HYBRID AND COLLABORATIVE PASSAGE RERANKING

Anonymous authors
Paper under double-blind review

ABSTRACT

In information retrieval system, the initial passage retrieval results may be unsatisfactory, which can be refined by a reranking scheme. Existing solutions to passage reranking focus on enriching the interaction between query and each passage separately, neglecting the context among the top-ranked passages in the initial retrieval list. To tackle this problem, we propose a ***Hyb**rid and Collaborative Passage Re**rank**ing* (**HybRank**) method, which leverages the substantial similarity measurements of upstream retrievers for passage collaboration and incorporates the lexical and semantic properties of sparse and dense retrievers for reranking. Besides, built on off-the-shelf retriever features, the flexible plug-in HybRank is capable of enhancing arbitrary passage list even from other rerankers. Extensive experiments demonstrate the stable improvements of performance over prevalent retrieval and reranking methods, and verify the effectiveness of the core components of HybRank.[1]

1 INTRODUCTION

Information retrieval is a fundamental component within the field of natural language processing (Chen et al., 2017). Retrieval aims to search a set of candidate documents from a large-scale corpus, and thus high recall retrieval with efficiency is required to cover more relevant documents as far as possible. Traditionally, retrieval has been dominated by sparse methods like TF-IDF and BM25 (Robertson & Zaragoza, 2009), which treat queries and documents as sparse bag-of-words vectors and match them in token-level. Recently, neural networks have become prevalent to deal with information retrieval, where queries and documents are encoded into dense contextualized vectors (Huang et al., 2020; Karpukhin et al., 2020; Ren et al., 2021a; Zhang et al., 2022), and then retrieval is performed with highly optimized vector search algorithms (Johnson et al., 2021).

Although numerous efforts have been dedicated to retrieval, the inherent efficiency requirement restrict the interaction between query and passage to a shallow level, leading to unsatisfactory retrieval results. Thus, in typical reranking (Nogueira & Cho, 2020; Sun et al., 2021), query and passage are concatenated and fed into a Transformer (Vaswani et al., 2017) pre-trained on large corpus, to estimate a more fine-grained relevance score and further enhance the retrieval results with richer interaction. These methods consider each passage in isolation, ignoring the context of the retrieved passage list. Some learning to rank (Rahimi et al., 2016; Xia et al., 2008) and pseudo-relevance feedback (Zamani et al., 2016; Zhai & Lafferty, 2001) methods utilize the ordinal relationship or listwise context of retrieved documents to further refine the retrieval. Moreover, the necessity of integrating listwise context is confirmed in multi-stage recommendation systems (Liu et al., 2022).

Inspired by the success of listwise modeling and collaborative filtering (Goldberg et al., 1992) in recommendation systems, we find that collaboration also exists among the passages in the retrieval list and has not been fully exploited. Intuitively, for a specific query, relevant passages tend to describe the same entities, events and relations (Lee et al., 2019), while the irrelevant ones involve multifarious objects. Therefore, a passage is more likely to be relevant with the query if most of other passages share similar content with it. Similarities between passages can be naturally derived from retrievers, like BM25 scores in sparse retrievers and dot product in dense retrievers.

In addition, the sparse and dense retrieval methods emphasize distinct linguistic aspects. Sparse retrieval relies on lexical overlap while dense retrieval focuses on semantic and contextual relevance.

[1]We will put an anonymous link to our code in the discussion forum, and will release our code once this work is accepted.

Several researchers have attempted to integrate the merits of these two types of methods. Karpukhin et al. (2020), Lin et al. (2020) and Luan et al. (2021) exploit the linear combination of these two types of retrieval scores. Seo et al. (2019), Khattab & Zaharia (2020) and Santhanam et al. (2022) index smaller units in sentence, *i.e.*, words or phrases, to obtain fine-grained similarity. Gao et al. (2021a) and Yang et al. (2021) retrains dense retriever from scratch with the supervision of sparse signals. Nevertheless, the linear score combination lacks sufficient interaction, indexing smaller units sacrifices retrieval efficiency due to tremendous amount of embeddings, while rebuilding of retrievers discards their origin ranking capability.

To fully exploit the context of retrieved passages list and explore more sufficient ensemble of heterogeneous retriever, we propose a ***Hyb****rid and Collaborative Passage* ***Rerank****ing* (HybRank) method, which leverages the collaboration within retrieved passages and incorporates diverse properties of retrievers for reranking. Our method is a flexible plug-in reranker ready to be applied upon arbitrary passage list, even those reranked by other methods. In this work, without loss of generality, we employ the two most representative types of retrievers: sparse and dense retriever. Given a query and an initial retrieval list, we first extract similarities between them and a set of anchor texts via both the sparse and dense retrievers. We project and group them to form a set of hybrid and collaborative sequences, each corresponding to a query or passage. Afterwards, the relevance scores between the query and these passages are evaluated in the light of these sequences.

Extensive experiments demonstrate the consistent performance improvement brought by HybRank over passage lists from prevalent retrievers and strong rerankers. We elaborate ablation studies on the collaborative information, feature hybrid, anchor-wise interaction and the number of anchor passages, verifying the impact and indispensability of these components in HybRank.

2 Preliminaries

In this section, we briefly describe the sparse and dense retrieval approaches.

2.1 Sparse Retrieval

Traditionally, text retrieval is dominated by token-matching, where texts are encoded into high-dimensional sparse vectors using the statistic information of tokens. The most commonly-used sparse retrieval methods include TF-IDF, BM25 and so forth. We adopt BM25 score as the similarity metric of sparse retrieval due to its robustness and popularity.

