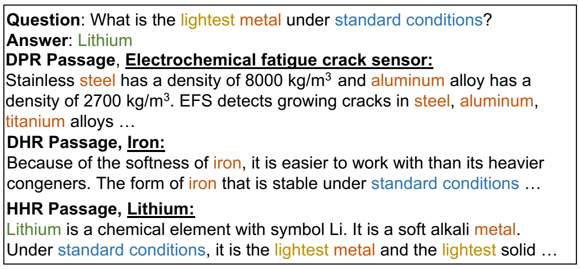
Hybrid Hierarchical Retrieval for Open-Domain Question Answering

Manoj Ghuhan Arivazhagan[+] Lan Liu[†] Peng Qi[†] Xinchi Chen[†]
William Yang Wang Zhiheng Huang
AWS AI Labs
{mghuhan, liuall, pengqi, xcc, wyw, zhiheng}@amazon.com

Abstract

Retrieval accuracy is crucial to the performance of open-domain question answering (ODQA) systems. Recent work has demonstrated that dense hierarchical retrieval (DHR), which retrieves document candidates first and then relevant passages from the refined document set, can significantly outperform the single stage dense passage retriever (DPR). While effective, this approach requires document structure information to learn document representation and is hard to adopt to other domains without this information. Additionally, the dense retrievers tend to generalize poorly on out-of-domain data comparing with sparse retrievers such as BM25. In this paper, we propose Hybrid Hierarchical Retrieval (HHR) to address the existing limitations. Instead of relying solely on dense retrievers, we can apply sparse retriever, dense retriever, and a combination of them in both stages of document and passage retrieval. We perform extensive experiments on ODQA benchmarks and observe that our framework not only brings in-domain gains, but also generalizes better to zero-shot TriviaQA and Web Questions datasets with an average of 4.69% improvement on recall@100 over DHR. We also offer practical insights to trade off between retrieval accuracy, latency, and storage cost. The code is available on github[*].



Question: What is the lightest metal under standard conditions?
Answer: Lithium
DPR Passage, **Electrochemical fatigue crack sensor:**
Stainless steel has a density of 8000 kg/m^3 and aluminum alloy has a density of 2700 kg/m^3. EFS detects growing cracks in steel, aluminum, titanium alloys ...
DHR Passage, Iron:
Because of the softness of iron, it is easier to work with than its heavier congeners. The form of iron that is stable under standard conditions ...
HHR Passage, Lithium:
Lithium is a chemical element with symbol Li. It is a soft alkali metal. Under standard conditions, it is the lightest metal and the lightest solid ...

Figure 1: An example showing the top-1 retrieved passage of DPR, DHR and HHR for a TriviaQA query. The title of the source document is underlined. DPR finds a passage with different metals from an irrelevant document. DHR retrieves a better passage yet focus a single iron metal. HHR finds the groundtruth passage.

1 Introduction

Open-domain question answering (ODQA) (Voorhees, 1999) aims to answer questions based on a large corpus without pre-specified context, and enjoys a broad scope of real-world applications such as chatbots, virtual assistants, search engines, etc. Recent ODQA systems often follow a two stage retrieve-then-read architecture (Zhu et al., 2021; Chen et al., 2017; Lee et al., 2019). Given a question, a *retriever* module first selects a candidate set of relevant contexts from a diversified large corpus such as Wikipedia; afterwards, a *reader* module consumes the retrieved evidence to predict an answer. Here, retrieval performance is crucial to the accuracy of the QA system as it determines whether the correct context to answer the question can be presented to the reader.

