

Reader-Guided Passage Reranking for Open-Domain Question Answering

Yuning Mao^{1*}, Pengcheng He², Xiaodong Liu³, Yelong Shen²,
Jianfeng Gao³, Jiawei Han¹, Weizhu Chen²

¹University of Illinois, Urbana-Champaign ²Microsoft Dynamics 365 AI ³Microsoft Research

¹{yuningm2, hanj}@illinois.edu

^{2,3}{penhe, xiaodl, yeshe, jfgao, wzchen}@microsoft.com

Abstract

Current open-domain question answering (QA) systems often follow a Retriever-Reader (R2) architecture, where the retriever first retrieves relevant passages and the reader then reads the retrieved passages to form an answer. In this paper, we propose a simple and effective passage reranking method, **Reader-guIDEd Reranker (RIDER)**, which does not involve any training and reranks the retrieved passages solely based on the top predictions of the reader before reranking. We show that RIDER, despite its simplicity, achieves 10 to 20 absolute gains in top-1 retrieval accuracy and 1 to 4 Exact Match (EM) score gains without refining the retriever or reader. In particular, RIDER achieves 48.3 EM on the Natural Questions dataset and 66.4 on the TriviaQA dataset when only 1,024 tokens (7.8 passages on average) are used as the reader input.

1 Introduction

Current open-domain question answering (QA) systems often follow a Retriever-Reader (R2) architecture, where the retriever first retrieves relevant passages and the reader then reads the retrieved passages to form an answer. Since the retriever retrieves passages from a large candidate pool (*e.g.*, millions of Wikipedia passages), it often fails to rank the most relevant passages at the very top. One line of work (Mao et al., 2020; Karpukhin et al., 2020) aims to improve the quality of passage retrieval and shows that significantly better retrieval accuracy as well as QA performance can be achieved when the retriever is improved.

An alternative solution is to rerank the initial retrieval results via a reranker, which is widely used in information retrieval (Nogueira and Cho, 2019; Qiao et al., 2019) and explored in early open-domain QA systems (Wang et al., 2018a;

Lee et al., 2018). However, current state-of-the-art open-domain QA systems (Karpukhin et al., 2020; Izacard and Grave, 2020b; Lewis et al., 2020) do not distinguish the order of the retrieved passages and instead equally consider a large number of retrieved passages (*e.g.*, 100), which could be computationally prohibitive as the model size of the readers becomes larger (Izacard and Grave, 2020b).

We argue that a Retriever-Reranker-Reader (R3) architecture is beneficial in terms of both model effectiveness and efficiency: passage reranking improves the retrieval accuracy of the retriever at top positions and allows the reader to achieve comparable performance with fewer passages as the input. However, one bottleneck of R3 is that the reranker, previously based on BiLSTM (Wang et al., 2018a; Lee et al., 2018) and nowadays typically BERT-based cross-encoder (Nogueira and Cho, 2019; Qiao et al., 2019), is costly to train and slows down the whole pipeline as well.

Can we achieve better performance without the bother of training an expensive reranker or refining the retriever (reader)? In this paper, we propose a simple and effective passage reranking method, named **Reader-guIDEd Reranker (RIDER)**, which does not require any training and reranks the retrieved passages solely based on their lexical overlap with the top predicted answers of the reader before reranking. Intuitively, the top predictions of the reader are closely related to the ground-truth answer and even if the predicted answers are partially correct (or incorrect), they may still provide useful signals suggesting which passages may contain the correct answer (Mao et al., 2020).

We conduct experiments on the Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Trivia) (Joshi et al., 2017) datasets. We demonstrate that R3 with RIDER, without any additional training, achieves 10 to 20 absolute gains in top-1 retrieval accuracy and 1 to 4 gains in Exact Match (EM) compared with the R2 architecture.

*Work was done during Yuning’s internship at Microsoft Dynamics 365 AI.

Notably, using only 1,024 tokens (7.8 passages on average) as the input of a generative reader, RIDER achieves EM=47.5/63.5 on NQ/Trivia when the predictions of the same generative reader (EM=45.3/62.2 in R2) are used for reranking, and EM=48.3/66.4 on NQ/Trivia when the predictions of an extractive reader (EM=43.8/62.7 in R2, accessible to more passages) are used for reranking.

Contributions. (1) We propose Reader-guIDed Reranker (RIDER), a simple and effective passage reranking method, which reranks the retriever results by the reader predictions without any training and can be easily applied to existing R2 open-domain QA systems for further improvements. (2) We demonstrate that the passages reranked by RIDER achieve significantly better retrieval accuracy and consequently lead to better QA performance without refining the retriever or reader. (3) Notably, R3 with RIDER as the reranker achieves comparable or better performance than state-of-the-art methods on two benchmark datasets when only 1,024 tokens are used as the reader input.

