A Study on the Efficiency and Generalization of Light Hybrid Retrieversd

Man Luo [1*] **Shashank Jain**[2] **Anchit Gupta**[2†] **Arash Einolghozati**[2†]
Barlas Oguz[2†] **Debojeet Chatterjee**[2†] **Xilun Chen**[2†]
Chitta Baral[1] **Peyman Heidari** [2]
[1] Arizona State University
[2] Meta Reality Lab
[1] {mluo26, chitta}@asu.edu
[2]{shajain, anchit, arashe, barlaso, debo}@fb.com
[2]{xilun, peymanheidari}@fb.com

Abstract

Hybrid retrievers can take advantage of both sparse and dense retrievers. Previous hybrid retrievers leverage indexing-heavy dense retrievers. In this work, we study *"Is it possible to reduce the indexing memory of hybrid retrievers without sacrificing performance?"* Driven by this question, we leverage an indexing-efficient dense retriever (i.e. DrBoost) and introduce a LITE retriever that further reduces the memory of DrBoost. LITE is jointly trained on contrastive learning and knowledge distillation from DrBoost. Then, we integrate BM25, a sparse retriever, with either LITE or DrBoost to form light hybrid retrievers. Our Hybrid-LITE retriever saves 13× memory while maintaining 98.0% performance of the hybrid retriever of BM25 and DPR. In addition, we study the generalization capacity of our light hybrid retrievers on out-of-domain dataset and a set of adversarial attacks datasets. Experiments showcase that light hybrid retrievers achieve better generalization performance than individual sparse and dense retrievers. Nevertheless, our analysis shows that there is a large room to improve the robustness of retrievers, suggesting a new research direction.

1 Introduction

The classical IR methods, such as BM25 (Robertson et al., 2009), produce sparse vectors for question and documents based on bag-of-words approaches. Recent research pays attention toward building neural retrievers which learn dense embeddings of the query and document into a semantic space (Karpukhin et al., 2020; Khattab and Zaharia, 2020). Sparse and dense retrievers have their pros and cons, and the hybrid of sparse and dense retrievers can take advantage of both worlds and achieve better performance than individual sparse and dense retrievers. Therefore, hybrid retrievers are widely used in practice (Ma et al., 2021b; Chen et al., 2021).

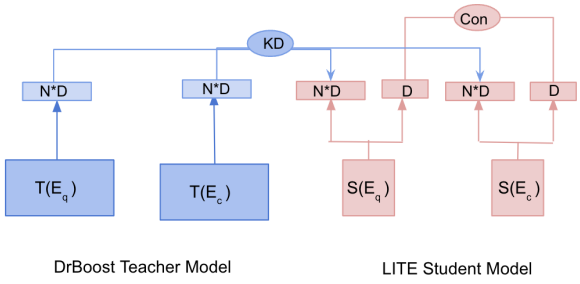

Figure 1: The teacher model (DrBoost) consists of N weak-learners and produces embeddings of dimension N*D. The student model (LITE) has one weak-learner and produces two embeddings: one has dimension of D, and one has dimension of N*D. The smaller embeddings learn to maximize the similarity between question and positive context embeddings, and the larger embeddings learn the embeddings from the teacher model.

Previous hybrid retrievers are composed of indexing-heavy dense retrievers (DR), in this work, we study the question *"Is it possible to reduce the indexing memory of hybrid retrievers without sacrificing performance?"* To answer this question, we reduce the memory by using the state-of-the-art indexing-efficient retriever, DrBoost (Lewis et al., 2021), a boosting retriever with multiple "weak" learners. Compared to DPR (Karpukhin et al., 2020), a representative DR, DrBoost reduces the indexing memory by 6 times while maintaining the performance. We introduce a LITE model that further reduces the memory of DrBoost, which is jointly trained on retrieval task via contrastive learning and knowledge distillation from DrBoost (see Figure 1). We then integrate BM25 with either LITE and DrBoost to form light hybrid retrievers (Hybrid-LITE and Hybrid-DrBoost) to assess whether light hybrid retrievers can achieve memory-efficiency and sufficient performance.

