No Parameter Left Behind: How Distillation and Model Size Affect Zero-Shot Retrieval

Guilherme Moraes Rosa,¹,²,³ Luiz Bonifacio,¹,² Vitor Jeronymo,¹,² Hugo Abonizio,¹,² Marzieh Fadaee,³ Roberto Lotufo,¹,² and Rodrigo Nogueira¹,²,³

¹NeuralMind, Brazil
²UNICAMP, Brazil
³Zeta Alpha, Netherlands

ABSTRACT

Recent work has shown that small distilled language models are strong competitors to models that are orders of magnitude larger and slower in a wide range of information retrieval tasks. This has made distilled and dense models, due to latency constraints, the go-to choice for deployment in real-world retrieval applications. In this work, we question this practice by showing that the number of parameters and early query-document interaction play a significant role in the generalization ability of retrieval models. Our experiments show that increasing model size results in marginal gains on in-domain test sets, but much larger gains in new domains never seen during fine-tuning. Furthermore, we show that rerankers largely outperform dense ones of similar size in several tasks. Our largest reranker reaches the state of the art in 12 of the 18 datasets of the Benchmark-IR (BEIR) and surpasses the previous state of the art by 3 average points. Finally, we confirm that in-domain effectiveness is not a good indicator of zero-shot effectiveness. Code is available at https://github.com/guilhermemr04/scaling-zero-shot-retrieval.git

KEYWORDS

Distillation, Ranking, Dense retrieval, Information Retrieval, Zero-shot Learning

1 INTRODUCTION

Language models have shown remarkable effectiveness in text ranking tasks [6, 7, 19, 23, 26], whose objective is to retrieve an ordered list of texts from a corpus according to their relevance to a given query [16]. This success suggests great potential for application in commercial search engines [8]. However, system latency is the main issue to overcome when deploying these language models, as they are computationally intensive and often have to deal with large amounts of data (e.g., collections of documents or web pages). For example, reranking 100 documents for just one query is equivalent to performing 100 predictions in another natural language task (e.g., text classification).

One approach to alleviating this cost is to use knowledge distillation [10]. The concept of distillation refers to a set of compression techniques in which a smaller model, called the student, is trained to reproduce the probability distributions predicted by a larger model, called the teacher. This approach is commonly used to decrease model size and control the effectiveness/efficiency trade-off in neural networks, thus reducing inference costs such as latency and memory requirements. Although knowledge distillation often degrades effectiveness, a potential increase in efficiency can make the trade-off beneficial for certain applications, such as in a text ranking task, since inference needs to be applied over many candidate texts for each query.

It is well known that distilled models fine-tuned on enough annotated data can achieve competitive effectiveness compared to larger models when inferring from similar examples [13, 34]. However, in real-world applications, training and inference data often come from different data distributions, as real-world data is

Figure 1: Distilled rerankers such as MiniLM have similar in-domain effectiveness to models 100 times larger (top). However, large rerankers such as monoT5-3B outperform distilled ones and dense models of equivalent size on zero-shot tasks (bottom).
generally more diverse than the datasets typically used for fine-tuning [5]. This difference can result in poor effectiveness [2, 29].

Another approach that offers fast retrieval is dense models, as vector representations of documents can be precomputed prior to retrieval. At retrieval time, only the query vector is computed, and a fast nearest-neighbor search infrastructure is used to retrieve the most similar document vectors to the query vector. Recent work on dense retrieval mostly compares these models with learned sparse representations [11, 12, 18, 36, 38, 42, 45], while neural rerankers are often ignored due to their computational costs at query time.

However, it is not clear how these retrieval methods compare in zero-shot tasks under different computational budgets. In this work, we begin by showing that distilled models can achieve competitive effectiveness on an in-domain task with models that are 100x larger. However, we show that in a zero-shot setting, models with a large parameter count have a clear superior effectiveness. Furthermore, our multi-billion parameter reranker achieves state-of-the-art results in the BEIR benchmark, outperforming dense retrieval models of similar size. This suggests that current design choices for dense retrieval methods still have much room for improvement.