While most work in information retrieval focus on document retrieval (Nguyen et al., 2016; Thakur et al., 2021), existing work in ODQA often splits documents into short passages and directly retrieve passages for the reader (Karpukhin et al., 2020; Izacard and Grave, 2020) to accommodate reader models that handle shorter sequences most effectively. A drawback of such single-stage passage retrieval approaches is that they tend to be susceptible to distracting passages that contain seemingly relevant local context but not the correct answer, since they cannot incorporate information from other parts of the document (see Figure 1). Further, the large number of passage candidates also contributes negatively to system throughput. To mitigate these issues, Liu et al. (2021) recently proposed a two-stage hierarchical retrieval framework where the retriever first retrieves relevant documents, then discern relevant passages within re-

[+] The work was done during an internship at AWS AI Lab.
[†] These authors contributed equally to this work.
[*] https://github.com/ghuhan17/HybridHierarchicalRetrieval.

Findings of the Association for Computational Linguistics: ACL 2023, pages 10680–10689
July 9-14, 2023 ©2023 Association for Computational Linguistics

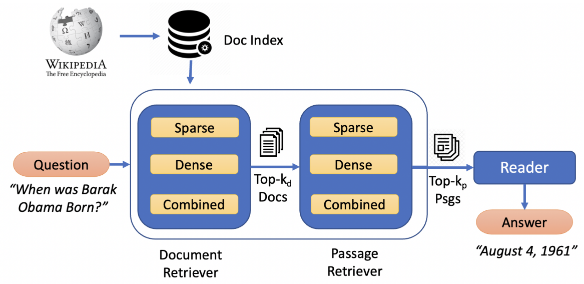


Figure 2: An overview of HHR. It consists of a document retrieval followed by a passage retrieval. Each stage uses one of three types of retrievers - sparse, dense, and combined, leading to a total of 9 configurations.

trieved documents. This helps prune passages that look relevant but are from irrelevant documents to improve answer accuracy, meanwhile greatly reducing the candidate set for passage retrieval and improve the inference speed of ODQA systems.

Despite its success, Liu et al.'s (2021) approach (dense hierarchical retrieval, DHR) relies on dense neural retrievers (Lee et al., 2019; Karpukhin et al., 2020) for both document retrieval and passage retrieval, which suffers from two key weaknesses. First, neural encoders used in retrieval are often limited in context length for effectiveness and efficiency, which is too short to encompass most documents. As a result, DHR needs to make use of the structure of Wikipedia documents, and represents the documents succinctly with title, abstract and table of contents, which is not always available for non-Wikipedia text. Second, dense retrievers have been shown to suffer from poor generalization on out-of-domain data (Thakur et al., 2021), whereas sparse retrievers like BM25 (Robertson et al., 2009) excel with lexical matches (Sciavolino et al., 2021).

In this work, we propose a hybrid hierarchical retrieval (HHR) framework to alleviate these issues. Specifically, we investigate the tradeoffs and complementary strengths of sparse retrievers and dense retrievers at both the document retrieval and passage retrieval stages for ODQA (see Figure 2). We find, among other things, that sparse retrievers can complement dense retrievers at both retrieval stages with a simple approach to aggregate results from both. Besides in-domain evaluation on the dataset that neural models are trained on, we also perform zero-shot evaluation on unseen datasets to compare the generalization of these retriever architectures. We find that sparse retrievers can help HHR generalize better to unseen data and potentially replace dense retrievers in document retrieval. In addition, we also study the accuracy, storage cost, and latency tradeoff for these architectures under the HHR framework, and offer practical insights to real-world ODQA systems that often need to factor these into consideration.

Our main contributions are as follows. First, we propose a hybrid hierarchical retrieval framework on ODQA, and extensively study tradeoffs and complementary strengths of sparse and dense retrievers in both document and passage retrieval. Second, we perform both in-domain and out-of-domain evaluation to provide insight into the generalization performance of different model choices. Finally, we present the accuracy-storage-latency landscape for HHR architectures and offer practical insights to real-world applications.

2 Background & Related Work

Open-domain question answering (ODQA). ODQA is a task that takes a question, such as *"Who got the first Nobel prize in physics?"*, and aims to find the answer from a large corpus. ODQA systems often rely on efficient and accurate retrievers to find relevant context to answer questions (Chen et al., 2017), where retrieval performance is usually critical to QA accuracy (Karpukhin et al., 2020).

