Huge Frozen Language Models as Readers for Open-Domain Question Answering

Anonymous Authors[1]

Abstract

In the open-book variant of the open-domain question-answering setting, an answer generator typically attends to 100+ retrieved documents when answering, and is thus often called a *reader*. Current readers are fine tuned for this long-context functionality. Because it is prohibitively expensive to fine tune huge models to attend to 100+ retrieved documents, readers tend to be relatively small, typically having fewer than 1B parameters. We introduce huge LMs into this pipeline as frozen readers. To do so, we use a re-ranking stage to condense relevant information from 100+ retrieved documents into the input sequence length of the frozen LM reader. We show that frozen LMs can reach and surpass leading fine tuning approaches on Natural Questions, a prominent open-domain question answering benchmark.

1. Introduction

The dominant approach for performing open-domain question answering (ODQA) is the *retrieve–read* framework (Chen et al., 2017; Lee et al., 2019; Karpukhin et al., 2020), also referred to as open-book question answering. Given a question, this approach first employs a *retriever* over a large *evidence corpus* (e.g. Wikipedia) to fetch a set of relevant documents that may contain the answer (typically, on the order of 100 documents are retrieved). A retrieval-augmented *reader* is then used to answer the question given these documents. Standard pretrained LMs are trained on context windows much shorter than 100 documents, and so they require long-context fine tuning in order to be used as readers. This operation is very expensive—prohibitively so for large LMs. Therefore, leading readers do not typically exceed 1B parameters (Karpukhin et al., 2020; Izacard & Grave, 2020a).

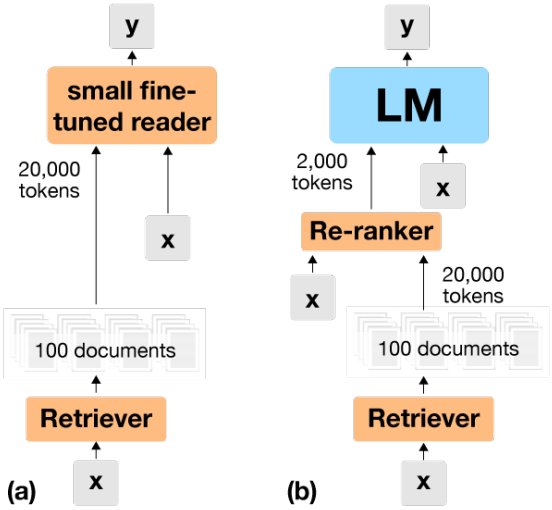

Figure 1: **(a)** The existing retrieve–read framework for open-domain question answering involves fine-tuning readers of specialized architectures with large context windows. **(b)** We re-rank the retrieved documents to increase the probability of the answer reaching the frozen LM context window. Blue indicates a "frozen", non-trained module; orange indicates a trained module.

An inherent drawback of relying on small retrieval-augmented readers is that they do not enjoy the world knowledge or deduction capabilities of huge LMs. There is thus an opportunity in combining the power of strong supervised retrievers with that of huge LMs. To address this, we used an external re-ranking module for increasing the chance of getting the answer in a small amount of passages that fits into the frozen LM's context window. While the retriever relevance scores are computed based on separate dense representations of the question and passage (Karpukhin et al., 2020; Ram et al., 2022), the re-ranker predicts each document's relevance score after jointly attending to both the question and the passage (Karpukhin et al., 2020). We prompt tune the frozen LM to extract answers from re-ranked documents

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2022 workshop on Knowledge Retrieval and Language Models. Do not distribute.

that appear in its context.

Our simple re-ranking approach facilitates non trivial performance by the frozen LM reader. Our results show that a frozen J1-Grande-17B model can surpass the score of the fine-tuned Fusion-in-Decoder (FiD) model of Izacard & Grave (2020a) on the (open) Natural Questions benchmark (Kwiatkowski et al., 2019), when both are given access to the same set of retrieved documents. We further boost the results by utilizing a stronger retrieval system, namely a hybrid approach combining Spider (Ram et al., 2022) and BM25 (Robertson & Zaragoza, 2009). Our frozen LM reader was able to perform significantly better than the strong end-to-end-trained EMDR^2 (Singh et al., 2021) and on par with the distilled-retriever FiD model of Izacard & Grave (2020b), both prominent fine-tuned models.

2. Experimental setup

At a high level, we trained a re-ranker to produce improved passage relevance scores by jointly attending to the question and passage. We then greedily added passages to our context in descending order, until the context length of our frozen LM reader was full. We thus prepared training data for prompt tuning our frozen LMs to serve as readers. The full details of our experimental setup follow.

Dataset & Evidence Corpus. We used the open-domain version of the popular Natural Questions ("NQ") benchmark (Kwiatkowski et al. (2019)), which was popularized by Lee et al. (2019) and has since been widely used for ODQA. The training data consists of $\sim$80K questions along with gold annotations of answers. As evidence corpus, we adopted the Wikipedia corpus as Karpukhin et al. (2020), which consists of roughly 21 million passages of 100 words each.