Specifically, given the query q and the document d, the BM25 score is obtained by summing the BM25 weights over the terms co-occurred in q and d:

$$f^s(q,d) = \mathrm{BM25}(q,d) = \sum_{t \in q \cap d} w_t^{\mathrm{RSJ}} \frac{c_{t,d}}{k_1((1-b) + b\frac{|d|}{l}) + c_{t,d}}, \tag{1}$$

where t is a term, w_t^{RSJ} is t's Robertson-Spärck Jones weight, $c_{t,d}$ is the frequency of t in d, $|d|$ is the document length and l is the average length of all documents in the collection. k_1 and b are tunable parameters. Refer to Robertson & Zaragoza (2009) for more details about BM25.

2.2 Dense Retrieval

Owning to the flexibility for a task-specific representation provided by learnable parameters, recent works leverage neural networks to encode text into dense vectors, and search similar documents for queries in vector space. Typically, the query and document are encoded separately, and the relevance score is measured by the similarity of their embeddings. Any neural architectures capable of encoding text into a single fixed-length vector are suitable for dense retrieval. We use the predominant Transformer (Vaswani et al., 2017) encoder and dot product similarity, formulated as

$$f^d(q,d) = \mathrm{T}_q(q)^\top \mathrm{T}_d(d), \tag{2}$$

where $\mathrm{T}_q(\cdot)$ and $\mathrm{T}_d(\cdot)$ are Transformer encoders for queries and documents. Dot product similarity permits offline pre-encoding of large corpus and efficient retrieval using highly optimized vector nearest neighbor searching library (Johnson et al., 2021).

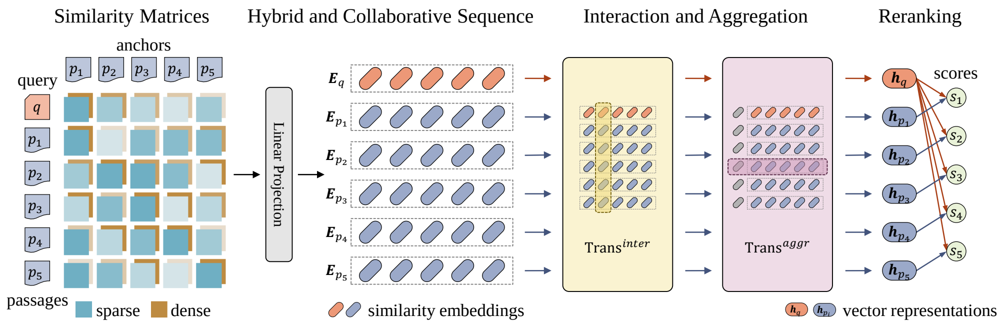


Figure 1: Illustration of HybRank pipeline. For a specific query, the passage list is initialized by an arbitrary retriever or reranker. We display a 5-passage list as an example. First, similarities between query, passages and anchors are derived from sparse and dense retrievers. Then, these similarities are converted to hybrid and collaborative sequences as the representations of query and passages. Finally, these sequences are encoded into dense vectors via interaction and aggregation, and the reranking scores are obtained by dot product of vectors.

3 METHOD

In mainstream information retrieval systems. the first-stage retrieval is designed to fetch a coarse candidate list from a large corpus $\mathcal{C}$. Inevitably, false positives, *i.e.*, irrelevant passages in the retrieval list, are returned in the first-stage retrieval. To improve the precision of retrieval systems, the follow-up procedure reranking aims to distinguish the relevant passages from others in the retrieval list. This paper focuses on the reranking stage.

Formally, given a query q and an initial passage list $\mathcal{P} = [p_1, p_2, \ldots, p_N]$ from upstream retriever, the reranking task is to reorder the passage list by reassigning scores $\mathcal{S} = [s_1, s_2, \ldots, s_N]$ for each of these passages. We denote positive passages in the list as $\mathcal{P}^+$ and negative ones as $\mathcal{P}^-$. In this section, we will present the details of HybRank. The pipeline are illustrated in Figure 1.

3.1 HYBRID AND COLLABORATIVE SEQUENCE

For a specific query, relevant passages tend to describe the same entities, events and relations from the query (Lee et al., 2019). In other words, most passages in the retrieval list would resemble to the true positive ones. Inspired by the success of collaborative filtering (Goldberg et al., 1992) in recommendation systems, we utilize the similarities between passages to distinguish the positive passages in the retrieval list.

Collaborative Sequence Similarity measurements can be naturally derived from retrievers, *e.g.*, BM25 score in sparse retriever and dot product in dense retriever, as discussed in Section 2. We compute the similarity between each passages and a set of anchors, which are the top-L passages of the retrieval list in this work and will collaborate to distinguish the positive passages. These similarity scores between passages can be pre-computed, as HybRank utilizes off-the-shelf retrievers. Denoting similarity score between passage p_i and p_j as $f_{ij} \in \mathbb{R}$, the passage p_i can be represented as a sequence of similarity scalars $\boldsymbol{x}_{p_i} = [f_{i1}, f_{i2}, \ldots, f_{iL}] \in \mathbb{R}^L$.

Nevertheless, according to our observation, the similarity scalars within a retrieval list tend to concentrate on a small range. This is a reasonable phenomenon for that retrievers fetch relatively similar passages from the large corpus. To obtain more distinctive features, we employ a temperature softmax to stretch the distribution of similarities. After that, a min-max normalization is applied to scale them into range $[-1, 1]$. These two transforms are formulated as

$$\begin{aligned} \boldsymbol{x} &= \operatorname{softmax}(\boldsymbol{x}/t), \\ \boldsymbol{x} &= 2 \cdot \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} - 1, \end{aligned} \tag{3}$$

where t is the temperature. Subscripts are omitted for brevity.

