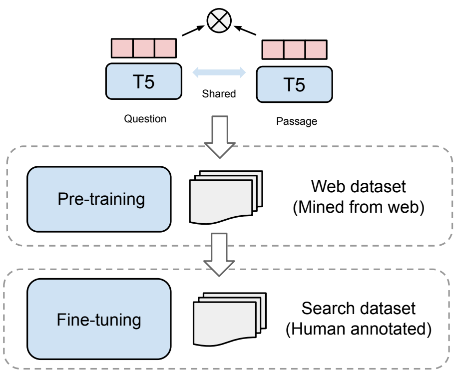

Figure 3: Multi-stage training for GTR models.

In this work, for dual encoder pre-training, we initialize the dual encoders from the T5 models and train on question-answer pairs collected from the Web. Recently, Sentence-T5 (Ni et al., 2021) explored different ways to extract strong text embeddings and achieved remarkable performance on SentEval and Sentence Textual Similarity tasks. We follow that setting to encode queries and passages via mean pooling from the T5 encoders and focus on the dense retrieval tasks.

For fine-tuning, our aim is to adapt the model to retrieval using a high quality search corpus so the model can learn to better match generic queries to documents. In this paper, we consider two datasets for fine-tuning: MS Marco (Nguyen et al., 2016) and Natural Questions (Kwiatkowski et al., 2019).

4 Experimental setup

4.1 Training Data

Community QA. To leverage the power of large scale models, we collect input-response pairs and question-answer pairs from online forums and QA websites such as Reddit and StackOverflow. This results in 2 billion question-answer pairs that we use to pre-train the dual encoder.

MS Marco. The MS Marco dataset (Nguyen et al., 2016) includes 532K query and document pairs, as search data for fine-tuning. The dataset is sampled from Bing search logs, which covers a broad range of domains and concepts.

Natural Questions. In the fine-tuning stage, we also consider the Natural Questions dataset (Kwiatkowski et al., 2019) , which has been widely used in related work (Karpukhin et al., 2020; Xiong

GTR Models	Base	Large	XL	XXL
# of params	110M	335M	1.24B	4.8B

Table 1: Number of parameters in the GTR models.

et al., 2020). This dataset consists of 130k query and passage pairs which are also human-annotated.

4.2 Configurations

We implement GTR models in JAX[3] and train them on Cloud TPU-V3. We consider different sizes of the T5 transformer (Vaswani et al., 2017) architecture including Base, Large, XL and XXL. Their number of parameters are listed in table 1. Note that we only use the encoder portion of the T5 models and thus the number of parameters are less than half of the full model size. We use the off-the-shelf checkpoints as the initial parameters and use the same sentencepiece vocabulary model.[4]

During pre-training and fine-tuning, we set the batch size to 2048 and use a softmax temperature τ of 0.01. We use Adafactor optimizer (Shazeer and Stern, 2018) and set the initial learning rate to 1e-3 with a linear decay. We train the model for 800K steps and 20K steps for the pre-training and fine-tuning stages, respectively.

For fine-tuning, we use the hard negatives released by RocketQA (Qu et al., 2021) when fine-tuning with MS Marco data and the hard negatives release by (Lu et al., 2021) for Natural Questions, which were proven to lead to better retriever performance. By default, we use the complete MS Marco dataset and the NQ dataset for fine-tuning.

When evaluating on the BEIR benchmark, we use sequences of 64 tokens for the questions and 512 for the documents in all datasets except Trec-News, Robust-04 and ArguAna. In particular, we set the document length to 768 for Trec-News and Robust-04 while setting the question length to 512 for ArguAna, in accordance to the average query and document lengths in these datasets.

4.3 Models for comparison

We consider various baselines, including sparse retrieval models: BM25, DocT5Query, and dense retrieval models: DPR, ANCE, TAS-B, and GenQ (Thakur et al., 2021). In the following sections, we only report the NDCG@10 metric due to the space limitation. The result on Recall@100 are consistent and included in the Appendix.

[3]https://github.com/google/jax

[4]https://github.com/google-research/text-to-text-transfer-transformer

Models	Dim. size
ColBERT	128
DPR, ANCE, TAS-B, GenQ, GTR	768
BM25, DocT5Query	-

Table 2: Dimension of different models. Most dual encoder models set the embedding dimension to 768.