Retrievers. To generate inputs for our re-ranker, we experimented with two different retrievers from the literature.

- **DPR-NQ** (Karpukhin et al., 2020): A supervised dense retriever trained in a contrastive fashion on NQ.
- **Spider-NQ + BM25** (Ram et al., 2022; Robertson & Zaragoza, 2009): A self-supervised dense retriever trained on the recurring span retrieval task. Here we use the hybrid model described in Ram et al. (2022), where the dense retriever is Spider, fine-tuned on NQ (similar to DPR) and the sparse model is BM25 (Robertson & Zaragoza, 2009).

Frozen LMs We experiment with the 7B parameter of Lieber et al. (2021), J1-Large, and its 17B parameter counterpart, J1-Grande (Lieber et al., 2022).

Re-ranker Training. We trained our re-ranker following the same protocol used by Karpukhin et al. (2020) to train their extractive reader. We based the re-ranker architecture on the 110M parameter BERT-base (Devlin et al., 2019) model, such that a forward pass through the re-ranker incurs negligible run-time cost relative to a single pass through the 7B/17B parameter frozen LMs. During training, we sampled one positive and 23 negative passages from the top 100 passages returned by the retrieval system for each question. The training objective was to maximize the marginal log-likelihood of the start and end of all the correct answer spans in the positive passage (the answer string may appear multiple times in one passage), combined with the log-likelihood of the positive passage being selected. We used a batch size of 16, and trained the re-ranker for up to 30K steps with a learning rate of $1 \cdot 10^{-5}$ using Adam (Kingma & Ba, 2014), linear scheduling with warm-up, and dropout rate of 0.1. Contemporary work (Anonymous, 2022) investigates a similar form of re-ranking, for the benefit of a fine-tuned reader.

Preparing data for prompt tuning. At inference time, we discarded the start and end scores of the extractive reader, and only used its passage-level scores as re-ranking scores. Given those, we greedily added passages to our context in descending order, until the context length of our frozen LM reader was full. We note an important subtlety in the way we prepared the data used to prompt tune our LMs. In initial experiments training the re-ranker, we observed clear overfitting on the training set: our re-ranker performed especially well on the inputs used to train it. We did not want this bias to impact our prompt tuning, which of course we wanted to generalize to test data. Therefore, we randomly split the training set into two halves, denoted training-A and training-B, over which we trained two re-rankers, denoted re-ranker-A and re-ranker-B. We then used re-ranker-B to process the training-A data and likewise used re-ranker-A to process the training-B data, merging the two to yield our LM prompt tuning training set. We trained a third re-ranker on the entire training set, denoted re-ranker-All, and used it in order to create the data for the development and test sets.

Prompt tuning. We prompt tuned our frozen J1-Large-7B and J1-Grande-17B LMs to serve as readers over the data prepared by the re-ranker. We used batch size 32, and considered learning rates in $\{1 \cdot 10^{-1}, 5 \cdot 10^{-1}\}$ for J1-Large and $\{3 \cdot 10^{-2}, 1 \cdot 10^{-1}\}$ for J1-Grande, reporting the best results on the development set and measuring test scores for the best development set configuration.

Baselines. We compare our model to numerous popular baselines, all of which are generative. Specifically, we consider RAG (Lewis et al., 2020), Retro (Borgeaud et al., 2021), EMDR^2 (Singh et al., 2021) and FiD/FiD-Distill

Passage score	Reader	Retriever	Recall @ J1 input	Avg. #docs	Dev EM
Retriever	J1-Large-7B	DPR	77.2	17	46.6
Re-ranker	J1-Large-7B	DPR	80.4	17	48.7
Retriever	J1-Large-7B	Spider+BM25	81.4	17	49.5
Re-ranker	J1-Large-7B	Spider+BM25	83.2	17	50.8

Table 1: A comparison between the Natural Questions development set exact match (EM) scores when greedily packing documents according to original retriever scores or to our trained re-ranker scores. Recall @ J1 input measures recall on the development set of the correct answer being shown to the frozen J1-Large LM in its context window, which on average can contain 17 of the 100 retrieved passages. The re-ranking technique boosts the performance of the frozen reader, as it exposes the correct answer to the frozen LM more often.