Feature Hybrid Similarity metrics of sparse and dense retrievers concentrate on lexical overlap and semantic relevance, respectively. To combine the lexical and semantic properties embedded in sparse and dense retrievers, we mix their similarity scores[2] by stacking them in a channel manner. Formally, we substitute the similarity scalar f_{ij} in $\boldsymbol{x}_{p_i}$ with a vector $\boldsymbol{x}_{ij} = [f_{ij}^s, f_{ij}^d] \in \mathbb{R}^2$, where f_{ij}^s is the sparse similarity computed as Eqn. 1 and f_{ij}^d is the dense similarity computed as Eqn. 2. After that, the representation of passage p_i is turned into a sequence of similarity vectors $\boldsymbol{X}_{p_i} = [\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{iL}] \in \mathbb{R}^{L\times 2}$. Additionally, we map these similarity vectors in the sequence to D dimensions with a trainable linear projection:

$$\boldsymbol{e}_{ij} = \boldsymbol{x}_{ij}\boldsymbol{W}, \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{2\times D}$ is a learnable parameter and $\boldsymbol{e}_{ij} \in \mathbb{R}^D$ are embedded similarities. Thereafter, the passage p_i's representation becomes a sequence of similarity embeddings $\boldsymbol{E}_{p_i} = [\boldsymbol{e}_{i1}, \boldsymbol{e}_{i2}, \ldots, \boldsymbol{e}_{iL}] \in \mathbb{R}^{L\times D}$, which comprises the similarity information between p_i and anchor passages originating from both sparse and dense retrievers. These similarities deliver substantial information for the collaboration of passages and hold both the lexical and semantic properties from retrievers. With the same procedure, we compute the similarities between query and anchors, and derive the query representation $\boldsymbol{E}_q = [\boldsymbol{e}_{q1}, \boldsymbol{e}_{q2}, \ldots, \boldsymbol{e}_{qL}] \in \mathbb{R}^{L\times D}$. Noted that the similarities from sparse and dense retriever are stretched and normalized individually before linear projection, as described in Eqn. 3.

Consequently, we obtain $N + 1$ collaborative sequences in total, each representing a passage or a query and consisting of their embedded similarities with L anchor passages.

3.2 Interaction and Aggregation

Following the prevalent sequence similarity learning paradigm in the field of natural language processing (Reimers & Gurevych, 2019; Gao et al., 2021b), we measure the relevance between the collaborative sequences of query and passages in vector space. The vector representations are obtained by an anchor-wise interaction and a sequence aggregation in HybRank.

Anchor-wise Interaction The j-th elements $\boldsymbol{e}_{*j}$ in these collaborative sequences $\boldsymbol{E}_*$ indicate the similarities between retrieved passages and the j-th anchor passage. The importance of these anchors varies since they are picked with a single strategy. Specifically, an anchor is worthy of more consideration if showing strong correlation with a majority of retrieved passages, and vice versa.

To assess the quality of anchor passages, we conduct anchor-wise interaction. Concretely, for each position j, we collect the j-th similarity embedding $\boldsymbol{e}_{*j}$ from query sequence and every passage sequences, and feed them into a Transformer encoder, denoted as

$$\boldsymbol{e}'_{qj}, \boldsymbol{e}'_{1j}, \boldsymbol{e}'_{2j}, \ldots, \boldsymbol{e}'_{Nj} = \mathrm{Trans}^{inter}(\boldsymbol{e}_{qj}; \boldsymbol{e}_{1j}; \boldsymbol{e}_{2j}; \ldots; \boldsymbol{e}_{Nj}), \tag{5}$$

where $\boldsymbol{e}'_{*j} \in \mathbb{R}^D$. Position embeddings are added to $\boldsymbol{e}_{*j}$ according to its rank "$*$" for retaining the passage rank information. Subsequently, the similarity embedding sequences $\boldsymbol{E}_*$ are converted to $\boldsymbol{E}'_* = [\boldsymbol{e}'_{*1}, \boldsymbol{e}'_{*2}, \ldots, \boldsymbol{e}'_{*L}]$ and enhanced with the importance information of anchor passages.

Sequence Aggregation We encode these sequences into dense vectors by aggregating the enhanced similarity embeddings. To be specific, we prepend a [CLS] embedding to the collaborative sequence, feed the extended sequence into another Transformer encoder and use the output of [CLS] as the representation of p_i, formulated as

$$\boldsymbol{h}_{p_i} = \mathrm{Trans}^{aggr}([\mathrm{CLS}] \oplus \boldsymbol{E}'_{p_i})_{[\mathrm{CLS}]}, \tag{6}$$

where $[\mathrm{CLS}] \in \mathbb{R}^{1\times D}$, $\boldsymbol{E}'_{p_i} \in \mathbb{R}^{L\times D}$ and $\oplus$ denotes the concatenation of two sequences. $\boldsymbol{h}_{p_i} \in \mathbb{R}^D$ is the vector representation of passage p_i. The query representation $\boldsymbol{h}_q \in \mathbb{R}^D$ is derived analogously. Global receptive field is provided by the anchor-wise interaction and sequence aggregation. We discuss more about the receptive field in Section A.

[2]In this paper, we refer to similarity score from sparse and dense retrievers as sparse similarity and dense similarity, respectively.

3.3 RERANKING AND TRAINING

Reranking Considering that query and passages have been converted into dense vectors encoded with collaborative information, we have several alternatives to judge the vector similarity as the relevance between query and passages. We use dot product in this work and thus the relevance between query q and p_i is computed by

$$s_i = \boldsymbol{h}_q^\top \boldsymbol{h}_{p_i}. \tag{7}$$

Then the passages are sorted in descending order of their relevance s_i with query.