We conduct experiments on the NaturalQuestion dataset (Kwiatkowski et al., 2019) and draw interesting results. First of all, LITE retriever maintains

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1617–1626

98.7% of the teacher model performance and reduces its memory by 2 times. Second, our Hybrid-LITE saves more than 13× memory compared to Hybrid-DPR, while maintaining more than 98.0% performance; and Hybrid-DrBoost reduces the indexing memory (8×) compared to Hybrid-DPR and maintains at least 98.5% of the performance. This shows that the light hybrid model can achieve sufficient performance while reducing the indexing memory significantly, which suggests the practical usage of light retrievers for memory-limited applications, such as on-devices.

One important reason for using hybrid retrievers in real-world applications is the generalization. Thus, we further study if reducing the indexing memory will hamper the generalization of light hybrid retrievers. Two prominent ideas have emerged to test generalization: out-of-domain (OOD) generalization and adversarial robustness (Gokhale et al., 2022). We study OOD generalization of retrievers on EntityQuestion (Sciavolino et al., 2021). To study the robustness, we leverage six techniques (Morris et al., 2020) to create adversarial attack testing sets based on NQ dataset. Our experiments demonstrate that Hybrid-LITE and Hybrid-DrBoost achieve better generalization performance than individual components. The study of robustness shows that hybrid retrievers are always better than sparse and dense retrievers. Nevertheless all retrievers are vulnerable, suggesting room for improving the robustness of retrievers, and our datasets can aid the future research.

2 Related Work

Hybrid Retriever integrates the sparse and dense retriever and ranks the documents by interpolating the relevance score from each retriever. The most popular way to obtain the hybrid ranking is applying linear combination of the sparse/dense retriever scores (Karpukhin et al., 2020; Ma et al., 2020; Luan et al., 2021; Ma et al., 2021a; Luo et al., 2022). Instead of using the scores, Chen et al. (2022) adopts Reciprocal Rank Fusion (Cormack et al., 2009) to obtain the final ranking by the ranking positions of each candidate retrieved by individual retriever. Arabzadeh et al. (2021) trains a classification model to select one of the retrieval strategies: sparse, dense or hybrid model. Most of the hybrid models rely on heavy dense retrievers, and one exception is (Ma et al., 2021a), where they use linear projection, PCA, and product quantization (Jegou et al., 2010) to compress the dense retriever component. Our hybrid retrievers use either DrBoost or our proposed LITE as the dense retrievers, which are more memory-efficient and achieve better performance than the methods used in (Ma et al., 2021a).

Indexing-Efficient Dense Retriever. Efficiency includes two dimensions: latency (Seo et al., 2019; Lee et al., 2021; Varshney et al., 2022) and memory. In this work, our primary focus is on memory, specifically the memory used for indexing. Most of the existing DRs are indexing heavy (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Luo, 2022). To improve the indexing efficiency, there are mainly three types of techniques. One is to use vector product quantization (Jegou et al., 2010). Second is to compress a high dimension dense vector to a low dimension dense vector, for e.g. from 768 to 32 dimension (Lewis et al., 2021; Ma et al., 2021a). The third way is to use a binary vector (Yamada et al., 2021; Zhan et al., 2021). Our proposed method LITE (§3.2) reduces the indexing memory by joint training of retrieval task and knowledge distillation from a teacher model.

Generalization of IR. Two main benchmarks have been proposed to study the OOD generalization of retrievers, BEIR (Thakur et al., 2021b) and EntityQuestion (Sciavolino et al., 2021). As shown by previous work (Thakur et al., 2021b; Chen et al., 2022), the generalization is one major concern of DR. To address this limitation, Wang et al. (2021) proposed GPL, a domain adaptation technique to generate synthetic question-answer pairs in specific domains. A follow-up work Thakur et al. (2022) trains BPR and JPQ on the GPL synthetic data to achieve efficiency and generalization. Chen et al. (2022) investigates a hybrid model in the OOD setting, yet different from us, they use a heavy DR and do not concern the indexing memory. Most existing work studies OOD generalization, and much less attention paid toward the robustness of retrievers (Penha et al., 2022; Zhuang and Zuccon, 2022; Chen et al.). To study robustness, Penha et al. (2022) identifies four ways to change the syntax of the queries but not the semantics. Our work is a complementary to Penha et al. (2022), where we leverage adversarial attack techniques (Morris et al., 2020) to create six different testing sets for NQ dataset (Kwiatkowski et al., 2019).