2 RELATED WORK

Pretrained language models are the state of the art in a wide range of NLP tasks, including text ranking. However, these models often have high computational costs as they consist of hundreds of millions of parameters, which poses challenges for deployment in real-world applications mainly due to latency constraints. To address this challenge SANH et al. [34] proposed an approach to pretrain smaller language models, called DistilBERT, that can be fine-tuned on a wide range of tasks, similar to larger models. They were able to reduce the size of the model by 40% while retaining 97% of its effectiveness and being 60% faster. Jiao et al. [13] proposed a new knowledge distillation method specially designed for transformer-based models to speed up inference and reduce model size while maintaining accuracy. The method performs the distillation of knowledge in the pretraining and task-specific learning stages to ensure that the model can capture the general-domain as well as the task-specific knowledge in BERT. Furthermore, they also presented TinyBERT, an effective distilled model that achieved more than 96.8% of its teacher’s effectiveness on GLUE benchmark [41], while being 7.5x smaller and 9.4x faster during inference.

Sun et al. [37] proposed MobileBERT, a distilled version of BERT-large that can be applied to many downstream tasks through a simple fine-tuning procedure. The authors showed that MobileBERT is 4.3× smaller and 5.5× faster than BERT-base, while achieving competitive results on various benchmarks such as GLUE [41] and SQuAD [28]. Wang et al. [43] presented an effective and simple approach to compress large pretrained transformer models, named deep self-attention distillation. They proposed to distill the self-attention module of the last transformer layer of the teacher model. The results showed that the student model outperforms state-of-the-art baselines while maintaining more than 99% of the accuracy on SQuAD and several GLUE benchmark tasks, and using only 50% of the teacher model parameters. Mukherjee and Awadallah [21] proposed a distillation approach that surpassed all the strategies employed in previous works. They demonstrated that this novel approach leads to massive compression of models, up to 35x in terms of parameters and 51x in terms of latency, while retaining 95% of the score for a named entity recognition task in 41 languages.

Zero-shot models demonstrated state-of-the-art effectiveness on various language tasks. For example, Nogueira et al. [24] proposed monoT5, a novel adaptation of a pretrained sequence-to-sequence language model designed for the text ranking task. The model is fine-tuned to generate a score that measures a document’s relevance to a given query and has achieved state-of-the-art zero-shot results on TREC 2004 Robust Track [40]. Furthermore, this approach significantly outperformed fine-tuned models in data scarcity scenarios. Pradeep et al. [25] used a monoT5 model trained only on a general domain text ranking dataset to reach the best or second best effectiveness on multiple tasks in specific domains, e.g. the medical domain [32], including documents about COVID-19 [47]. Rosa et al. [33], using the same T5-based model developed for text ranking, showed for the legal domain that a zero-shot model can perform better than models fine-tuned on the task itself. They argued that in a limited annotated data scenario, a zero-shot model fine-tuned only on a large general domain dataset may generalize better on unseen data than models fine-tuned on a small domain-specific dataset. Sanh et al. [35] created a system to map any NLP task to prompts written in natural language and converted many supervised datasets into that format. Next, they fine-tuned an 11 billion-parameter T5 model on that combined dataset and found that the model achieves great zero-shot effectiveness across multiple unseen datasets, outperforming models up to 16x its size. Lin et al. [17] fine-tuned a 7.5 billion-parameters multilingual model in a number of languages to study its zero-shot learning capabilities across multiple tasks. The model achieved state-of-the-art results in over 20 languages, outperforming even some larger language models on various tasks such as natural language inference.

Additionally, larger language models demonstrated stronger zero-shot and few-shot capabilities compared to smaller models. Brown et al. [3] presented GPT-3, a language model with 175 billion parameters. The authors evaluated the model’s ability to learn from a few examples and demonstrated that scaling up the number of parameters greatly improves zero-shot and few-shot effectiveness. Wei et al. [44] presented instruction tuning, a novel method to improve the zero-shot learning capacity of scaled language models by providing task-specific instructions using natural language. This method substantially improved zero-shot effectiveness and the model surpassed the GPT-3 on several datasets. Zhong et al. [48] proposed meta-tuning, a method for optimizing zero-shot learning of scaled language models by fine-tuning on a collection of datasets in a question answering format. The model surpassed similar size language models, including state-of-the-art ones. Furthermore, recent results suggest that we can still improve effectiveness by following this trend of increasing model size. To understand the impact of scaling in few-shot learning, Chowdhery et al. [4] introduced the Pathways Language Model (PaLM), a 540 billion parameter language model that has achieved state-of-the-art few-shot results in hundreds of language tasks, also outperforming multiple fine-tuned state-of-the-art models.
Table 1: Results (nDCG@10) on the BEIR benchmark. We use MRR@10 for MS MARCO. All results except MS MARCO are zero-shot. 1 Distilled models.