Passage retrieval. Since most reader models in ODQA systems struggle to effectively handle long contexts, ODQA retrieval is often performed at the passage level (usually around 100 words long). Earlier work (Chen et al., 2017; Yang et al., 2019) relied on bag-of-words-based sparse retrievers such as BM25 (Robertson et al., 2009). More recent work showed that neural retrievers can generate effective dense representations for retrieval when trained on ODQA (Lee et al., 2019; Karpukhin et al., 2020; Liu et al., 2021). Sciavolino et al. (2021) showed, however, that these dense retrievers tend to generalize worse to unseen entities during training since they lack the capacity for lexical matching, which is a strong suit for sparse retrievers and important for out-of-domain generalization.

Hierarchical retrieval. Passage retrievers are limited by the context available in each passage, and can retrieve spurious passages to hurt answer performance. A remedy is to incorporate document-level relevancy during passage retrieval. Qi et al. (2021) explored combining document and passage relevancy scores in a BM25 retriever for ODQA. Liu et al. (2021) applied this idea to dense retrievers with a hierarchical retrieval framework (DHR), where a document retriever first retrieves documents of high relevancy, followed by a passage retriever to rerank passages within those documents, and our work extends this approach.

3 Methodology

Our hybrid hierarchical retrieval (HHR) framework extends DHR, a hierarchical retriever built on dense retrievers that first retrieves top-k_d documents and then top-k_p passages from those documents. We follow DHR to build the dense retrievers in HHR, and expand both document and passage retrievers to work with sparse retrievers to address limitations of the DHR approach (Figure 2). Specifically, in DHR, to make documents amenable to neural encoders with limited context length, the authors proposed to leverage the document structure of Wikipedia articles to construct a *document summary* that contains the document abstract and table of contents. While effective, this also potentially limits the applicability of this approach to corpora where this information is not available. In contrast, a sparse retriever can easily handle documents of arbitrary lengths efficiently without the need of structure information. Besides, dense retrievers tend to generalize poorly to out of domain data. We extend each of the document retrieval and passage retrieval stages with the option to use sparse retrievers to help alleviate this issue, and to help us understand the tradeoff between the two.

Besides switching between *sparse* and *dense* retrievers in HHR, we also introduce a simple heuristic to *combine* results from both retrievers at the same stage by simply interleaving their top-$k/2$ results for top-k retrieval to better understand the complementary strengths of sparse and dense retrievers. This yields a total of 9 possible configurations for HHR for our extensive studies. Finally, for both sparse and dense passage retrievers in HHR, we implement on-the-fly passage reranking for all passages in the top retrieved documents with pre-computed passage representations. This helps reduce the latency of the passage retrievers in our implementation and provide more realistic insights into the accuracy-storage-latency tradeoff of different HHR settings in real-world systems.

4 Experimental Setup

Data Three commonly used ODQA datasets are considered: Natural Questions (NQ) (Kwiatkowski et al., 2019), Web Questions (WebQ) (Berant et al., 2013), and TriviaQA (Joshi et al., 2017). Following Liu et al. (2021), we use the Wikipedia dump from Dec. 20, 2018 as the source corpus. When splitting documents into passages, the text under same sections are split into non-overlapping passages with a maximum of 100 words.

Training and evaluation We followed the same configuration as DHR to train the document and passage encoders up to 40 epochs on 8 V100 Tensor Core GPUs. In order to measure the retriever framework in both in-domain and zero-shot settings, the dense encoders are trained only on NQ dataset and tested on all three datasets.

5 Experiments and Results

5.1 Main Results

We present the in-domain and zero-shot evaluation results for all datasets in Table 1. Following DHR, we retrieve 100, 500, 500 documents in the first stage for NQ, WebQ, and TriviaQA, respectively. We compare HHR against DHR and single stage sparse and dense passage retrievers. In addition, we also study the effect of retrieving varying numbers of documents in HHR (see Figure 3). We find that:

First, **dense retrievers are crucial for in-domain performance, yet sparse retrievers bring gains with complementary strengths.** We see that replacing either or both components in the DHR baseline (*Dense+Dense*) with sparse retriever leads to noticeable drops in recall@100 on NQ, with 1.7% for replacing the document retriever, 2.3% for the passage retriever, and 7.2% for both. However, adding sparse retriever to document, passage or both retrievers brings 2.0%, 0.9% and 2.4% gain.