Training In order to assign high scores to relevant passages and low scores to irrelevant ones, HybRank needs to pull together the representation of relevant passages and query, while push the representation of irrelevant ones as apart from the query as possible. As there may exists more than one positive passage in the list, vanilla softmax loss fails to be directly applied to HybRank. We adopt the supervised contrastive loss (Khosla et al., 2020) to cope with multiple positives, which performs summation over positives outside the log function in softmax. The loss is formulated as

$$\mathcal{L}(q, \mathcal{P}) = -\frac{1}{|\mathcal{P}^+|} \sum_{p_i \in \mathcal{P}^+} \log \frac{\exp(s_i/\tau)}{\sum_{p_j \in \mathcal{P}} \exp(s_j/\tau)}, \tag{8}$$

where $|\mathcal{P}^+|$ is the number of positive passages in the retrieval list, and τ is the tunable temperature.

4 EXPERIMENTS

4.1 DATASETS

Natural Questions (Kwiatkowski et al., 2019) consists of real questions from Google search engine with golden passages from Wikipedia pages and answer span annotations. Following the settings from Karpukhin et al. (2020), we report the test set top-k accuracy (R@k), which evaluates the percentage of queries whose top-k retrieved passages contain the answers.

MS MARCO (Bajaj et al., 2018) collects queries from Bing search logs and was originally designed for machine reading comprehension. Following previous works (Qu et al., 2021; Ren et al., 2021b), we evaluate the dev set R@k as well as Mean Reciprocal Rank (MRR), which means the average reciprocal of the first retrieved relevant passage rank.

4.2 IMPLEMENTATION DETAILS

HybRank is a flexible plug-in reranker, which can be applied on arbitrary passage list even these reranked by other methods. We adopt the dense retrievers, which outperform sparse ones after elaborated pre-training (Chang et al., 2020; Gao & Callan, 2021; 2022) and fine-tuning (Sachan et al., 2021), as well as strong cross-encoder based rerankers, to initialize the passage list. We simply select all of the passages in the initial list as anchors. The impact of anchor passages will be discussed in Section 4.4. These methods are implemented using RocketQA toolkit[3] and Pyserini toolkit (Lin et al., 2021a) which is built on Lucene[4] and FAISS (Johnson et al., 2021).

The hyper-parameters in HybRank are as follows. The temperature t in the feature normalization is set to 100 and 10 for sparse and dense similarity, respectively. We randomly initialize a 2-layer Transformer encoder for Trans^{inter} and 1-layer for Trans^{aggr} using Huggingface Transformers (Wolf et al., 2020). The embedding dimension, MLP inner-layer dimension and number of heads are 64, 256 and 8, respectively. The temperature τ in the loss function is 0.07. We adopt the Adam optimizer with an initial learning rate 1×10^{-3} with the warm-up ratio 0.1, followed by a cosine learning rate decay. We use gradient clipping of 2 and weight decay of 1×10^{-6}. We train the model for 100 epochs with batch size 32, which takes about 13 hours on Natural Questions and 4 days on MS MARCO. All experiments are conducted on a single NVIDIA RTX 3090 GPU.

Table 1: The reranking performance of HybRank on Natural Questions. We build HybRank upon DPR (Karpukhin et al., 2020), DKRR (Izacard & Grave, 2021), ANCE (Xiong et al., 2021), RocketQA (Qu et al., 2021) and RocketQAv2 (Ren et al., 2021b). The performance of these baselines and HybRank built upon them are on the side of arrows. Improvements brought by HybRank are highlighted in bold.

Methods	R@1	R@5	R@20
DPR-Multi + HybRank	45.82 → 51.99 (**+6.17**)	68.12 → 72.71 (**+4.59**)	80.30 → 83.24 (**+2.94**)
DPR-Single + HybRank	47.95 → 53.13 (**+5.18**)	69.39 → 73.05 (**+3.66**)	80.97 → 82.99 (**+2.02**)
DKRR + HybRank	50.36 → 52.85 (**+2.49**)	74.10 → 74.46 (**+0.36**)	84.27 → 84.49 (**+0.22**)
ANCE + HybRank	52.66 → 53.63 (**+0.97**)	72.66 → 73.57 (**+0.91**)	83.05 → 83.88 (**+0.83**)
RocketQA-retriever + HybRank	51.75 → 56.07 (**+4.32**)	74.02 → 77.04 (**+3.02**)	83.99 → 85.68 (**+1.69**)
RocketQA-reranker + HybRank	54.60 → 59.83 (**+5.23**)	76.59 → 78.73 (**+2.14**)	85.01 → 86.40 (**+1.39**)
RocketQAv2-retriever + HybRank	55.57 → 56.98 (**+1.41**)	75.98 → 76.65 (**+0.67**)	84.46 → 85.76 (**+1.30**)
RocketQAv2-reranker + HybRank	57.17 → 59.50 (**+2.33**)	75.98 → 78.34 (**+2.36**)	84.71 → 86.26 (**+1.55**)

Table 2: The reranking performance of HybRank on MS MARCO. We built HybRank upon DistilBERT-KD (Hofstätter et al., 2021a), ANCE (Xiong et al., 2021), TCT-ColBERT-v1 (Lin et al., 2020), TAS-B (Hofstätter et al., 2021b), TCT-ColBERT-v2 (Lin et al., 2021b), RocketQA (Qu et al., 2021) and RocketQAv2 (Ren et al., 2021b). The performance of these baselines and HybRank built upon them are on the side of arrows. Improvements brought by HybRank are highlighted in bold.