We conduct experiments on four different sizes of our GTR models (GTR-Base, GTR-Large, GTR-XL, and GTR-XXL). We also consider three different settings for GTR to investigate the scaling up effect for different training stages:

- GTR. This is the full GTR model that conducts both pre-training and fine-tuning.
- GTR-FT. This is a fine-tune only version of GTR, where the T5 dual encoders are fine-tuned on the MS Marco dataset.
- GTR-PT. This is a pre-training only version of GTR, where the T5 dual encoders are only pre-trained on the CommunityQA dataset.

We evaluate our models on BEIR (Thakur et al., 2021) as discussed in section 2.2. We consider two retrieval metrics: NDCG@10 and Recall@100 following BEIR. Due to space limitations, we report the Recall@100 results in appendix A.

5 Evaluation Results

We present three groups of experiments to study the a) in-domain performance on MS Marco, b) out-of-domain generalization performance on BEIR, and c) data efficiency.

5.1 Results on MS Marco

We first analyze in-domain performance based on the evaluation results on MS Marco. As show in table 3, with scaling up, the models achieve consistent improvement on NDCG@10. We observe similar improvements on other evaluation metrics including MRR@10 and Recall@1000 and reported the numbers in table 7 of appendix A. This shows that increasing model capacity leads to better in-domain performance.

5.2 Results on BEIR generalization tasks

The next set of experiments investigates the effect of increasing model capacity on out-of-domain (OOD) performance.

NDCG@10 / Model	Lexical / Sparse		Dense					Ours			
	BM25	docT5query	DPR	ANCE	TAS-B	GenQ	ColBERT	GTR-Base	GTR-Large	GTR-XL	GTR-XXL
MS Marco	0.228	0.338	0.177	0.388	0.408	0.408	0.401	0.420	0.430	0.439	**0.442**
Trec-Covid	0.656	**0.713**	0.332	0.654	0.481	0.619	0.677	0.539	0.557	0.584	0.501
BioASQ	0.465	0.431	0.127	0.306	0.383	0.398	**0.474**	0.271	0.320	0.317	0.324
NFCorpus	0.325	0.328	0.189	0.237	0.319	0.319	0.305	0.308	0.329	**0.343**	0.342
NQ	0.329	0.399	0.474	0.446	0.463	0.358	0.524	0.495	0.547	0.559	**0.568**
HotpotQA	**0.603**	0.58	0.391	0.456	0.584	0.534	0.593	0.535	0.579	0.591	0.599
FiQA-2018	0.236	0.291	0.112	0.295	0.300	0.308	0.317	0.349	0.424	0.444	**0.467**
Signal-1M	**0.330**	0.307	0.155	0.249	0.289	0.281	0.274	0.261	0.265	0.268	0.273
Trec-News	0.398	**0.42**	0.161	0.382	0.377	0.396	0.393	0.337	0.343	0.350	0.346
Robust04	0.408	0.437	0.252	0.392	0.427	0.362	0.391	0.437	0.470	0.479	**0.506**
ArguAna	0.315	0.349	0.175	0.415	0.429	0.493	0.233	0.511	0.525	0.531	**0.540**
Touché-2020	**0.367**	0.347	0.131	0.240	0.162	0.182	0.202	0.205	0.219	0.230	0.256
Quora	0.789	0.802	0.248	0.852	0.835	0.830	0.854	0.881	0.890	0.890	**0.892**
DBPedia-entity	0.313	0.331	0.263	0.281	0.384	0.328	0.392	0.347	0.391	0.396	**0.408**
SCIDOCS	0.158	**0.162**	0.077	0.122	0.149	0.143	0.145	0.149	0.158	0.159	0.161
Fever	0.753	0.714	0.562	0.669	0.700	0.669	**0.771**	0.660	0.712	0.717	0.740
Climate-Fever	0.213	0.201	0.148	0.198	0.228	0.175	0.184	0.241	0.262	**0.270**	0.267
SciFact	0.665	**0.675**	0.318	0.507	0.643	0.644	0.671	0.600	0.639	0.635	0.662
CQADupStack	0.299	0.325	0.153	0.296	0.314	0.347	0.350	0.357	0.384	0.388	**0.399**
Avg	0.413	0.429	0.234	0.389	0.414	0.410	0.429	0.416	0.444	0.452	**0.457**
Avg w/o MS Marco	0.423	0.434	0.237	0.389	0.415	0.410	0.431	0.416	0.445	0.453	**0.458**

Table 3: NDCG@10 on the BEIR benchmark. The best result on a given dataset is marked in bold GTR models are pre-trained on CommunityQA dataset and the complete MS Marco dataset. GTR models achieve better NDCG when increasing size from Base to XXL, outperforming the previous best sparse model DocT5Query and dense retrieval model TAS-B.