| Parameters          | BM25  | MinnLI\(^1\) | monoT5 | Dense Models |
|---------------------|-------|---------------|--------|--------------|
|                     |       | 22M | 60M | 220M | 3B       | ColBERT-v2\(^1\) | GTR | SGPT |
| MS MARCO            | 0.1870 | 0.3901 | 0.3566 | 0.3810 | 0.3980 | 0.3970 | 0.3880 | -     |
| TREC-COVID          | 0.5947 | 0.7188 | 0.6928 | 0.7775 | 0.7948 | 0.7380 | 0.5010 | \textbf{0.8730} |
| NCFCorpus           | 0.3218 | 0.3501 | 0.3180 | 0.3570 | \textbf{0.3837} | 0.3380 | 0.3420 | 0.3630 |
| BioASQ              | 0.5224 | 0.5335 | 0.4880 | 0.5240 | \textbf{0.5740} | - | 0.3240 | 0.4130 |
| Natural Questions   | 0.3055 | 0.5525 | 0.4733 | 0.5674 | \textbf{0.6334} | 0.5620 | 0.5680 | 0.5240 |
| HotpotQA            | 0.6330 | 0.7324 | 0.5996 | 0.6950 | \textbf{0.7589} | 0.6670 | 0.5990 | 0.5930 |
| FEVER               | 0.6513 | 0.8180 | 0.7191 | 0.8018 | \textbf{0.8495} | 0.7850 | 0.7400 | 0.7830 |
| Climate-FEVER       | 0.1651 | 0.2555 | 0.2116 | 0.2451 | 0.2802 | 0.1760 | 0.2670 | \textbf{0.3050} |
| DBPedia             | 0.3180 | 0.4652 | 0.3437 | 0.4195 | \textbf{0.4777} | 0.4460 | 0.4080 | 0.3990 |
| TREC-NEWS           | 0.3952 | 0.4464 | 0.3848 | 0.4475 | 0.4727 | - | 0.3460 | \textbf{0.4810} |
| Robust04            | 0.4485 | 0.4801 | 0.4222 | 0.5016 | \textbf{0.6148} | - | 0.5060 | 0.5140 |
| ArguAna             | 0.5302 | 0.5021 | 0.1274 | 0.1946 | 0.3803 | 0.4630 | \textbf{0.5400} | 0.5140 |
| Touc'hé-2020        | \textbf{0.4422} | 0.2812 | 0.2643 | 0.2773 | 0.2995 | 0.2630 | 0.2560 | 0.2540 |
| CQA DupStack        | 0.2788 | 0.3611 | 0.3474 | 0.3808 | \textbf{0.4155} | - | 0.3990 | 0.3810 |
| Quora               | 0.7886 | 0.8037 | 0.8259 | 0.8230 | 0.8407 | - | \textbf{0.8920} | 0.8460 |
| SCIDOCS             | 0.1490 | 0.1629 | 0.1436 | 0.1649 | \textbf{0.1970} | 0.1540 | 0.1610 | \textbf{0.1970} |
| SciFact             | 0.6789 | 0.6812 | 0.6963 | 0.7356 | \textbf{0.7773} | 0.6930 | 0.6620 | 0.7470 |
| FiQA-2018           | 0.2361 | 0.3599 | 0.3377 | 0.4136 | \textbf{0.5137} | 0.3560 | 0.4670 | 0.3720 |
| Signal-1M (RT)      | \textbf{0.3304} | 0.2964 | 0.2711 | 0.2771 | 0.3140 | - | 0.2730 | 0.2670 |
| Avg (excl. MARCO)   | 0.4328 | 0.4889 | 0.4259 | 0.4780 | \textbf{0.5321} | - | 0.4580 | 0.4903 |
| Improvement over BM25| -     | 0.0561 | -0.0069 | 0.0452 | \textbf{0.0993} | - | 0.0252 | 0.0575 |