3 Model

In this section, we first review DrBoost (Lewis et al., 2021), and our model LITE which further reduces the memory of DrBoost, and lastly, we describe the hybrid retrievers that integrate light dense retrievers (i.e. LITE and DrBoost) and BM25.

3.1 Reivew of DrBoost

DrBoost is based on ensemble learning to form a strong learner by a sequence of weak leaners, and each weak learner is trained to minimize the mistakes of the combination of the previous learners. The weak learner has the similar architecture as DPR (Karpukhin et al., 2020) (review of DPR is given in Appendix A), but the output vectors are compressed to a much lower dimension by a linear regression layer W,

$$\mathrm{v}_q^i = \mathrm{W}_q \cdot \mathrm{V}_q^i, \quad \mathrm{v}_c^i = \mathrm{W}_c \cdot \mathrm{V}_c^i, \tag{1}$$

where $\mathrm{V}_{q/c}^i$ are the representation of question/document given by the embeddings of special tokens `[CLS]` of a high dimension, $\mathrm{v}_{q/c}^i$ are the lower embeddings produced by the i^{th} weak learner. The final output representation of DrBoost is the concatenation of each weak learners' representations as expressed by Eq. 2.

$$\mathbf{q} = [\mathrm{v}_q^1, \ldots, \mathrm{v}_q^n], \quad \mathbf{c} = [\mathrm{v}_c^1, \ldots, \mathrm{v}_c^n], \tag{2}$$

where n is the total number of weak learners in the DrBoost. The training objective of DrBoost is

$$\mathcal{L}_{con} = -\log \frac{e^{\mathrm{sim}(q,c^+)}}{e^{\mathrm{sim}(q,c^+)} + \sum_{j=1}^{j=n} e^{\mathrm{sim}(q,c_j^-)}}, \tag{3}$$

where $\mathrm{sim}(q, c)$ is the inner-dot product.

3.2 LITE: Joint Training with Knowledge Distillation

Since DrBoost has N encoders, the computation of query representations takes N times as a single encoder. To save latency, Lewis et al. (2021) trains a student encoder which learns the N embeddings from the teacher encoders. As a result, while the student model consists of only one encoder, it produces the same indexing memory as the teacher model. Here, we want to further reduce the student indexing memory. To achieve this, we introduce a LITE retriever (see Figure 1), which produces two embeddings for an input text: one has a smaller dimension ($\mathrm{v}_{q/c,s}$) for retrieval task, and the other one is a larger dimension ($\mathrm{v}_{q/c,l}$) for learning knowledge from the N teacher models. The small and large embeddings are obtained by compressing the `[CLS]` token embedding via separate linear regression layers, mathematically,

$$\mathrm{v}_{q/c,s} = \mathrm{W}_{q/c,s} \cdot \mathrm{V}_{q/c}, \quad \mathrm{v}_{q/c,l} = \mathrm{W}_{q/c,l} \cdot \mathrm{V}_{q/c} \tag{4}$$

$\mathrm{v}_{q/c,s}$ is optimized by the contrastive loss (E.q. 3). And $\mathrm{v}_{q/c,l}$ learns the teacher model embeddings. The knowledge distillation (KD) loss is composed of three parts (Eq. 5): 1) the distance between student question embeddings and the teacher question embeddings, 2) the distance between student context embeddings and the teacher context embeddings, and 3) the distance between student question embeddings and the teacher positive context embeddings.