Methods	MRR@10	R@10	R@50
DistilBERT-KD + HybRank	32.50 → 36.24 (**+3.74**)	58.77 → 64.40 (**+5.63**)	79.24 → 82.02 (**+2.78**)
ANCE + HybRank	33.01 → 36.44 (**+3.43**)	59.44 → 64.63 (**+5.19**)	80.10 → 82.79 (**+2.69**)
TCT-ColBERT-v1 + HybRank	33.49 → 36.23 (**+2.74**)	60.46 → 64.96 (**+4.50**)	80.67 → 83.44 (**+2.77**)
TAS-B + HybRank	34.44 → 36.38 (**+1.94**)	62.94 → 65.77 (**+2.83**)	83.44 → 84.71 (**+1.27**)
TCT-ColBERT-v2 + HybRank	35.85 → 37.55 (**+1.70**)	63.64 → 66.39 (**+2.75**)	83.31 → 84.97 (**+1.66**)
RocketQA-retriever + HybRank	35.76 → 37.96 (**+2.20**)	64.01 → 67.12 (**+3.11**)	83.41 → 85.59 (**+2.18**)
RocketQA-reranker + HybRank	40.50 → 40.98 (**+0.48**)	69.81 → 70.40 (**+0.59**)	86.46 → 86.55 (**+0.09**)
RocketQAv2-retriever + HybRank	37.28 → 38.69 (**+1.41**)	65.72 → 67.92 (**+2.20**)	84.04 → 85.70 (**+1.66**)
RocketQAv2-reranker + HybRank	41.15 → 41.40 (**+0.25**)	69.99 → 70.37 (**+0.38**)	86.55 → 86.68 (**+0.13**)

4.3 Results

Table 1 and Table 2 summarize the performance of HybRank and baselines on the Natural Questions and MS MARCO datasets. More detailed evaluation results are listed in Section C. Some of these retrievers involve both sparse and dense similarity from different perspectives. DPR (Karpukhin et al., 2020) selects hard negative samples from passages returned by BM25; DKRR (Izacard & Grave, 2021) starts its iterative training with passages retrieved using BM25; TCT-ColBERT-v1 (Lin et al., 2020) proposes an alternative approximation for linear combination of dense and sparse retrieval; TCT-ColBERT-v2 (Lin et al., 2021b) further studies the dense-sparse hybrid in terms of quality, time and space. Besides, ANCE (Xiong et al., 2021) discovers new negatives via nearest neighbor searching during model training; TAS-B (Hofstätter et al., 2021b) proposes balanced sampling strategies to compose informative training batches; DistilBERT-KD (Hofstätter et al., 2021a) leverages cross-architecture knowledge distillation for model-agnostic training.

From the results we can observe that HybRank shows a consistent improvements over upstream retrievers and even rerankers. In general, HybRank based on stronger baselines can produce better reranking results. Specifically, HybRank built upon the retriever of RocketQA outperforms the reranker of RocketQA on Natural Questions, and the same phenomenon can be observed on RocketQAv2 in most evaluation metrics. Additionally, HybRank built opon their rerankers further im-

[3]`https://github.com/PaddlePaddle/RocketQA`.

[4]`https://lucene.apache.org`.

Table 3: The results of ablation study for collaborative features, anchor-wise interaction and anchor passages. Query-passage similarities, anchor-wise interaction and collaborative features are omitted in "w/o q-p", "w/o inter" and "w/o collab", respectively. Anchors are randomly selected in "r/d anchor".

	R@1	R@5	R@10	R@20	R@50
retriever	45.82	68.12	75.24	80.30	84.57
r/d anchor	46.18	68.84	75.43	80.91	85.01
w/o q-p	47.12	69.17	75.54	80.47	85.07
w/o inter	49.92	69.61	76.32	81.02	84.99
w/o collab	50.78	**72.91**	**79.28**	83.10	85.79
HybRank	**51.99**	72.71	79.03	**83.24**	**85.93**

Table 4: The results of ablation study for feature hybrid. "list" specifies what type of retriever is used when initializing passage list. "feature" indicates the input feature and "none" stands for the assessment of initial passage list.

list	feature	R@1	R@5	R@10	R@20	R@50
sparse	none	23.82	45.18	55.54	63.93	73.55
	sparse	30.50	50.39	59.00	67.26	75.24
	dense	47.01	64.68	**70.39**	**74.49**	**77.81**
	hybrid	**47.15**	**64.82**	69.78	74.32	77.65
dense	none	45.82	68.12	75.24	80.30	84.57
	dense	46.70	68.45	75.04	80.19	84.88
	sparse	50.89	71.86	78.98	83.16	85.90
	hybrid	**51.99**	**72.71**	**79.03**	**83.24**	**85.93**

proves the performance on both datasets. These results prove the advantage of reranking based on arbitrary off-the-shelf retrievers and even other reranked results, which distinguishes HybRank from other reranking methods.

The most surprising aspect of these results is that, in spite of inferior reranking results, weak retrievers gain more relative improvements from HybRank than strong ones. This result may be explained by the fact that HybRank relies heavily on the complementary information provide by sparse similarity. Weak retrievers receive relatively more valuable information from sparse similarity than strong retrievers, and accordingly improve more performance over upstream retrievers. We will discuss more on sparse-dense hybrid in Section 4.4.

4.4 Analysis

In this section, we conduct ablation studies and discuss the impact of the core components of HybRank: the hybrid and collaborative features, the anchor-wise interaction and the number of anchor passages. All experiments are performed on Natural Questions dataset with DPR-Multi retriever.

Collaborative Feature The main difference between HybRank and other works is, it leverages the collaborative information between retrieved passages. To verify the impact of passage collaboration on reranking, we omit the collaborative feature in "w/o collab" by substituting query-passage similarities for collaborative sequences, *i.e.*, representing each passage as a one-token sequence according to its similarities with query. Besides, we exclude the query-passage similarity in "w/o q-p" by representing query via a learnable token rather than aggregated collaborative sequence. The results are presented in Table 3, where "retriever" denotes the assessment of initial passage list.