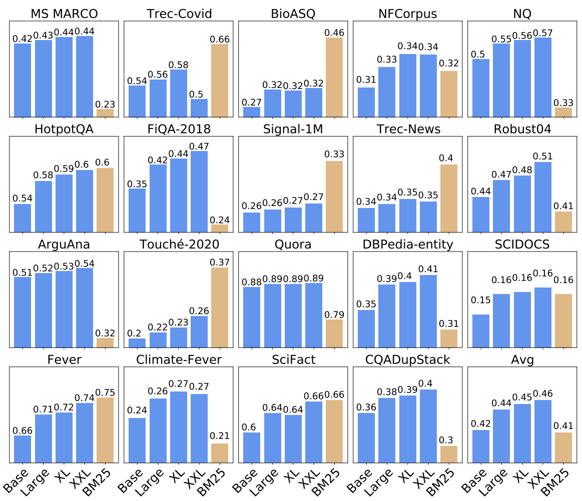

Figure 4: Comparison with BM25 on NDCG@10. The GTR-Base model outperforms BM25 on 9 datasets and the larger GTR models continue to improve on these 9 tasks. The GTR-XXL model catches up or surpasses BM25 on the other 5 datasets and only under-performs on 5 of the remaining tasks.

As shown in table 3, we observe a clear gain on out-of-domain performance in terms of NDCG@10 when the model size increases. The GTR-Large model already outperforms the previous best dense retrieval model TAS-B as well as the best sparse model DocT5Query. Scaling up to GTR-XXL leads to another jump in retrieval performance. Similar improvements are found on Recall@100 as shown in the Appendix's table 8. On average, the scaling up process demonstrates an encouraging ascending trend that eventually outperforms all baseline methods on all evaluation metrics. This confirms that scaling up is a valid path towards generalizability.

Previously, dual encoders failed to match the performance of BM25 for tasks that require better lexical matching capabilities. Thus, we wanted to investigate what kind of tasks can get improved by scaling up the model size. Figure 4 presents a detailed comparison of all sizes of GTR models against the BM25 baseline.

For tasks like NQ where dual encoders have been previously shown to be more effective than BM25, increasing the model size continues to advance the performance of dual encoders. This suggests scaling up can further boost the head start of dense models over sparse models on these datasets.

For tasks like BioASQ and NFCorpus, where dual encoders previously struggled to match the performance of BM25 for inherent reasons, we discovered that scaling up consistently improves the retrieval performance. In particular, for NFCorpus, our Base model under-performs BM25 but the XL model outperforms BM25 by 5.5% (0.343 vs. 0.325). This exciting finding verifies our assumption that scaling up can further exploit the powerful semantic matching capabilities of the dual encoder models and enable them to ultimately outperform BM25.

	GTR-FT		GTR		
Ratio of data	Large	XL	Large	XL	XXL
NDCG@10 on MS Marco					
10%	0.402	0.397	0.428	0.426	0.379
100%	0.415	0.418	0.430	0.439	0.442
Zero-shot average NDCG@10 w/o MS Marco					
10%	**0.413**	0.418	**0.452**	**0.462**	**0.465**
100%	0.412	**0.433**	0.445	0.453	0.458

Table 4: Comparisons of NDCG@10 for GTR models trained with different amount of fine-tuning data. With only 10% of the MS Marco data, both GTR-FT and GTR models (large to XXL) achieve worse in-domain performance; meanwhile they obtain comparable or even superior out-of-domain performance than using the complete MS Marco data.