$$\mathcal{L}_{KD} = \|\mathrm{v}_{q,l} - \mathbf{q}\|^2 + \|\mathrm{v}_{c,l} - \mathbf{c}\|^2 + \|\mathrm{v}_{q,l} - \mathbf{c}^+\|^2 \tag{5}$$

The final objective of the student model is,

$$\mathcal{L}_{joint} = \mathcal{L}_{con} + \mathcal{L}_{KD}. \tag{6}$$

In contrast to the distillation method in DrBoost, which solely learns the embeddings from the teacher model, LITE is simultaneously trained on both the retrieval task and the knowledge distillation task. During the inference time, LITE only utilizes the retrieval embeddings ($\mathrm{v}_{c,s}$) to achieve indexing-efficiency. It is also notable that LITE is a flexible training framework capable of incorporating most neural retrievers as its backbone models, despite our work being solely reliant on DrBoost.

3.3 Memory Efficient Hybrid Model

Our hybrid models retrieve the final documents in a re-ranking manner. We first retrieve the top-k documents using BM25 and dense retriever (DrBoost or LITE) separately. The document scores produced by these two retrievers are denoted by S_{BM25} and S_{DR} respectively. We apply MinMax normalization to original socres to obtain S'_{BM25} and S'_{DR} ranging from $[0, 1]$. For each document, we get a new score for final ranking:

$$S_{\mathrm{hybrid}} = w_1 \times S'_{\mathrm{BM25}} + w_2 \times S'_{\mathrm{DR}}, \tag{7}$$

where w_1 and w_2 denote the weights of BM25 and DrBoost scores respectively. In our experiments, we simply set equal weights (i.e. 0.5) to each method. If a context is not retrieved by either retriever, then its score for that retriever is 0.

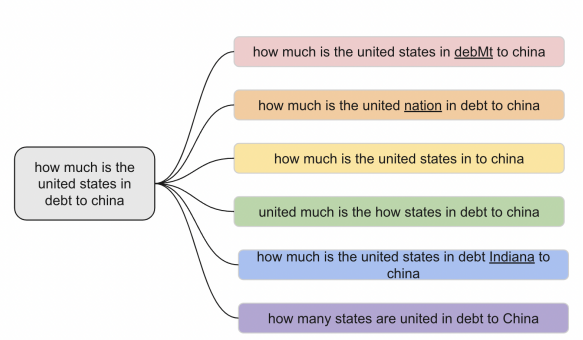

Figure 2: Examples of the adversarial attack questions. Underline denotes the change from the original question. The examples from the top to the bottom are augmented by CS, WD, SR, WOS, SI, and BT.

4 Adversarial Attack Robustness Dataset

Adversarial attacks are used to asses model's robustness, where testing samples are obtained by small perturbations of the original samples, and such perturbations keep the label unchanged. To test the robustness of IR systems, we create 6 different adversarial attacks[1] for NQ (Kwiatkowski et al., 2019). Each method is chosen because they do not change the original meaning of the queries and the relevant documents should be the same as the original relevant documents (see Figure 2). The six methods include: *Char-Swap (CS):* augments words by swapping characters out for other characters; *Word Deletion (WD):* delete a word randomly from the original query; *Synonym Replacement (SR):* replaces a word in the query with a synonym from the WordNet (Miller, 1995); *Word-Order-Swap (WOS):* swaps the order of the words in the original query; *Synonym Insertion (SI):* insert a synonym of a word from the WordNet to the original query; *Back-Translation (BT)* translates the original query into a target language and translates it back to the source language. Figure 2 shows an example of each attacked instance[2].

5 Experiments and Results

Existing Methods. We include four existing methods in this work, DrBoost (Lewis et al., 2021), DPR (Karpukhin et al., 2020), SPAR (Chen et al., 2021) and a heavy hybrid model BM25 + DPR (Karpukhin et al., 2020). In Table 1, the performance of DrBoost is from the original paper and the performance of the other three methods are from (Chen et al., 2021).