Table 3 indicates that "w/o collab" shows an appreciable gain over "retriever", demonstrating that query-passage similarity is an essential and indispensable feature for HybRank. The most remarkable phenomenon is, "w/o q-p" surpasses "retriever" by a large margin, despite the fact that "w/o q-p" is completely unaware of the query. Namely, HybRank has the ability to distinguish the positive even only with the collaborative information among passages. Furthermore, standing on the shoulder of query-passage similarity, HybRank achieves even better results than "w/o collab", which sufficiently substantiates the reranking capability of collaborative information.

Anchor-wise Interaction Apart from the collaborative sequence itself, anchor-wise interaction provides extra collaboration between sequences. We eliminate the Trans^{inter} and directly aggregate the linear projected collaborative sequence to study the effectiveness of anchor-wise interaction.

Table 3 shows that there is a noticeable drop of performance with anchor-wise interaction removed. The discrepancy could be attributed to the restricted receptive field. "w/o inter" individually encodes each collaborative sequences of query and passages into dense vectors without anchor-wise interaction. In this manner, the relevance of these sequences is evaluated only in vector space where sequence information are severely compressed and not expressive enough. In contrast, equipped with anchor-wise interaction, HybRank is capable of obtaining a global receptive field. Each elements in these sequences captures the context of elements in all sequences, enabling more informative vector representation and fine-grained relevance estimation.

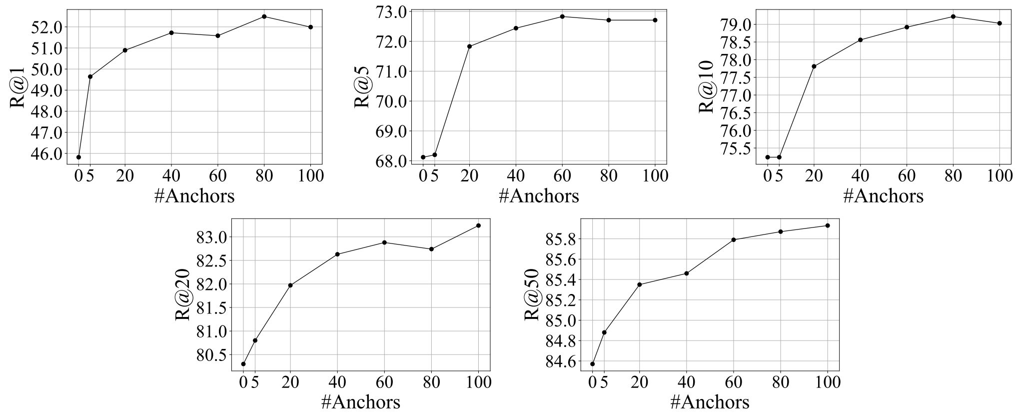


Figure 2: Impact of the number of anchor passages. We conduct experiments with anchor number $5, 20, 40, 60, 80, 100$. The metric of anchor number 0 denotes the assessment of initial retrieval list.

Feature Hybrid Despite the fact that the similarities of sparse and dense retriever reflect different aspect of linguistics, *i.e.*, lexical overlap and semantic relevance, both of them tend to have collaborative property. Hence, it is more nature and easier to mix sparse and dense retrieval from the the perspective of collaboration. To illustrate the complementarity of sparse and dense features and the necessity of feature hybrid in HybRank, we separately validate the effect of the two individual features and their hybrid. The ablations are conducted not only on initial passage list retrieved by dense retriever, but also list retrieved by sparse retriever for integrity and comparison.

Identical trends can be observed from the two set of experiments in Table 4. There are limited performance gains when retrievers used for passage retrieval and similarity computation are same, but dramatically increases when they are different. Furthermore, slight additional improvements can be seen with the hybrid of the two features on both settings. These phenomena reveal that the main performance gains originate from the retriever different with that in retrieval stage, while the same type only plays an auxiliary role. Consequently, we draw the credible conclusion that different types of similarities provide additional complementary information over the initial passage list.

Moreover, regardless of feature used, HybRank achieves better results on initial passage list retrieved by dense retriever than sparse one, as more positives are contained in the dense retrieved list. This also corroborates the findings of Section 4.3 that superior initial passage list leads to better reranking results with HybRank.

Number of Anchor Passages We evaluate the performance of HybRank under different number of anchors to study its impact. What can be clearly seen in Figure 2 is a consistent growth of performance as the anchor number L increases. The underlying philosophy is that with more anchor passages, the passage list can derive more agreement to facilitate the collaboration between passages and alleviate the effect of noisy ones. The positive correlation between the performance and anchor number indicates the effect of collaborative information in the retrieval list.

Despite the consistent growth with anchor number, the rate of performance increase begins to slow down when the number of anchors is greater than 60. Anchor passages are used for deriving collaborative information, and thus with more diverse anchors we can obtain more distinctive collaborative features. As the anchor number approaches to 100, the diversity of passages levels off, leading to stable performance with larger anchor numbers.

As L increase to a very large number, the average relevance of anchors will degrade to a low level. A legitimate concern may be that poor quality anchor set would pollute the collaborative aspect. Due to the $O(L^2)$ computational complexity of sequence aggregation in HybRank, it is hard to directly perform experiments on large L. But we simulate the poor quality anchor set by randomly selecting anchor passages from corpus $\mathcal{C}$. "r/d anchor" in Table 3 indicates that random anchors slightly improves the performance but still lags far behind the relevant anchors, demonstrating the benefits of collaborative information and the predominance of the anchor quality.

Nevertheless, the selection of anchor passages is flexible. Ideally, more elaborated anchor passage selection would further enhance the performance of HybRank. We leave the exploration of other anchor selecting strategy as a future work.