	GTR-FT	GTR-PT	GTR
Fine-tuning	✓	✗	✓
NDCG@10 on MS Marco			
Base	0.400	0.258	0.420
Large	0.415	0.262	0.430
XL	0.418	0.259	0.439
XXL	0.422	0.252	0.442
Zero-shot average NDCG@10 w/o MS Marco			
Base	0.387	0.295	0.416
Large	0.412	0.315	0.445
XL	**0.433**	0.315	0.453
XXL	0.430	**0.332**	**0.458**

Table 5: Comparisons (NDCG@10) of the models trained with and without pre-training and fine-tuning. Notably, the GTR-FT XL model already achieves an average zero-shot NDCG@10 of 0.433, which outperforms the previous best dual encoder model TAS-B (NDCG@10=0.415).

5.3 Data efficiency for large retrievers

To better understand the data efficiency for large dual encoders, we trained models using different portions of the MS Marco dataset during fine-tuning. In particular, we sampled a subset of the training data by keeping only 10% of the training queries as well as their relevant (positive) passages and irrelevant (hard negative) passages.

As shown in table 4, using 10% of training data reduces the in-domain performance of the GTR models on MS Marco. For the GTR-FT (fine-tuning only) models, using 10% of the data leads to a mixed result of out-of-domain performance.

On the other hand, for full GTR models, using 10% of the MS Marco dataset is sufficient for fine-tuning. In particular, the GTR-Large, XL and XXL models achieve comparable or even better OOD performance than fine-tuning on the complete MS Marco dataset. This might suggest that GTR models have the benefit of data efficiency and could use less training data for domain adaptation.

6 Ablation Study and Analysis

In this section we present ablations and analysis to further understand the effects of scaling up, the impact of fine-tuning and pre-training, and the GTR model's behavior.

6.1 Effect of scaling up for different training stages

The first ablation study aims to investigate how scaling up effects dual encoder pre-training and fine-tuning. Results are listed in table 5.

For fine-tuning only models, scaling up benefits both in-domain and out-of-domain performance. For pre-training only models, the improvement on in-domain performance is not obvious; meanwhile for out-of-domain tasks, scaling up also improves the generalization. Finally with both pre-training and fine-tuning, GTR models consistently improve over GTR-FT models of all sizes. This shows the power of combining scaling up and a generic pre-training stage.

6.2 Importance of the fine-tuning dataset

In table 5, we compare GTR and GTR-PT on the BEIR benchmark to understand the importance of fine-tuning on MS Marco. The table shows that there is a clear gap between GTR models before and after fine-tuning. The result shows the necessity of leveraging a high quality dataset (e.g. search data) to fine-tune the dual encoders.

In table 6, we compare fine-tuning GTR on NQ instead of MS Marco. Compared to MS Marco, NQ only covers Wikipedia documents and is much smaller in size, which allows us to investigate the performance of GTR when fine-tuned on a less generalizable dataset. In addition, fine-tuning on NQ can give us a fair comparison with DPR (Karpukhin et al., 2020).

As shown in table 6, the GTR-base model fine-tuned on NQ outperforms the original DPR model, which uses a BERT-Base model as the encoder backbone. This demonstrates the effectiveness of our pre-training on the Web dataset as well as the hard negatives introduced from Lu et al. (2021) for NQ. Fine-tuning on NQ leads to inferior per-

Model	Fine-tuning dataset	Zero-shot average NDCG@10
DPR	NQ	0.237
GTR-Base	NQ	0.360
GTR-Large	NQ	0.379
GTR-XL	NQ	**0.407**
GTR-Large	MS Marco	0.445
GTR-XL	MS Marco	<u>0.453</u>

Table 6: Comparisons of GTR models fine-tuned on MS Marco and NQ. We report the zero-shot average NDCG@10. Scaling up improves model performance both on NQ and MS Marco.