Our Baselines. Three baselines are presented, BM25, DPR_{32}, and DrBoost-2. DPR_{32} refers to DPR with a linear projection layer to representation to 32 dimension. DrBoost-2 takes DPR_{32} as the first weak learner, and uses it to mine negative passages to train the next weak learner and then combine these two models. We do not go beyond 2 weak learners because our goal is to achieve memory-efficiency while increasing the number of encoders in the DrBoost will yield larger indexing.

Our Models. LITE and the three light hybrid models are presented. LITE is trained by the method we introduce in §3.2 with the distilled knowledge from DrBoost-2 teacher model. We present three hybrid models BM25 + LITE, BM25 + DPR_{32}, and BM25 + DrBoost-2, which are memory-efficient compared to existing methods. Next we present the experiments and the findings.

5.1 Memory Efficiency and Performance

LITE achieves much better performance compared to DPR_{32} even though both use the same amount of memory. LITE also maintains more than 98% knowledge of its teacher (DrBoost-2), and importantly saves 2× of indexing memory. Such results shows the effectiveness of LITE.

Hybrid-LITE achieves better performance than DrBoost-2 while using less indexing memory. Hybrid-LITE also matches the performance of DrBoost in terms of R@100 (87.4 v.s. 87.2) while using 3× less memory. Compared with Hybrid-DPR, Hybrid-LITE maintains 98.4% performance but uses 13× less memory. Compared with the SOTA model SPAR, Hybrid-LITE achieves 98.2% performance and uses 25× less memory.

Hybrid-DrBoost-2 achieves almost similar performance as DrBoost which contains 6 encoders. This shows the effects of BM25 match the capacity of 4 encoders in the DrBoost. We also compare Hybrid-DrBoost-2 with BM25 + DRP or SPAR, where our model achieves almost 99% performance but uses less than 8× or 16× of memory.

5.2 Out-of-Domain Generalization

We study the out-of-domain generalization of retriever on EntityQuestion (Sciavolino et al., 2021), which consists of simple entity centric questions but shown to be difficult for dense retrievers. We train the model on NQ and test on EQ.

[1]We use TextAttack library (Morris et al., 2020).
[2]The adversarial robustness dataset is available in this link.

Method	Index-M (GB)	NQ		EntityQuestion	
		R@20	R@100	R@20	R@100
Existing Method					
DrBoost	15.4/13.5	81.3	87.4	51.2	63.4
DPR	61.5	79.5	86.1	56.6	70.1
BPR	2	77.9	85.7	-	-
BM25+DPR	63.9	82.6	88.6	73.3	82.3
SPAR	123.0	83.6	88.8	74.0	82.0
Our Baseline					
BM25	2.4	63.9	78.8	71.2	79.7
DPR_{32}	2.5	70.4	80.0	31.1	45.5
DrBoost-2	5.1	77.3	84.5	41.3	54.2
Our Model					
LITE	2.5	75.1	83.4	35.0	48.1
Hybrid-LITE	4.9	79.9	87.2	71.5	80.8
Hybrid-DPR_{32}	4.9	77.7	86.2	70.8	80.5
Hybrid-DrBoost-2	7.5	80.4	87.5	72.4	81.4

Table 1: Performance of existing methods, our baselines and our hybrid model on NQ dataset. The performance of DrBoost on NQ is using 6 weak learners (15.4 GB indexing memory) and of EntityQuestion is using 5 weak learners (13.5 GB).

First of all, our experimental results show that the performance of DPR_{32}, DrBoost-2, and LITE are much worse than BM25 on EQ. Nevertheless, our hybrid models improve both BM25 and dense retriever performance. Our light hybrid models achieve similar performance as hybrid-DPR and SPAR, which demonstrates that our light hybrid retrievers exhibit good OOD generalization.

5.3 Adversarial Attack Robustness

The robustness is evaluated in terms of both performance (higher R@K means more robust) and the average drop w.r.t the original performance on NQ dataset (smaller drop means more robust).