Second, **sparse retrievers significantly improve HHR's generalization on zero-shot datasets**. We corroborate previous work's (Sciavolino et al., 2021) finding that dense retrievers struggle to generalize in zero-shot settings. Likewise, DHR underperforms the optimal setting by 3.79% on WebQ, and leads to the worst performance on TriviaQA per recall@100. Adding sparse retrievers to both stages, *Combined+Combined* brings an average of 4.69% recall@100 improvement on WebQ and TriviaQA.

Third, **sparse document retriever can completely replace dense document retriever**. While this leads to performance drop in-domain, we see that dense passage retrievers can help make up for the performance gap, and that the gap diminishes

Retriever	NQ			WebQ			TriviaQA		
	R@20	R@100	EM	R@20	R@100	EM	R@20	R@100	EM
BM-25*	60.19	75.98	—	56.45	73.77	—	72.78	81.03	—
DPR* (DHR impl.)	82.87	89.16	42.4†	73.87	81.64	35.5†	80.43	85.40	56.9†
DHR*	85.07	89.92	43.6†	73.98	82.69	36.6†	80.81	85.69	57.0†
BM-25	61.19	76.68	32.83	56.45	73.77	17.52	75.13	82.98	41.17
DPR (DHR impl.)	82.35	88.92	41.77	69.19	79.63	19.29	70.72	80.42	37.91
Sparse+Sparse	65.21	81.11	36.04	60.04	76.87	18.75	75.38	83.17	41.77
Sparse+Dense	80.30	86.68	38.48	72.19	82.19	19.83	77.13	84.76	41.09
Sparse+Combined	79.75	86.07	38.50	72.10	78.64	19.83	**80.30**	85.76	42.54
Dense+Sparse	73.88	86.07	39.97	64.67	73.28	18.85	74.54	81.80	39.57
Dense+Dense (DHR)	83.05	88.34	41.52	69.98	79.92	18.95	70.86	79.72	37.89
Dense+Combined	83.93	89.22	41.66	71.95	80.71	19.64	76.96	82.72	39.93
Combined+Sparse	66.12	84.24	38.78	57.78	76.62	18.90	74.56	83.20	41.91
Combined+Dense	**84.40**	90.33	41.86	72.10	82.19	20.42	73.99	83.42	40.10
Combined+Combined	84.02	**90.75**	**41.97**	**72.29**	**82.97**	**20.62**	79.87	**86.04**	**43.14**

Table 1: Passage retrieval and end-to-end QA accuracy on test sets (R@k for recall at top-k passages and EM for answer exact match). BM25 and DPR are passage retriever baselines. * indicates referenced numbers from Liu et al. (2021) without iterative retriever training except for QA results (indicated with †). DHR demonstrates about 1% EM gain in QA performance for NQ and WebQ from applying iterative training to DPR. Grayed out values correspond to in-domain performance on WebQ and TriviaQA and are not directly comparable.

Retriever	Storage (GB)	Latency (s)
Sparse Document	5	0.310 / 0.389
Dense Document	16	0.131 / 0.135
Sparse Passage	23	0.004 / 0.018
Dense Passage	75	0.012 / 0.057

Table 2: Storage cost and retrieval latency on Wikipedia corpus. The two latency results for the document retrievers correspond to the time taken to retrieve 100 and 500 documents in the first stage respectively. The latency numbers for the passage retrievers show the time taken to retrieve 100 passages using a pool of 100 and 500 documents from the first stage respectively.

as more documents are retrieved in the first stage (Figure 3). Further, in the zero-shot setting, replacing dense document retrievers with sparse ones can actually lead to gains (of 1.26% and 3.15% recall@100 on WebQ and TriviaQA, respectively). This helps HHR generalize better to out-of-domain data not only lexically, but also removes the requirement of document structure information needed by the dense document retriever.