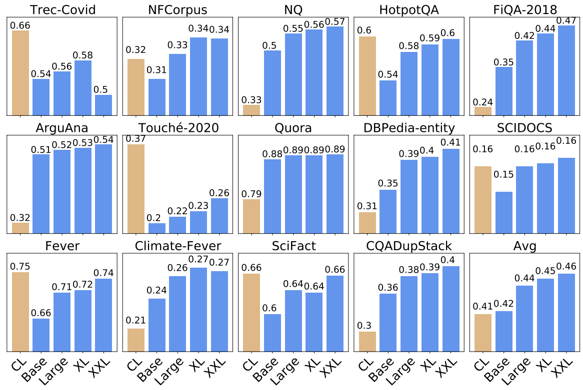

Figure 5: Comparison with Izacard et al. (2021) on NDCG@10. "CL" denotes Izacard et al. (2021) with contrastive learning on C4 and Wiki while others denote our GTR models with different sizes. Note that they only report results on 15 datasets of the BEIR benchmark.

formance compared to fine-tuning on MS Marco, which is consistent with prior work (Thakur et al., 2021). However, importantly, scaling up GTR size improves zero-shot performance on BEIR when fine-tuning on NQ. This shows that the benefit of scaling up holds for different fine-tuning datasets. Furthermore, when scaling from Large to XL, we observe a larger gain when fine-tuning with NQ than with MS Marco, indicating that scaling up helps more when using weaker fine-tuning data.

6.3 Comparison of different dual encoder pre-training strategies

In a concurrent work (Izacard et al., 2021), researchers proposed to conduct contrastive learning (CL) pre-training data from C4 and Wiki dataset in an unsupervised way. In particular, their pre-training data is constructed by randomly choosing two spans from a single document and conduct word deletion or replacement to each span. In contrast, GTR uses Web-mined QA data as the pretraining data.

We compare the performance of our GTR models to their models to gain insights into different pre-training data for dual encoders. As shown in fig. 5, on over half of the datasets, models with our pre-training approach under-perform CL-Pretrain with the base size; while as the model size increases, GTR-Large and -XXL models show significant gains over CL-Pretrain. The best GTR-XXL model achieves 0.49 for NDCG@10 on average while CL-Pretrain achieves 0.46. This demonstrates that scaling up can mitigate the disadvantage of the potentially inferior pre-training data. Note that our pre-training is additive to CL-Pretrain and we can leverage the pre-training on C4 and Wiki to further improve the results. We leave this exploration as future work.

6.4 Document length vs model capacity

Previously, BEIR has shown that models trained with cosine similarity prefer short documents while those trained with dot-product prefer long documents (Thakur et al., 2021). We investigate whether scaling up affect this observation. Specifically, we compute the median lengths (in words) of the top-10 retrieved documents for all queries. Results are shown in fig. 6.

Though all GTR models are trained using cosine similarity, we found that scaling up the model size has influence over the lengths of retrieved documents. We observe an increasing trend of document length for DB-Pedia, Fever, HotpotQA, Signal-1M, Trec-News, and Web-Touche2020 with scaling up. In particular, for Web-Touche2020, the lengths of the retrieved documents grow drastically as the models scale up: The largest GTR-XXL retrieves documents that are on average twice as long compared with the smallest GTR-Base. This plays in our favor since Thakur et al. (2021) show that the majority of relevant documents in Web-Touche2020 are longer.

On the other hand, the only exception we observe is the Trec-Covid dataset, where GTR-XXL model retrieves much shorter documents than those retrieved by the smaller size counterparts. This may explain the inferior performance of GTR-XXL on Trec-Covid shown in table 3 and table 8. We leave it as future work to explore the effects of using the dot-product as similarity function for large dual encoders.

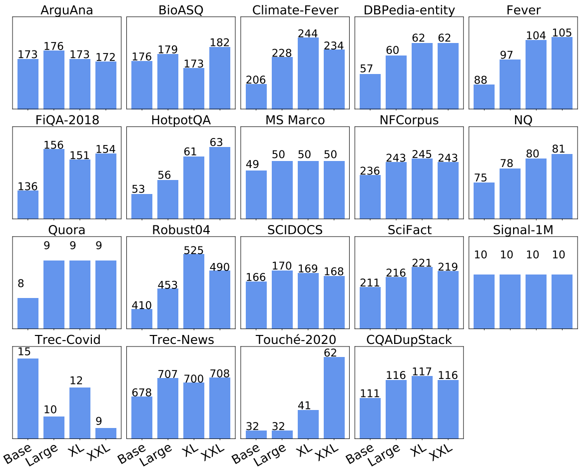

Figure 6: Median lengths (in words) of top-10 retrieved documents for all queries.