From Table 2, we observe that all models perform worse compared to the original performance on all adversarial attack sets, which showcase that the current retrievers are not robust enough. Interestingly, while it is expected that BM25 will be robust on word-order-swap (WOS) attack, it is not straightforward that a dense retriever is also robust on this type of questions. This shows that the order of the words in the question is not important for the dense retriever neither. We also see that char-swap (CS) is the most difficult attack, which means that both types of retrievers might not perform well when there are typos in the questions.

Diving into the individual performance of each retriever, we see that some models are more robust than others. For example, LITE is more robust than DPR_{32}. We also compare the hybrid model with the pure dense retriever counterparts (e.g. compare hybrid Drboost-2 with DrBoost-2), and find that hybrid models are consistently more robust. This suggests that the hybrid model can mitigate the performance drop of both BM25 and dense retriever.

Method	R@100							
	Ori	CS	WD	SR	WOS	SI	BT	Drop
BM25	78.8	68.2	71.7	74.5	78.3	77.2	71.2	5.9
DPR_{32}	80.8	61.9	65.8	75.3	76.4	73.3	71.1	10.3
LITE	83.4	69.3	71.8	78.9	81.2	79.0	75.6	7.9
DrBoost-2	84.5	71.6	80.1	74.7	82.6	80.4	77.9	7.8
DPR_{768}	86.1	74.8	78.9	82.5	85.0	83.4	80.3	5.5
+DPR_{32}	86.2	74.4	78.0	82.7	84.9	83.2	78.6	6.1
+LITE	87.2	76.5	78.0	83.7	86.6	85.4	80.8	5.1
+DrBoost-2	87.5	77.7	**84.6**	81.0	86.7	85.9	81.9	5.2
+DPR_{768}	**88.3**	**78.6**	82.9	**85.4**	**87.7**	**86.6**	**82.6**	**4.4**

Table 2: Ori: Original question; CS: CharSwap; WD:Word deletion; WSR: WordNet synonym replacement; WOR: Word order swaps; RSI :Random synonym insertion; BT: Back Translation. The smaller the Average Drop is, the more robust the model is.

6 Conclusion

To achieve indexing efficiency, in this work, we study light hybrid retrievers. We introduce LITE, which is jointly trained on retrieval task via contrastive learning and knowledge distillation from a more capable teacher models which requires heavier indexing-memory. While in this work, we mainly take DrBoost as the teacher model, LITE is a flexible training framework that can be incorporated with most of the neural retriever. Then, we integrate BM25 with LITE or DrBoost to form light hybrid retrievers. Our light hybrid models achieve sufficient performance and largely reduce the memory. We also study the generalization of retrievers and suggest that all sparse, dense, and hybrid retrievers are not robust enough, which opens up a new avenue for research.

Limitation

The main limitation of this work is the technical novelty of hybrid retriever. Hyrbid-DrBoost is built on top of DrBoost, and the interpolation of BM25 with DrBoost. However, we would like to point out that our study can serve as an important finding for real-life applications. Previous retrievers are built on top of indexing-heavy dense retrievers, such as DPR. This limits their applications where memory is a hard constraints, for example, on-devices. Our study suggests that a light hybrid retriever can save memory but maintain sufficient performance.

References

Negar Arabzadeh, Xinyi Yan, and Charles LA Clarke. 2021. Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2862–2866.

Tao Chen, Mingyang Zhang, Jing Lu, Michael Bendersky, and Marc Najork. 2022. Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models. In *European Conference on Information Retrieval*, pages 95–110. Springer.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *arXiv preprint arXiv:2110.06918*.

Xuanang Chen, Jian Luo, Ben He, Le Sun, and Yingfei Sun. Towards robust dense retrieval via local ranking alignment.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Tejas Gokhale, Swaroop Mishra, Man Luo, Bhavdeep Sachdeva, and Chitta Baral. 2022. Generalized but not robust? comparing the effects of data modification methods on out-of-domain generalization and adversarial robustness. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2705–2718.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128.

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.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647.