5 RELATED WORK

5.1 TEXT RETRIEVAL

Retrieval is the first stage of information retrieval which requires high recall to cover more relevant document in the retrieval list. Traditional sparse approaches like TF-IDF and BM25 (Robertson & Zaragoza, 2009) rely on lexical overlap between query and documents. Although having dominated the field of text retrieval for a long time, these sparse methods suffer from lexical gap (Berger et al., 2000), namely, the synonymy problem. To tackle this issue, earlier techniques (Nogueira et al., 2019; Dai & Callan, 2020) adopt neural networks to reinforce the sparse methods. Recently proposed dense retrieval approaches (Karpukhin et al., 2020; Xiong et al., 2021) directly encode the query and passages into dense vectors via dual-encoder, which captures semantic in text and are capable of low-latency search via highly optimized algorithms, *e.g.*, FAISS (Johnson et al., 2021).

These two types of methods are not mutually exclusive and one's weakness is the other's strength. Some researchers combine the sparse and dense methods by score ensemble, improved training or trade-off model between sparse and dense retriever. Karpukhin et al. (2020) samples hard negatives from sparse retriever for the training of dense retriever. Seo et al. (2019), Khattab & Zaharia (2020) and Santhanam et al. (2022) index terms or phrases instead of documents for more fine-grained similarity and higher efficiency. Lin et al. (2020) and Luan et al. (2021) explore the linear sparse-dense score combination and its alternatives. Gao et al. (2021a) and Yang et al. (2021) leverages the lexical matching or token-level interaction signals to train the dense retriever.

However, among these methods, score ensemble lacks sufficient interaction of sparse and dense methods, smaller units indexing sacrifices efficiency, and retraining one type of retrieval method with the help of the other discards its origin ranking capability. In contrast, our method can be applied to arbitrary passage list, incorporating the lexical and semantic properties of off-the-shelf retrievers and meanwhile ensuring the generality and flexibility.

5.2 TEXT RERANKING

The second stage reranking is based on the results of retrieval system and aims to create a more fine-grained comparison within retrieval list. Typically, cross-encoder is utilized to capture the interactions between query and passage in token-level. Nogueira & Cho (2020) and Sun et al. (2021) adopts BERT (Devlin et al., 2019) to achieve token-level interactions with attention mechanism (Vaswani et al., 2017). To reduce the massive computation overhead (Reimers & Gurevych, 2019), Khattab & Zaharia (2020) and Gao et al. (2020) propose a lightweight interaction on dense representations from retrievers. While based on first-stage retrieval, these methods individually compute the relevance for each retrieved passage, omitting the extra information implied by the whole list and requiring multiple runs.

Several pseudo-relevance feedback approaches (He & Ounis, 2009; Zamani & Croft, 2016; Zamani et al., 2016) aim to refine the query model with the top-retrieved documents. Listwise context is also well explored in multi-stage recommendation systems (Liu et al., 2022), such as PRM (Pei et al., 2019), which regards each item as a token, learns the mutual influence between items using self-attention and reranks all items altogether. Different from prior studies, our method extracts the collaborative feature from retrieval list, represents the query and each passages as hybrid and collaborative sequences, and measures the relevance between query and passages using these sequences.

6 CONCLUSION

We introduce HybRank, a hybrid and collaborative passage reranking method. HybRank extracts the similarities between texts via off-the-shelf retrievers, to form hybrid and collaborative sequences as the representations of query and passages. Efficient reranking is based on these sequences incorporating the lexical and semantic properties of sparse and dense retrievers. Extensive experiments confirm the effectiveness of HybRank built upon arbitrary initial passage list. Elaborated ablation studies investigate the impact of core components in HybRank. We hope our work could provide inspiration for researchers in the field of information retrieval, and steer more exploration on collaboration and correlation between texts.

REFERENCES

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268*, 2018.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 192–199. ACM Press, 2000.

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *Proceedings of the International Conference on Learning Representations*, 2020.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1870–1879. Association for Computational Linguistics, 2017.

Zhuyun Dai and Jamie Callan. Context-Aware Term Weighting For First Stage Passage Retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1533–1536. ACM, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Luyu Gao and Jamie Callan. Condenser: A Pre-training Architecture for Dense Retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 981–993. Association for Computational Linguistics, 2021.

Luyu Gao and Jamie Callan. Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 2843–2853. Association for Computational Linguistics, 2022.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Modularized Transfomer-based Ranking Framework. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4180–4190. Association for Computational Linguistics, 2020.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. Complementing Lexical Retrieval with Semantic Residual Embedding. *arXiv:2004.13969*, 2021a.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910. Association for Computational Linguistics, 2021b.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

Ben He and Iadh Ounis. Finding Good Feedback Documents. In *Proceedings of the ACM Conference on Information and Knowledge Management*, pp. 2011–2014. Association for Computing Machinery, 2009.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial Attention in Multidimensional Transformers. *arXiv:1912.12180*, 2019.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. *arXiv:2010.02666*, 2021a.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 113–122. ACM, 2021b.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. Embedding-based Retrieval in Facebook Search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2553–2561. ACM, 2020.

Gautier Izacard and Edouard Grave. Distilling Knowledge from Reader to Retriever for Question Answering. In *Proceedings of the International Conference on Learning Representations*, 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781. Association for Computational Linguistics, 2020.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48. ACM, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020.

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. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096. Association for Computational Linguistics, 2019.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2356–2362. ACM, July 2021a.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Distilling Dense Representations for Ranking using Tightly-Coupled Teachers. *arXiv:2010.11386*, 2020.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. In-Batch Negatives for Knowledge Distillation with Tightly-Coupled Teachers for Dense Retrieval. In *Proceedings of the Workshop on Representation Learning for NLP*, pp. 163–173. Association for Computational Linguistics, 2021b.