Domain adaptation is often necessary when applying NLP models to real-world scenarios as it deals with the challenge of transferring knowledge from a source domain to a different domain at the time of inference, i.e., the model must be able to extrapolate well across different domains. Although there is extensive study on the effectiveness of NLP models on in-domain data, we still know little about their ability to extrapolate to out-of-domain datasets. To better understand this issue, Zhan et al. [46] investigated the extrapolation effectiveness of dense retrieval models. They found that dense retrieval models can achieve competitive effectiveness compared to reranker models on in-domain tasks, but extrapolate substantially worse to new domains. Their results also indicated that pretraining the model on target domain data improves the extrapolation capability. Ren et al. [31] presented an extensive and detailed examination of the zero-shot capability of dense retrieval models to better understand them. The authors analyzed key factors related to zero-shot retrieval effectiveness, such as source dataset, potential bias in target dataset and also reviewed and compared different dense retrieval models. Ni et al. [22] presented Generalizable T5-based dense Retrievers (GTR), a model designed to deal with the problem of poor effectiveness outside the domain of previous dual encoder models. The authors found that increasing the size of the model brings a significant improvement in out-of-domain generalization. The model outperformed existing sparse and dense models in a variety of retrieval tasks, while being fine-tuned on only 10% of the MS MARCO dataset. Muennighoff [20] argued that although GPT transformers are the largest language models available, information retrieval tasks are dominated by encoder-only or encoder-decoder transformers. To change this scenario, Muennighoff [20] presented the SGPT model, a 5.8 billion parameter model that achieved state-of-the-art results on several out-of-domain information retrieval datasets and also outperformed much larger models with up to 175 billion parameters.

3 EXPERIMENTS

In our experiments, we evaluate three different models: BM25 [15], MiniLM-L6 [43] and monoT5 [24] in three different sizes: 60M (small), 220M (base) and 3B parameters. We use BM25 implemented in Pyserini [15], a Python library for information retrieval. MiniLM is a distilled reranker proposed by Wang et al. [43] that achieves competitive effectiveness compared to state-of-the-art models in multiple retrieval tasks. MonoT5 is an adaptation of the T5 model [27] proposed by Nogueira et al. [24]. The model takes a query-document pair as input and is designed to generate a probability score that quantifies the relevance between them. All three of our transformer models are fine-tuned on MS MARCO [1] and available at Hugging Face. 1 MS MARCO is a large-scale dataset for passage ranking task, consistent of 8.8 million passages taken from the Bing search engine. The fine-tuning procedures we use in this work are

---

1https://huggingface.co/castorini/monot5-3B-msmarco-10k,
https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2,
https://huggingface.co/castorini/monot5-base-msmarco-10k,
https://huggingface.co/castorini/monot5-3B-msmarco-10k
described in detail in Reimers and Gurevych [30] and Nogueira et al. [24] for MiniLM and monoT5, respectively. At inference time, rerankers sort a list of 1000 texts retrieved by BM25 according to their relevance to a query.

We evaluate the zero-shot effectiveness of our models on the BEIR benchmark Thakur et al. [39], which consists of 18 publicly available datasets from different domains, such as web, biomedical and financial. With the exception of the MS MARCO dataset, all of our results are zero-shot as our models are fine-tuned on MS MARCO and directly evaluated on other datasets.

4 RESULTS
Results in Table 1 show that while MiniLM achieves excellent results on MS MARCO, including competitive effectiveness to a much larger model such as monoT5-3B, the distilled model underperforms in comparison to monoT5-3B on almost all datasets outside the domain seen during fine-tuning. For example, on the FiQA dataset, which comprises text in the financial domain, MiniLM performs nearly 10 points below our largest reranker. Other significant results include the experiments on the SciFact and Natural Questions datasets, in which the MiniLM is also outperformed by monoT5-3B by almost 10 and 8 points, respectively. Nonetheless, average results show that MiniLM achieves competitive effectiveness compared to monoT5-base, which has 10 times more parameters, suggesting that MiniLM is able to retain some degree of generalization capability.