5.2 Accuracy-Storage-Latency Landscape

We report the trade off between retrieval accuracy, latency and storage cost for different HHR configurations based on inference on a single CPU core. Figure 4, presents the accuracy-storage-latency plot of different retriever configurations for the NQ, WebQ and TriviaQA datasets. We find that in the in-domain case of NQ, all settings involving dense document retrievers are Pareto-efficient on accuracy and latency, while the passage retriever presents a tradeoff among accuracy, latency, and storage. In contrast, in zero-shot settings the sparse document retrievers can be Pareto-efficient when complemented with dense passage retrievers. Table 2 presents the storage cost and retrieval latency for the four essential components in HHR framework, namely, sparse and dense retrievers in document and passage retriever stages. Sparse retrievers are storage efficient compared to the dense retrievers at the same level as they use inverted index to represent the text whereas dense retrievers use the dense embeddings. We also observe that the PyLucene's sparse document retriever is slower than the FAISS dense document retriever whereas our implementation of the on-the-fly sparse passage retriever is faster than its dense counterpart. Dense passage retrieval takes the most storage cost to store the embedding dictionary among others.

6 Conclusion

In this paper, we study a Hybrid Hierarchical Retrieval (HHR) framework for ODQA that integrates sparse and dense retrievers through a simple aggregation strategy in document and passage retrievers. We demonstrate that sparse retrievers complement dense retrievers in-domain and greatly improve the poor generalization from dense encoders to zero-shot data. Our framework addresses the limitation of DHR by achieving better zero-shot performance without relying on document structure information. We also study the accuracy-storage-latency land-

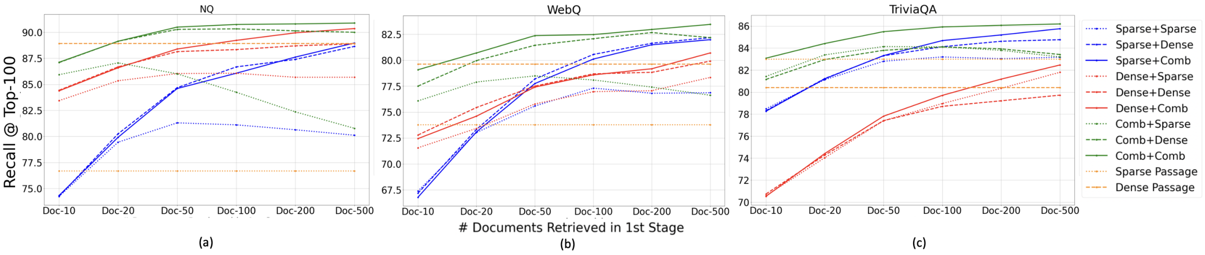


Figure 3: Effect of retrieving different number of documents in the document retriever stage for the (a) NQ (b) WebQ (c) TriviaQA datasets. For all the settings, we retrieve 100 passages in the second stage.

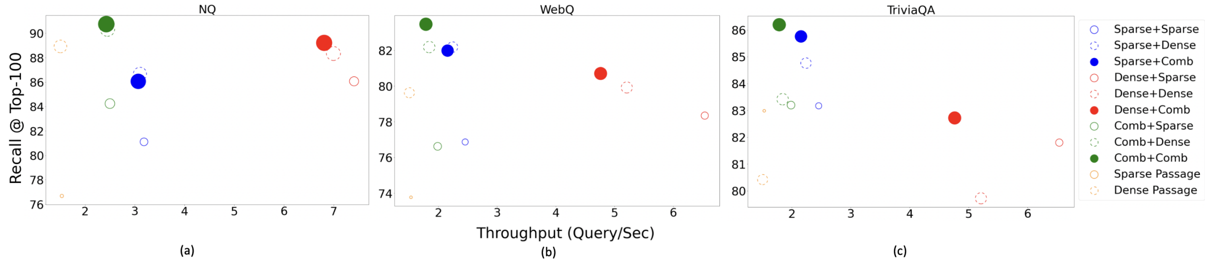


Figure 4: Accuracy-storage-latency landscape of different retriever settings for the (a) NQ (b) WebQ (c) TriviaQA datasets. The storage cost is indicated by the size of the markers. For all the settings, we retrieve 100, 500, 500 documents in the 1st stage for the NQ, WebQ and TriviaQA datasets respectively and 100 passages in the 2nd stage for all the datasets.