2 Method

2.1 Task Formulation

We assume that an open-domain QA system with an R2 architecture is provided. We denote the initially retrieved passages of the retriever as \mathbf{R} . We denote the top- N predictions of the reader on the top- k passages of \mathbf{R} (denoted as $\mathbf{R}^{[k]}$) as $\mathbf{A}^{[N]}$. The goal of RIDER is to rerank \mathbf{R} to \mathbf{R}' using $\mathbf{A}^{[N]}$ such that the retrieval accuracy is improved and better end-to-end QA results are achieved when $\mathbf{R}'^{[k]}$ is used as the reader input instead of $\mathbf{R}^{[k]}$.

2.2 Passage Reranking

Given an initially retrieved passage list \mathbf{R} and top- N predictions of the reader $\mathbf{A}^{[N]}$, RIDER forms a reranked passage list \mathbf{R}' as follows. RIDER scans \mathbf{R} from the beginning and appends to \mathbf{R}' every passage $p \in \mathbf{R}$ that contains any reader prediction $a \in \mathbf{A}^{[N]}$. Then, the remaining passages are appended to \mathbf{R}' according to their original order. Despite its simplicity, we observe that RIDER leads to consistent gains in terms of both retrieval accuracy and QA performance without refining the retriever (reader) or even any training itself.

2.3 Passage Reading

We consider a scenario where the number of passages that can be used as the reader input is limited,

which is common in reality due to efficiency consideration (e.g., real-time response) or model capacity (e.g., input length limit). We use a generative reader initialized by BART-large (Lewis et al., 2019), which concatenates the question and top-10 retrieved passages (without reranking during training), trims them to 1,024 tokens (7.8 passages are left on average) as the input, and learns to generate the answer in a Seq2Seq manner (Mao et al., 2020; Min et al., 2020). We further add a shuffle strategy during the training of the reader by randomly shuffling the top retrieved passages before concatenation. In this way, the reader appears to be more robust to the reranked passages during inference and achieve better performance.

3 Experiments

3.1 Experimental Setup

Datasets. We conduct experiments on the open-domain version of two widely used QA benchmarks – Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Trivia) (Joshi et al., 2017), whose statistics are listed in Table 1.

Dataset	Train / Val / Test	Q-len	A-len	#-A
NQ	79,168 / 8,757 / 3,610	12.5	5.2	1.2
Trivia	78,785 / 8,837 / 11,313	20.2	5.5	13.7

Table 1: Dataset statistics that show the number of samples, the average question (answer) length, and the average number of answers for each question.

Evaluation Metrics. Following prior studies (Mao et al., 2020; Karpukhin et al., 2020), we use top- k retrieval accuracy to evaluate the performance of the retriever and Exact Match (EM) to measure the performance of the reader. *Top- k retrieval accuracy* is the proportion of questions for which the top- k retrieved passages contain (at least) one answer span. One can use it as an upper bound of how many questions are answerable by an extractive reader. *Exact Match (EM)* is the proportion of the predicted answer spans being exactly the same as one of the ground-truth answers, after string normalization such as article and punctuation removal.

Source of \mathbf{R} . We take the top retrieved passages of GAR (Mao et al., 2020) on Trivia and its combination with DPR (Karpukhin et al., 2020) on NQ as the initial retrieval results \mathbf{R} for reranking.

Source of $\mathbf{A}^{[N]}$. To obtain the top- N predicted answers for RIDER, we first use the generative reader \mathbf{G} in Sec. 2.3 that is trained on the passages with-

Data Input	NQ					Trivia				
	Top-1	Top-5	Top-10	Top-20	Top-100	Top-1	Top-5	Top-10	Top-20	Top-100
R	46.8	70.7	77.0	81.5	88.9	53.2	73.1	77.0	80.4	85.7
R' by G (N=1)	58.6	71.4	76.9	81.6	88.9	68.8	74.8	77.5	80.4	85.7
R' by G (N=10)	56.4	72.2	77.3	81.6	88.9	66.9	75.3	77.9	80.8	85.7
R' by E (N=1)	60.4	72.1	77.3	81.7	88.9	71.9	77.5	79.8	81.8	85.7
R' by E (N=5)	53.5	75.2	80.0	83.2	88.9	63.2	77.9	80.7	82.8	85.7
R' by E (N=10)	50.3	74.3	80.0	84.2	88.9	59.8	77.1	80.5	82.9	85.7

Table 2: **Top-k retrieval accuracy on the test sets before and after reranking.** **G** and **E** denote generative and extractive readers, respectively, whose top predictions are used for reranking.