Patrick Lewis, Barlas Oğuz, Wenhan Xiong, Fabio Petroni, Wen-tau Yih, and Sebastian Riedel. 2021. Boosted dense retriever. *arXiv preprint arXiv:2112.07771*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Man Luo. 2022. Neural retriever and go beyond: A thesis proposal. *arXiv preprint arXiv:2205.16005*.

Man Luo, Arindam Mitra, Tejas Gokhale, and Chitta Baral. 2022. Improving biomedical information retrieval with neural retrievers.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2020. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. *arXiv preprint arXiv:2004.14503*.

Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. 2021a. Simple and effective unsupervised redundancy elimination to compress dense vectors for passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2854–2859.

Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. 2021b. A replication study of dense passage retriever. *arXiv preprint arXiv:2104.05740*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the robustness of retrieval pipelines with query variation generators. In *European Conference on Information Retrieval*, pages 397–412. Springer.

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.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441.

Nandan Thakur, N. Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. 2021a. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663.

Nandan Thakur, Nils Reimers, and Jimmy Lin. 2022. Domain adaptation for memory-efficient dense retrieval. *arXiv preprint arXiv:2205.11498*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021b. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Neeraj Varshney, Man Luo, and Chitta Baral. 2022. Can open-domain qa reader utilize external knowledge efficiently like humans? *arXiv preprint arXiv:2211.12707*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577*.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly optimizing query encoder and product quantization to improve retrieval performance. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2487–2496.

Shengyao Zhuang and Guido Zuccon. 2022. Characterbert and self-teaching for improving the robustness of dense retrievers on queries with typos. *arXiv preprint arXiv:2204.00716*.

A Preliminary

BM25 Robertson et al. (2009), is a bag-of-words ranking function that scores the query (Q) and document (D) based on the term frequency. The following equation is the one of the most prominent instantiations of the function,

$$score(D,Q) = \sum_{i=1}^{n} \mathrm{IDF}(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k1\cdot(1-b+b\cdot\frac{|D|}{avgdl})}, \tag{8}$$

where $\mathrm{IDF}(q_i)$ is the inverse document frequency of query term q_i, $f(q_i, D)$ is the frequency of q_i in document D, $|D|$ is the length of the document D, and $avgdl$ is the average length of all documents in the corpus. In practice, $k_1 \in [1.2, 2.0]$ and $b = 0.75$. BM25 is an unsupervised method that generalizes well in different domains (Thakur et al., 2021a).

DPR Dense passage retriever involves two encoders: the question encoder E_q produces a dense vector representation V_q for an input question q, and the context encoder E_c produces a dense vector V_c representation for an input context c. Both encoders are BERT models and the output vectors are the embeddings of the special token `[CLS]` in front of the input text (Eq. 9).

$$\mathrm{V}_q = \mathrm{E}_q(q)\texttt{[CLS]}, \quad \mathrm{V}_c = \mathrm{E}_c(c)\texttt{[CLS]}. \tag{9}$$

The score of c w.r.t q is the inner-dot product of their representations (Eq 10).

$$\mathrm{sim}(q,c) = \mathrm{V}_q^\top \mathrm{V}_c. \tag{10}$$

DPR uses contrastive loss to optimize the model such that the score of positive context c^+ is higher than the score of the negative context c^-. Mathematically, DPR maximizes the following objective function,

$$\mathcal{L}_{con} = -\log \frac{e^{\mathrm{sim}(q,c^+)}}{e^{\mathrm{sim}(q,c^+)} + \sum_{j=1}^{j=n} e^{\mathrm{sim}(q,c_j^-)}}, \tag{11}$$

where n is the number of negative contexts. For better representation learning, DPR uses BM25 to mine the hard negative context and the in-batch negative context to train the model.

Metric	O-DrBoost	R-DrBoost	LITE-DrBoost	H-LITE-DrBoost
R@20	77.3	75.6	77.9	**81.0**
R@100	84.5	83.9	84.7	**87.5**

Table 3: Three DrBoost (with 2 weak learners) and one hybrid retriever. O-DrBoost: the original DrBoost, R-DrBoost:replace the first weak learner in O-DrBoost with LITE, LITE-DrBoost: use LITE as the first weak learner and mine negative using LITE to train a new weak learner to form a DrBoost, H-LITE-DrBoost: hybrid BM25 with LITE-DrBoost.