scape. We believe these findings are critical to the real-world adoption of HHR to ODQA systems.

7 Limitations

Our HHR framework uses a simple combination strategy to take top-$k/2$ documents (or passages) from dense and sparse retrievers for top-k retrieval in the combined setting. The overlap between dense and sparse results can lead to less than k retrieved results to be consumed by next stage. Future work might improve the aggregation strategy by: 1) developing a more advanced combination strategy that is not solely based on rank but also retrieval scores, and 2) accounting document retriever scores in the passage retrieval stage to rerank the passages according to both the global document relevancy and local passage relevancy. Futhermore, we only evaluate against Wikipedia ODQA datasets due to its presence of document structure information to be used in the dense document retriever. However, future work can extend the evaluation to other ODQA corpus.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. corr abs/1702.08734 (2017). *arXiv preprint arXiv:1702.08734*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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.

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.

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.

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

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.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models.

Ellen M Voorhees. 1999. The TREC-8 question answering track report. *TREC*, 99:77–82.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. pages 1–21.

A Implementation details of sparse and dense retrievers in HHR

Sparse Retrieval For the document level retriever, we used PyLucene[*] to build an inverted document index offline and perform BM25 based retrieval. Comparing with dense document retriever, it can run efficiently without document structure information. For the passage level retriever, the TF and IDF of all passage tokens are computed and pre-stored in a dictionary offline. During inference time, the BM25 score is computed on-the-fly. Due to the refined passage candidates, the passage retriever does not require construction of a huge passage index and can run faster than single-stage passage retrievers.

Dense Retrieval We follow the same dense retrieval set up as DHR with a small change. While DHR applies an iterative training strategy to perform a second step training from hard-negatives grounded from first step retriever, we train both the document and passage level retrievers without any iteration. While the document embeddings are indexed using FAISS (Johnson et al., 2017) to perform document retriever, the passage retrieval is performed on-the-fly with the cached passage embedding dictionary.

B Similarities and Differences between different HHR configurations

In this section, we analyze pairwise HHR configurations to understand their similarities and differences. Table 3 shows the percentage of queries in the NQ, WebQ and TriviaQA datasets respectively for which the correct evidences are retrieved exclusively by the row HHR configuration but not by the column HHR configuration. Higher percentage for a given pair of HHR configuration indicates that the row configuration is more exclusive and is able to answer a different set of queries when compared to the column configuration.

We can observe the following high-level takeaways: 1) Observing the *Sparse+Sparse* row in all the 3 tables indicate that Sparse retrievers are more effective and exclusively retrieve correct evidence for more queries in zero-shot settings as the percentages are more for WebQ and TriviaQA compared to NQ dataset. 2) Observing the *Combined+Combined* column shows that there are only few percentage of queries that

[*] https://lucene.apache.org/pylucene/

other HHR configurations are able to answer over *Combined+Combined*. This is expected as *Combined+Combined* leverages both sparse and dense retrievers in both the stages. Out of other HHR configurations, *Sparse+Dense* seems to have the highest percentage over *Combined+Combined*. Ensembling the results of these two configuration might result in an improved performance over *Combined+Combined* and can be an interesting future work.

Table 4 shows for a given pair of HHR configurations, the percentage of of queries, out of the total correctly retrieved queries by the row configuration, for which the column configuration also retrieves the correct evidence. Higher percentage indicates that the column HHR configuration is also able to correctly retrieve evidences for the queries as the row configuration. In case of NQ dataset, we observe that configurations that have dense retrievers in both the stages are able to correctly retrieve evidences for most of the queries of the other configurations. However, this percentage decreases in the case of WebQ and TriviaQA datasets.