As shown in Figure 1, the average results of monoT5-3B and SGPT-5.8B demonstrate that strong zero-shot effectiveness in new text domains can be achieved by increasing the number of model parameters and without fine-tuning on in-domain data. However, our 3 billion parameter reranker model outperformed SGPT, a state-of-the-art dense retrieval model almost twice as large, and ColBERTv2 [36], a distilled dense model. Another interesting comparison is that monoT5 and GTR use the same underlying model, T5, but the reranker has better effectiveness than the dense retriever. GTR’s effectiveness increases at a slower rate with more parameters in comparison to monoT5’s. We suspect that this “saturation” is due to the decision to keep the size of document and query vectors fixed to 768 dimensions as the model size increases. Results for the MiniLM model reveal a similar behavior: the MiniLM reranker outperforms its dense version on both MS MARCO and BEIR. These results suggest that early query-document interactions in the reranker architecture and a high number of parameters (and hidden dimensions) have a positive impact on zero-shot effectiveness.

4.1 Latency
A major drawback of rerankers in comparison to dense retrievers is their latency, as the former requires one inference pass of a potentially large neural network per query-document pair, while dense retrievers can compute the vector representation of documents prior to retrieval, thus saving considerable computation at query time.

In Table 2 we compare effectiveness (nDCG@10 on BEIR) with efficiency (seconds/query) of rerankers of different sizes. Our experiments were performed on an A100 GPU with 40GB. Despite receiving only 50 documents per query, monoT5-3B consistently outperforms all other smaller models, while performing almost as efficiently as MiniLM, a distilled model just under 100 times smaller.

5 CONCLUSION
In this work we studied how distillation and parameter count influence the zero-shot effectiveness of neural retrievers. We begin by showing that in-domain effectiveness, i.e., when retrievers are finetuned and evaluated on the same dataset such as MS MARCO, is not a good proxy for zero-shot effectiveness, which corroborates recent claims by Lin et al. [14] and Gupta et al. [9] and Zhan et al. [46].

Furthermore, we show that a distilled reranker has better zero-shot effectiveness than much larger non-distilled rerankers, which is an important and desirable feature of deployed models. However, our largest reranker significantly outperforms smaller rerankers and achieves a new state of the art across almost all datasets used in our zero-shot experiments. This suggests that a large number of parameters may play a significant role in the generalization capability of pretrained language models.

Lastly, our results confirm the findings of Zhan et al. [46], who show that dense retrievers have poorer generalization than rerankers to new domains. Our study, however, shows that this is the case across a wide range of model sizes, suggesting that much work needs to be done in dense retrieval methods.

| Reranker | Params | docs/query | nDCG@10 | s/query |
|----------|--------|------------|---------|--------|
| MiniLM | 22M | 1000 | 0.4889 | 0.64 |
| monoT5 | 60M | 1000 | 0.4259 | 2.02 |
| monoT5 | 220M | 1000 | 0.4780 | 3.00 |
| monoT5 | 3B | 1000 | 0.5321 | 14.63 |
| monoT5 | 3B | 50 | 0.5152 | 0.73 |

Table 2: Rerankers effectiveness (nDCG@10) vs efficiency (s/query) on BEIR. Candidate document sets are from BM25.

REFERENCES
[1] Payal Bajaj, Daniel Campos, Nick Crawwell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberger, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. arXiv preprint arXiv:1611.09268 (2018).
[2] Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains. arXiv preprint arXiv:2102.12206 (2021).
[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Pranjal Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsyvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Iard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Heryuk Michalewski, Xavier Garcia, Vedant Mistry, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyuntaek Rosa, et al.
[47] Edwin Zhang, Nikhil Gupta, Rodrigo Nogueira, Kyunghyun Cho, and Jimmy Lin. 2020. Rapidly Deploying a Neural Search Engine for the COVID-19 Open Research Dataset. In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020.

[48] Ruiqi Zhong, Kristy Lee, and Dan Klein Zheng Zhang. 2021. Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections. arXiv preprint arXiv:2104.04670 (2021).
A APPENDIX

In this section, we provide graphs that show the zero-shot effectiveness when scaling the parameter count for each dataset on the BEIR benchmark.

Figure 2: Model size vs effectiveness on in-domain (MS MARCO) vs out-of-domain (others) data. Effectiveness increase with respect to the number of model parameters only on out-of-domain datasets.