Model	Reader	Retriever	Test EM
RAG (Lewis et al., 2020)	Fine-tuned BART-Large	DPR	44.5
Retro (Borgeaud et al., 2021)	Fine-tuned Retro 7.5B	DPR	45.5
FiD (Izacard & Grave, 2020a)	Fine-tuned T5-Large	DPR	51.4
Frozen LM reader, no re-ranker	J1-Large-7B	DPR	48.8
Frozen LM reader (Ours)	J1-Large-7B	DPR	49.9
Frozen LM reader (Ours)	J1-Grande-17B	DPR	**51.6**
EMDR^2 (Singh et al., 2021)	Fine-tuned T5-Base	EMDR^2	52.5
FiD-Distill (Izacard & Grave, 2020b)	Fine-tuned T5-Large	Distilled DPR	**53.7**
Frozen LM reader (Ours)	J1-Large-7B	Spider+BM25	51.9
Frozen LM reader (Ours)	J1-Grande-17B	Spider+BM25	**53.7**

Table 2: Exact match (EM) results on the test set of Natural Questions for different generative approaches. The frozen J1-Grande-17B model performs best among fine-tuned models using DPR as their retriever (upper part). In addition, it surpasses or matches prominent fine tuning methods which use stronger retrievers (bottom part).

(Izacard & Grave, 2020a;b). For fair comparison, we differentiate models that use DPR for retrieval from those that leverage stronger ones.

Ablations. To help us to understand the contribution of the re-ranking module, we ran the same experiment when greedily packing passages into the context window of the frozen LM based on the original retriever relevance scores, which are computed based on separate dense representations of the question and passage.

3. Results

We now describe our experimental results. Table 1 shows the utility of using a re-ranker when packing documents into the context window of our LM, which on average can contain 17 of the 100 retrieved passages. When using DPR (Karpukhin et al., 2020) as our retrieval system, we increased the recall at the input to our LM (*i.e.*, the percentage of questions for which the answer appears in the context window of the frozen LM) from 77.2% to 80.4%, thereby improving downstream performance (measured by exact match) by 2.1 points (from 46.6% to 48.7%). Similarly, we observed significant gains from re-ranking when leveraging stronger retrievers like Spider+BM25.

Table 2 shows the results of our systems on the test set of NQ, compared to various generative baselines. In the setting where all models use the same retriever—DPR—our frozen J1-Grande-17B reader obtained the best result, surpassing the score of the FiD model (Izacard & Grave, 2020a) which was fine-tuned to attend to all 100 retrieved documents at decoding time.

Our frozen J1-Large-7B outperformed the similarly-sized Retro-7.5B model (Borgeaud et al., 2021), which has a similar decoder-only architecture, but was highly customized to the open-book setting: it was pretrained with a retrieval component and then fine tuned to attend to 20 passages. The frozen J1-Large-7B surpassed Retro by 3.3 points with no re-ranker, *i.e.*, when the $\sim$ 17 passages shown at its input are a subset of the 20 passages shown to Retro, showing that frozen, decoder-only LMs can outperform specialized ODQA reader architectures when given the same set of retrieved documents. J1-Large-7B surpassed Retro by 4.4

points when the ∼17 passages at its input are re-ranked.

When not limited to the DPR retriever, our frozen J1-Grande-17B matched the performance of the strong fine-tuned FiD-Distill model (Izacard & Grave, 2020b), and outperformed EMDR^2 (Singh et al., 2021), which jointly fine tuned both retriever and reader end-to-end.

4. Conclusion

Overall, our results demonstrate that huge frozen language models serve as excellent readers for open domain question answering, and do not fall behind more elaborate prominent fine-tuned readers.

Anonymous. Re2g: Retrieve, rerank, generate, 2022. URL https://openreview.net/forum?id=_R7UMusdRsc.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Driessche, G. v. d., Lespiau, J.-B., Damoc, B., Clark, A., Casas, D. d. L., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J. W., Elsen, E., and Sifre, L. Improving language models by retrieving from trillions of tokens, 2021. URL https://arxiv.org/abs/2112.04426.

Chen, D., Fisch, A., Weston, J., and Bordes, A. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Izacard, G. and Grave, E. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020a.

Izacard, G. and Grave, E. Distilling knowledge from reader to retriever for question answering. *CoRR*, abs/2012.04584, 2020b. URL https://arxiv.org/abs/2012.04584.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Lee, K., Chang, M.-W., and Toutanova, K. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL https://aclanthology.org/P19-1612.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Lieber, O., Sharir, O., Lenz, B., and Shoham, Y. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*, 2021.

Lieber, O., Shahaf, G., Asida, T., Padnos, D., and Lenz, B. https://www.ai21.com/blog/introducing-j1-grande, 2022.

Ram, O., Shachaf, G., Levy, O., Berant, J., and Globerson, A. Learning to retrieve passages without supervision. In *North American Association for Computational Linguistics (NAACL)*, 2022.

Robertson, S. and Zaragoza, H. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.

1561/1500000019. URL https://doi.org/10.1561/1500000019.

Singh, D., Reddy, S., Hamilton, W., Dyer, C., and Yogatama, D. End-to-end training of multi-document reader and retriever for open-domain question answering. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 25968–25981. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/da3fde159d754a2555eaa198d2d105b2-Paper.pdf.