Dataset	Top-1	Top-3	Top-5	Top-10
NQ	43.8 (45.3)	57.4	62.6	67.8 (54.2)
Trivia	62.7 (61.8)	72.6	75.5	78.5 (67.7)

Table 3: **EM of top-N predictions of the reader.** Results are mostly on reader **E**. Only top-1 and top-10 EM are shown (in the brackets) for reader **G**, as its 10 predictions are sampled without particular order.

out reranking and used for final passage reading in R3, which represents an apple-to-apple comparison to the R2 architecture without any additional information but higher-quality input. We also experiment with an extractive reader **E** that has access to more passages, where the goal is to study whether we can rerank passages via other signals and further improve **G** such that it outperforms both **G** and **E** when they are in R2. We use the extractive reader in Mao et al. (2020) with BERT-base (Devlin et al., 2019) representation and span voting. For the generative reader, we either take its top-1 prediction with greedy decoding or sample 10 answers with appropriate decoding parameters (*e.g.*, temperature).¹ For the extractive reader, the top predictions are the text spans with the highest scores and we set $N = 1, 5, 10$.

3.2 Experimental Results

3.2.1 Quality of Reranking Signals

We first analyze the EM of the top-N reader predictions $A^{[N]}$ used for reranking, *i.e.*, we consider a question correctly answered as long as one of the top-N predictions matches the answer. The standard EM is a special case with $N = 1$. As listed in Table 3, the reader EM can be improved by up to 24 on NQ and 15.8 on Trivia if we consider the top-10 predictions instead of only the first prediction, suggesting that there is significant potential if we leverage multiple answer candidates for reranking.

¹There are duplicate samples and on average $\bar{N} = 6$.

	Method	NQ	Trivia
Extractive	Hard EM (Min et al., 2019a)	28.1	50.9
	Path Retriever (Asai et al., 2019)	32.6	-
	ORQA (Lee et al., 2019)	33.3	45.0
	Graph Retriever (Min et al., 2019b)	34.5	56.0
	REALM (Gua et al., 2020)	40.4	-
	DPR (Karpukhin et al., 2020)	41.5	57.9
	BM25 (Mao et al., 2020)	37.7	60.1
	GAR	41.8	62.7
	GAR+DPR	43.8	-
Generative	GPT-3 (Brown et al., 2020)	29.9	-
	T5 (Roberts et al., 2020)	36.6	60.5
	SpanSeqGen (Min et al., 2020)	42.2	-
	RAG (Lewis et al., 2020)	44.5	56.1
	FID-base (Izacard and Grave, 2020b)	48.2	65.0
	FID-large (Izacard and Grave, 2020b)	51.4	67.6
	BM25 (Mao et al., 2020)	35.3	58.6
	GAR (Mao et al., 2020)	38.1	61.8
	RIDER (GAR)	-	66.4
	GAR+DPR (Mao et al., 2020)	45.3	-
	RIDER (GAR+DPR)	48.3	-

Table 4: **End-to-end QA comparison of state-of-the-art methods in EM.**

3.2.2 Passage Retrieval

We list the top-k retrieval accuracy before and after passage reranking in Table 2. RIDER significantly improves the retrieval accuracy at top positions (especially top-1) without refining the retriever. In particular, we observe that when taking more reader predictions (*i.e.*, larger N), the top-k retriever accuracy tends to improve more at a larger k and less at a smaller k . For example, an improvement of about 3 points is achieved for top-5 and top-10 accuracy when increasing N from 1 to 5 on NQ for reader **E**, but the top-1 retrieval accuracy also drops significantly (although still better than without reranking), which shows that there is a trade-off between the answer coverage and noise. Note that the top-100 retrieval accuracy is unchanged after reranking since we rerank the top-100 passages.

3.2.3 Passage Reading

Comparison w. the state-of-the-art. We show the QA performance comparison between RIDER and state-of-the-art methods in Table 4. We observe that RIDER improves GAR (or GAR+DPR) on both datasets by a large margin, despite that they use the same generative reader and no further model training is conducted. Such results indicate that RIDER provides higher-quality input for the reader and better performance can be achieved with the same input length. Moreover, the results of RIDER are better than most of the methods that take much more passages as input, except for FID-large (Izacard and Grave, 2020b) that takes 100 passages as input and has more model parameters.

Ablation Study. A detailed analysis of RIDER with different reranking signals is shown in Table 5. We can see that by reranking based on the prediction of the generative reader G (with input $R^{[:k]}$), RIDER achieves around 1 to 2 gains in EM, which shows that RIDER can improve end-to-end QA performance without any additional information. By iterative reranking (R'') using the reader predictions after first reranking, the performance of RIDER is further improved. RIDER achieves even better performance when using the predictions of the extractive reader E (with input R) for reranking, which is consistent with the better results of E on the retrieval accuracy after reranking. Also, it is interesting to see that RIDER outperforms the extractive reader that has access to more passages.