	Configuration	Sparse+ Sparse	Sparse+ Dense	Sparse+ Comb	Dense+ Sparse	Dense+ Dense	Dense+ Comb	Comb+ Sparse	Comb+ Dense	Comb+ Comb	Average
NQ	Sparse+Sparse	0	1.2	0.8	3.5	3.5	2.6	1.2	1.6	0.7	1.7
	Sparse+Dense	9.3	0	1.5	6.6	3.7	3.3	3.9	1.6	1.4	3.5
	Sparse+Comb	8.1	0.8	0	6.3	3.7	3.2	3.2	1.4	0.9	3.1
	Dense+Sparse	9.4	4.4	4.9	0	1.6	0.5	3.3	1.2	0.9	2.9
	Dense+Dense	13.2	5.4	6	5.5	0	0.6	6.1	0.5	0.9	4.2
	Dense+Comb	13	5.7	6.2	5	1.2	0	6	1.2	0.8	4.3
	Comb+Sparse	8.4	3.1	3	4.7	3.6	2.8	0	1.7	0.6	3.1
	Comb+Dense	13.3	5.2	5.8	7.1	2.5	2.5	6.3	0	0.7	4.8
	Comb+Comb	13	5.7	5.9	7.3	3.5	2.8	5.7	1.4	0	5.0
WebQ	Sparse+Sparse	0	2.7	1.5	4.6	5.7	4.5	3	3.7	2.5	3.1
	Sparse+Dense	10.7	0	2.2	7.6	5.1	4.7	5.7	2.7	2.5	4.6
	Sparse+Comb	9.1	1.8	0	7.3	5.8	4.7	4.9	3.3	2.2	4.3
	Dense+Sparse	8	2.9	3.1	0	3.8	1.6	3.5	3.3	2	3.1
	Dense+Dense	11.5	2.9	4	6.3	0	1.7	6.3	0.7	1.9	3.9
	Dense+Comb	11.2	3.3	3.8	4.9	2.6	0	5.9	2.2	1.4	3.9
	Comb+Sparse	7.7	2.3	2	4.9	5.3	3.9	0	3.6	1.6	3.5
	Comb+Dense	11.8	2.7	3.8	8	3	3.6	7	0	1.8	4.6
	Comb+Comb	11.3	3.2	3.3	7.4	4.8	3.5	5.6	2.5	0	4.6
TriviaQA	Sparse+Sparse	0	2.4	0.8	5.2	7.8	4.8	1.7	4.3	1.6	3.2
	Sparse+Dense	4.1	0	1.1	5.4	6.8	4.6	3.2	3	1.7	3.3
	Sparse+Comb	3.3	2	0	6	7.8	5.2	2.8	4	1.7	3.6
	Dense+Sparse	3.6	2.1	1.8	0	4.4	1	2.4	2.7	1.3	2.1
	Dense+Dense	4.4	1.7	1.9	2.6	0	1	3.2	0.5	0.9	1.8
	Dense+Comb	4.2	2.3	2	2	3.7	0	3.2	2.4	0.9	2.3
	Comb+Sparse	2.4	2.2	0.9	4.7	7.3	4.5	0	4	1	3.0
	Comb+Dense	4.6	1.6	1.8	4.7	4.1	3.3	3.6	0	1.1	2.8
	Comb+Comb	4.2	2.6	1.7	5.5	6.9	4.2	2.9	3.4	0	3.5

Table 3: This table presents the percentage of queries within datasets, where only the row HHR configuration successfully retrieves correct evidence while the column HHR configuration fails, for a given pair of HHR configurations. For all the numbers shown, we retrieve 100, 500, 500 documents for NQ, WebQ and TriviaQA respectively in the first stage and 100 passages in the second stage for all the datasets.