7 Related Work

Neural information retrieval. Document retrieval is an important task in the NLP and information retrieval (IR) communities. Traditionally, lexical based approaches trying to match the query and document based on term overlap, such as TF-IDF and BM25 (Robertson and Zaragoza, 2009), have achieved great success in this task. Recently, neural based approaches, which go beyond the simple term matching, are being quickly adopted by the community and achieve state-of-the-art performance on multiple retrieval tasks, such as passage retrieval (Karpukhin et al., 2020), question answering (Ahmad et al., 2019), conversational question answering (Qu et al., 2020) and bitext retrieval (Feng et al., 2020).

Dual encoders for neural retrieval. Dual encoders have demonstrated to be one type of neural retrievers that can achieve great performance compared to traditional sparse models such as BM25 for a wide range of retrieval tasks (Karpukhin et al., 2020; Gillick et al., 2018). One key aspect to their success is the adoption of pre-trained language models, which enables the dual encoders to have backbone contextual embeddings to initialize from. Other techniques such as negative mining (Xiong et al., 2020; Lu et al., 2021; Sachan et al., 2021) and large training batch sizes (Qu et al., 2021) have also shown great effectiveness. However, few of the previous works have discussed the effect of the backbone model's capacity.

Zero-shot neural retrieval. Recent works have shown great improvement under the zero-shot setting for dual encoders by leveraging distillation and synthetic data generation (Thakur et al., 2021; Hofstätter et al., 2021; Ma et al., 2020). Both of these techniques, and scaling up backbone models, are effective ways to close the gap between dual encoders and the upper bound of the single-product approaches with fixed-dimension embeddings. On the other hand, multi-vector approaches introduce more interactions between dense embeddings, which could also benefit from scaling up the backbone multi-vector encoders. We hope that our observation about scaling up model sizes for single dot-product based methods can be combined with these techniques and further push the frontier of neural retrieval models.

8 Inference latency

One caveat for scaling up model size is the increment in the latency overhead. We investigate the inference speed in terms of microseconds (ms) for all GTR models with batch size 1 and input length 128. We found the latency increases from 17 ms, 34 ms, 96 ms to 349 ms. The GTR-Base model has close latency compared to TAS-B while the largest GTR-XXL model has a similar latency to the re-ranking models (Thakur et al., 2021). With the recent work towards making large models efficient with sparsification, distillation and prompt-tuning, we hope the inference time for large dual encoders can be significantly reduced in the future.

9 Conclusion

This paper presents the Generalizable T5 Retriever (GTR), a scaled-up dual encoder model with a fixed-size bottleneck layer. We show that scaling up the model size brings significant improvement on retrieval performance across the board on the BEIR zero-shot retrieval benchmark, especially for out-of-domain generalization. The GTR-XXL model achieves state-of-the-art performance on BEIR, outperforming many models that use earlier interactions between queries and documents. This sheds light on the research direction to keep improving the single vector representation model through better backbone encoders. The findings here are also complementary with other recent works that improve the dual encoder training, including distilling from a ranker / scorer model, using a better contrasting pre-training objective and scaling up the encoders for multi-vector retrieval models.

References

Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. ReQA: An evaluation for end-to-end answer retrieval models. In *Workshop on Machine Reading for Question Answering*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.

D. Gillick, A. Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *ArXiv*, abs/1811.08008.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Matthew Henderson, Rami Al-Rfou, B. Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and R. Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. *arXiv preprint arXiv:2104.06967*.

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 preprint arXiv:2112.09118*.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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*, 7:453–466.

Jing Lu, Gustavo Hernández Ábrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2021. Multi-stage training with improved negative contrast for neural passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6091–6103.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural retrieval via domain-targeted synthetic query generation. *arXiv e-prints*, pages arXiv–2004.

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.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21/140.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. 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 6648–6662, Online. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

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)*.

Dan Vanderkam, Rob Schonberger, Henry Rowley, and Sanjiv Kumar. 2013. Nearest neighbor search in google correlate. Technical report, Google.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv preprint arXiv:2105.13626*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Yinfei Yang, Gustavo Hernández Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Improving multilingual sentence embedding using bidirectional dual encoder with additive margin softmax. *arXiv preprint arXiv:1902.08564*.

Yinfei Yang, Daniel Matthew Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, G. Ábrego, Steve Yuan, C. Tar, Yun-Hsuan Sung, B. Strope, and R. Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *ACL*.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256, Portland, Oregon, USA. Association for Computational Linguistics.