Data Input		NQ	Trivia
R		45.3	62.2
R'_1 by G (N=1)	(45.3+1.1)	46.4	(62.2+0.7) 62.9
R'_2 by G (N=10)	(45.3+2.1)	47.4	(62.2+0.9) 63.1
R'' by R'_1	(46.4+1.1)	47.5	(62.9+0.6) 63.5
R' by E (N=1)	(43.8+3.2)	47.0	(62.7+3.4) 66.1
R' by E (N=5)	(43.8+4.5)	48.3	(62.7+2.5) 65.2
R'' by best R'	(48.3+0)	48.3	(66.1+0.3) 66.4

Table 5: **Performance comparison of RIDER in EM when different reranking signals are used.** The numbers in the brackets represent the performance of the reader used for reranking and relative gains.

4 Related Work

Reranking for Open-domain QA. Reranking has been widely used in information retrieval to refine the initial retrieval results. Early effort on passage reranking for open-domain QA uses su-

pervised (Lee et al., 2018) or reinforcement learning (Wang et al., 2018a) based on BiLSTM. More recently, BERT-based rerankers that treat the query and passage as a sentence pair (*i.e.*, cross-encoders) achieve superior performance (Nogueira and Cho, 2019; Qiao et al., 2019). However, the training of cross-encoders is rather costly. Moreover, the representations of cross-encoders cannot be pre-computed and matched via Maximum Inner Product Search (MIPS) as in bi-encoders (Karpukhin et al., 2020) but measured online between the query and each passage, which results in slower inference as well. Another line of work (Das et al., 2018; Qi et al., 2020) reranks the passages by updating the query and often involves a complicated learning process such as R2 interactions. Alternatively, some prior studies (Wang et al., 2018b; Iyer et al., 2020) directly rerank the top predicted answers instead of the passages using either simple heuristics or additional training. In contrast, RIDER utilizes downstream signals (*i.e.*, the predictions of a reader) to rerank the passages based on the surface forms of the texts without any training.

Retriever-Reader Distillation. Recent studies (Izacard and Grave, 2020a; Yang and Seo, 2020) show that distillation from the preference of the reader can improve the retriever performance, where the reader preference is measured by the attention scores of the reader over different passages and the retriever is refined by learning to approximate the scores. RIDER, to some extent, can also be seen as one way to distill the reader, but RIDER is much simpler in that no further training is involved for either the retriever or reader, and explicit reader predictions instead of latent attention scores are leveraged to improve the retriever results directly.

5 Conclusion

In this work, we propose RIDER, a simple and effective passage reranking method for open-domain QA that does not involve any training or computationally expensive inference. RIDER can be easily integrated into existing R2 systems for further improvements. Without fine-tuning the retriever or reader, RIDER improves the retrieval accuracy and the QA results on two benchmark datasets significantly. In particular, RIDER achieves comparable or better performance than state-of-the-art methods with less reader input and allows for more efficient open-domain QA systems.

Acknowledgments

We thank Hao Cheng and Sewon Min for helpful discussions.

References

- Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2018. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations*.
- 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.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.
- Srinivasan Iyer, Sewon Min, Yashar Mehdad, and Wen-tau Yih. 2020. Reconsider: Re-ranking using span-focused cross-attention for open domain question answering. *arXiv preprint arXiv:2010.10757*.
- Gautier Izacard and Edouard Grave. 2020a. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.
- Gautier Izacard and Edouard Grave. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Jinhyuk Lee, Seongjun Yun, Hyunjae Kim, Miyoung Ko, and Jaewoo Kang. 2018. Ranking paragraphs for improving answer recall in open-domain question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 565–569, Brussels, Belgium. Association for Computational Linguistics.
- 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, Florence, Italy. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.
- Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.
- Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.
- Peng Qi, Haejun Lee, Oghenetegiri Sido, Christopher D Manning, et al. 2020. Retrieve, rerank, read, then iterate: Answering open-domain questions of arbitrary complexity from text. *arXiv preprint arXiv:2010.12527*.
- Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.
- Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018a. R³: Reinforced reader-ranker for open-domain question answering. *The AAAI Conference on Artificial Intelligence (AAAI)*.
- Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018b. Evidence aggregation for answer re-ranking in open-domain question answering. In *International Conference on Learning Representations*.
- Sohee Yang and Minjoon Seo. 2020. Is retriever merely an approximator of reader? *arXiv preprint arXiv:2010.10999*.