B Ablation Study

In this section, we conduct ablation studies to see the effects of the proposed methods, and all models are trained and tested on NQ dataset.

B.1 LITE Can Improve DrBoost

Recall that DPR_{32} is one encoder in DrBoost-2, and since LITE performs better than DPR_{32} (see Table 1), we ask the question can LITE replaces DPR_{32} to form a stronger DrBoost-2 model? To answer this question, we compare the performance of R-DrBoost-2 (i.e. replace DPR_{32} with LITE) with the original DrBoost-2. From Table 3, We observe that R-DrBoost-2 performs worse than DrBoost-2, indicating that the encoders in the DrBoost indeed relate and complement to each other and replacing an unrelated encoder degrades the performance. Then we ask another question, can we train a weak learner that minimizes the error of LITE, and combine LITE with the new weak learner to form a stronger DrBoost (L-DrBoost-2)? Table 3 shows L-DrBoost-2 is better than DrBoost-2, and hybrid L-DrBoost-2 is better than hybrid DrBoost-2 as well (81.0 v.s. 80.4 on R@20). This indicates that starting with a stronger weak learner can yield a stronger DrBoost.

B.2 Hybrid model consistently improves the DrBoost performance.

We study six DrBoost models with 1-6 weak learners. In Figure 3, we see that the performance of hybrid models consistently improves the DrBoost performance, demonstrating the results of BM25 and DrBoost complement each other and combining two models improves individual performance. We also see that the improvement is larger when the DrBoost is weaker, e.g. hybrid model significantly improves DPR_{32}.

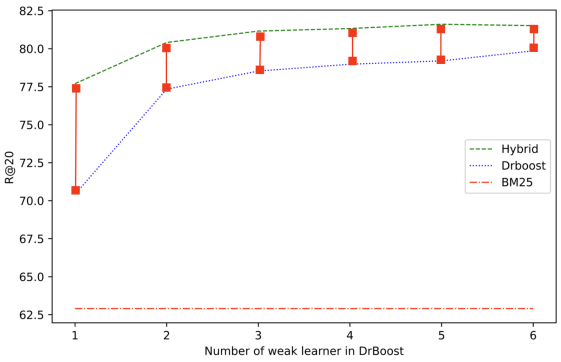

Figure 3: Compare DrBoost, BM25 and the Hybrid models performance.

Model	Method	NQ	
		R20	R100
Hybrid(32*2)	Simple Sum	79.03	84.63
	Multiplication	79.03	84.63
	MinMax and Sum	**80.41**	**87.47**
Hybrid(32*6)	Simple Sum	**81.61**	86.12
	Multiplication	81.19	86.12
	MinMax and Sum	81.52	**88.28**

Table 4: Compare three hybrid scores. We study two hybrid model, BM25 with 2 weak learners (32*2) and BM25 with 6 weak learners (32*6)

B.3 Different Hybrid Scores

In our hybrid model, besides the hybrid scores we introduced in §3.3, we also study two different hybrid scores of BM25 and the DrBoost. Simple Summation is to add two scores together, and multiplication is to mutiply two scores. We compare two hybrid models' performance, Hybrid-DrBoost-2 and Hybrid-DrBoost-6. Table 4 shows that the MinMax normalization performs the best (except that simple summation is slightly better in terms of R@20 for hybrid models with 6 weak learners).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

☐ 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?
Left blank.

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

B ☑ Did you use or create scientific artifacts?

Section 2

☑ B1. Did you cite the creators of artifacts you used?
section 2 and 3.

☐ 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)?
Section 3.

☐ 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.?
Left blank.

☑ 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.
Left blank.

C ☑ Did you run computational experiments?

Left blank.

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

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?
Left blank.

☐ 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?
Not applicable. Left blank.

☑ 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.)?
Left blank.

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.?
Not applicable. Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.