	Configuration	Sparse+ Sparse	Sparse+ Dense	Sparse+ Comb	Dense+ Sparse	Dense+ Dense	Dense+ Comb	Comb+ Sparse	Comb+ Dense	Comb+ Comb	Average
NQ	Sparse+Sparse	100	98.5	99	95.5	95.6	96.7	98.4	98	99.2	97.9
	Sparse+Dense	89.3	100	98.2	92.4	95.7	96.2	95.5	98.2	98.4	96.0
	Sparse+Comb	90.5	99.1	100	92.6	95.7	96.3	96.3	98.3	99	96.4
	Dense+Sparse	88.9	94.8	94.2	100	98.1	99.4	96.1	98.5	99	96.6
	Dense+Dense	85.1	93.9	93.2	93.8	100	99.3	93.1	99.5	99	95.2
	Dense+Comb	85.4	93.7	93	94.3	98.6	100	93.3	98.7	99.1	95.1
	Comb+Sparse	90.2	96.4	96.4	94.6	95.8	96.7	100	98	99.4	96.4
	Comb+Dense	85.3	94.2	93.6	92.1	97.3	97.2	93.1	100	99.2	94.7
	Comb+Comb	85.7	93.8	93.5	91.9	96.2	97	93.7	98.5	100	94.5
WebQ	Sparse+Sparse	100	96.4	98	93.8	92.4	93.9	95.9	95	96.6	95.8
	Sparse+Dense	86.9	100	97.3	90.7	93.8	94.3	93.1	96.8	96.9	94.4
	Sparse+Comb	88.9	97.8	100	91	92.9	94.2	94	95.9	97.3	94.7
	Dense+Sparse	89.7	96.3	96.1	100	95.1	98	95.4	95.7	97.5	96.0
	Dense+Dense	85.7	96.4	95	92.2	100	97.9	92.1	99.1	97.7	95.1
	Dense+Comb	86.1	95.9	95.3	93.9	96.8	100	92.8	97.3	98.2	95.1
	Comb+Sparse	90.2	97.1	97.4	93.8	93.3	95.1	100	95.4	98	95.6
	Comb+Dense	85.6	96.8	95.4	90.2	96.4	95.6	91.5	100	97.8	94.4
	Comb+Comb	86.4	96.1	96	91.1	94.2	95.8	93.2	97	100	94.4
TriviaQA	Sparse+Sparse	100	97.1	99.1	93.8	90.7	94.2	97.9	94.9	98.1	96.2
	Sparse+Dense	95.2	100	98.7	93.6	92	94.5	96.2	96.5	98	96.1
	Sparse+Comb	96.1	97.6	100	93	90.9	94	96.7	95.3	98.1	95.7
	Dense+Sparse	95.6	97.4	97.8	100	94.6	98.8	97	96.6	98.4	97.4
	Dense+Dense	94.5	97.8	97.6	96.7	100	98.7	95.9	99.4	98.9	97.7
	Dense+Comb	94.9	97.2	97.6	97.6	95.5	100	96.2	97.1	98.9	97.2
	Comb+Sparse	97.2	97.4	98.9	94.4	91.3	94.7	100	95.3	98.9	96.5
	Comb+Dense	94.5	98.1	97.9	94.4	95	96	95.6	100	98.7	96.7
	Comb+Comb	95.1	96.9	98	93.6	92	95.1	96.6	96.1	100	95.9

Table 4: This table shows the percentage of queries, out of the total correctly retrieved queries datasets by the row configuration, for which the column configuration also retrieves correct evidence, for a given pair of HHR configurations. For all the numbers shown, we retrieve 100, 500, 500 documents for NQ, WebQ and TriviaQA respectively in the first stage and 100 passages in the second stage for all the datasets.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3, Section 4 and Section 5

☑ B1. Did you cite the creators of artifacts you used?
Section 3, Section 4 and Section 5

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We follow standard dataset splits

C ☑ Did you run computational experiments?

Section 4 and Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3, Section 4, and Section 5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4 and Appendix A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.