A More results

A.1 Comparisons on MS Marco

Table 7 shows the comparisons of GTR models and the baselines. Note that the best RocketQA model used additional augmented data other than MS Marco to improve the model performance while all others do not. Our best GTR-XXL models outperforms RocketQA on both MRR and recall.

Model	NDCG@10	MRR@10	Recall@1000
ANCE	0.388	0.330	0.959
TAS-Balanced	0.408	0.340	0.975
ColBERT	0.401	0.360	0.968
RocketQA	/	0.370	0.979
GTR-Base	0.420	0.366	0.983
GTR-Large	0.430	0.379	**0.991**
GTR-XL	0.439	0.385	0.989
GTR-XXL	**0.442**	**0.388**	0.990

Table 7: Comparisons of different models on MS Marco. Scaling up can improve GTR models' in-domain performance.

A.2 Recall on BEIR

Table 8 presents the Recall@100 of GTR models and the baselines. Similar to NDCG@10, we observe that scaling up dual encoders lead to significant gains on the BEIR benchmark in terms of recall.

Recall@10 / Model	Lexical / Sparse		Dense					Ours			
	BM25	docT5query	DPR	ANCE	TAS-B	GenQ	ColBERT	GTR-Base	GTR-Large	GTR-XL	GTR-XXL
MS Marco	0.658	0.819	0.552	0.852	0.884	0.884	0.865	0.898	0.908	0.911	**0.916**
Trec-Covid	0.498	**0.541**	0.212	0.457	0.387	0.456	0.464	0.411	0.434	0.457	0.407
BioASQ	**0.714**	0.646	0.256	0.463	0.579	0.627	0.645	0.441	0.490	0.483	0.483
NFCorpus	0.250	0.253	0.208	0.232	0.280	0.280	0.254	0.275	0.298	**0.318**	0.300
NQ	0.760	0.832	0.880	0.836	0.903	0.862	0.912	0.893	0.930	0.936	**0.946**
HotpotQA	0.740	0.709	0.591	0.578	0.728	0.673	0.748	0.676	0.725	0.739	**0.752**
FiQA-2018	0.539	0.598	0.342	0.581	0.593	0.618	0.603	0.670	0.742	0.755	**0.780**
Signal-1M	**0.370**	0.351	0.162	0.239	0.304	0.281	0.283	0.263	0.261	0.268	0.268
Trec-News	0.422	0.439	0.215	0.398	0.418	0.412	0.367	0.475	0.525	0.512	**0.544**
Robust04	**0.375**	0.357	0.211	0.274	0.331	0.298	0.31	0.324	0.365	0.364	0.372
ArguAna	0.942	0.972	0.751	0.937	0.942	0.978	0.914	0.974	0.978	0.980	**0.983**
Touché-2020	0.538	**0.557**	0.301	0.458	0.431	0.451	0.439	0.281	0.282	0.297	0.301
Quora	0.973	0.982	0.470	0.987	0.986	0.988	0.989	0.996	0.996	**0.997**	**0.997**
DBPedia-entity	0.398	0.365	0.349	0.319	**0.499**	0.431	0.461	0.418	0.480	0.480	0.494
SCIDOCS	0.356	0.360	0.219	0.269	0.335	0.332	0.344	0.340	0.358	0.358	**0.366**
Fever	0.931	0.916	0.840	0.900	0.937	0.928	0.934	0.923	0.941	0.944	**0.947**
Climate-Fever	0.436	0.427	0.390	0.445	0.534	0.450	0.444	0.522	0.552	**0.569**	0.556
SciFact	0.908	**0.914**	0.727	0.816	0.891	0.893	0.878	0.872	0.899	0.911	0.900
CQADupStack	0.606	0.638	0.403	0.579	0.622	0.654	0.624	0.681	0.714	0.729	**0.740**
Avg	0.601	0.615	0.425	0.559	0.610	0.605	0.604	0.596	0.625	0.632	**0.634**
Avg w/o MS Marco	0.598	0.603	0.418	0.543	0.594	0.590	0.590	0.580	0.609	0.616	**0.619**

Table 8: Recall@100 on the BEIR benchmark. The best result on a given dataset is marked in bold.