Weiwen Liu, Yunjia Xi, Jiarui Qin, Fei Sun, Bo Chen, Weinan Zhang, Rui Zhang, and Ruiming Tang. Neural Re-ranking in Multi-stage Recommender Systems: A Review. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 5512–5520. International Joint Conferences on Artificial Intelligence Organization, 2022.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021.

Rodrigo Nogueira and Kyunghyun Cho. Passage Re-ranking with BERT. *arXiv:1901.04085*, 2020.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document Expansion by Query Prediction. *arXiv:1904.08375*, 2019.

Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, and Dan Pei. Personalized Re-Ranking for Recommendation. In *Proceedings of the ACM Conference on Recommender Systems*, pp. 3–11. Association for Computing Machinery, 2019.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5835–5847. Association for Computational Linguistics, 2021.

Razieh Rahimi, Azadeh Shakery, Javid Dadashkarimi, Mozhdeh Ariannezhad, Mostafa Dehghani, and Hossein Nasr Esfahani. Building a Multi-Domain Comparable Corpus Using a Learning to Rank Method. *Natural Language Engineering*, 22(4):627–653, 2016.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pp. 3982–3992. Association for Computational Linguistics, 2019.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. PAIR: Leveraging Passage-Centric Similarity Relation for Improving Dense Passage Retrieval. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pp. 2173–2183. Association for Computational Linguistics, 2021a.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2825–2835. Association for Computational Linguistics, 2021b.

Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. End-to-End Training of Neural Retrievers for Open-Domain Question Answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pp. 6648–6662. Association for Computational Linguistics, 2021.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734. Association for Computational Linguistics, 2022.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 4430–4441. Association for Computational Linguistics, 2019.

Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. Few-Shot Text Ranking with Meta Adapted Synthetic Weak Supervision. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pp. 5030–5043. Association for Computational Linguistics, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning to Rank: Theory and Algorithm. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, volume 307, pp. 1192–1199. ACM, 2008.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *Proceedings of the International Conference on Learning Representations*, 2021.

Yinfei Yang, Ning Jin, Kuo Lin, Mandy Guo, and Daniel Cer. Neural Retrieval for Question Answering with Cross-Attention Supervised Data Augmentation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pp. 263–268. Association for Computational Linguistics, 2021.

Hamed Zamani and W Bruce Croft. Estimating Embedding Vectors for Queries. In *Proceedings of the ACM International Conference on the Theory of Information Retrieval*, pp. 123–132, 2016.

Hamed Zamani, Javid Dadashkarimi, Azadeh Shakery, and W. Bruce Croft. Pseudo-Relevance Feedback Based on Matrix Factorization. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1483–1492. ACM, 2016.

ChengXiang Zhai and John D. Lafferty. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the International Conference on Information and Knowledge Management*, pp. 403–410. ACM, 2001.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. Adversarial Retriever-Ranker for Dense Text Retrieval. In *Proceedings of the International Conference on Learning Representations*, 2022.

A Receptive Field and Complexity

Interestingly, from another perspective, the anchor-wise interaction plus sequence aggregation equals to a column-wise and a row-wise attention applied on the matrix formulated by similarities of query, passages and anchors. Global receptive field is provided by these two axial-wise attention (Ho et al., 2019). Consequently, similarity vector $\boldsymbol{x}_{ij}$ perceives with each other, and the vector representations of query and passages are aware of the collaborative information among others.

A more direct approach to obtain global receptive field is element-wise interaction. Concretely, we can feed the concatenation of all sequences $\boldsymbol{E}$ into a single Transformer encoder, and output representations for each passage and query via multiple separate [CLS] tokens. However, due to the self-attention operation in Transformer, the computational complexity of element-wise interaction achieves $O(N^2L^2)$. In contrast, our method reduce the complexity to $O(N^2L + NL^2)$, by decomposing the element-wise attention upon the matrix composed of similarity sequences into axial-wise.

B Datasets Details

Table 5: Statistics of Natural Questions and MS MARCO datasets. Q and P are abbreviations for query and passage, respectively.

Datasets	#Q in Train	#Q in Dev	#Q in Test	#P	Avg. Q Length	Avg. P Length
Natural Questions	58,812	-	3,610	21,015,324	9.20	100.0
MS MARCO	502,939	6,980	6,837	8,841,823	5.97	56.58

C Full Evaluation Results

We present the full evaluation results on Natural Questions and MS MARCO in Table 6 and 7.

Table 6: The reranking performance of HybRank on Natural Questions. We build HybRank upon DPR-Multi (Karpukhin et al., 2020), DPR-Single (Karpukhin et al., 2020), DKRR (Izacard & Grave, 2021), ANCE (Xiong et al., 2021), the retriever and reranker of RocketQA (Qu et al., 2021) and RocketQAv2 (Ren et al., 2021b). Improvements brought by HybRank are highlighted in bold.

Methods	R@1	R@5	R@10	R@20	R@50
DPR-Multi	45.82	68.12	75.24	80.30	84.57
DPR-Multi + HybRank	51.99 (**+6.17**)	72.71 (**+4.59**)	79.03 (**+3.79**)	83.24 (**+2.94**)	85.93 (**+1.36**)
DPR-Single	47.95	69.39	75.93	80.97	84.90
DPR-Single + HybRank	53.13 (**+5.18**)	73.05 (**+3.66**)	78.84 (**+2.91**)	82.99 (**+2.02**)	85.93 (**+1.03**)
DKRR	50.36	74.10	79.78	84.27	87.89
DKRR + HybRank	52.85 (**+2.49**)	74.46 (**+0.36**)	80.50 (**+0.72**)	84.49 (**+0.22**)	88.06 (**+0.17**)
ANCE	52.66	72.66	78.70	83.05	86.29
ANCE + HybRank	53.63 (**+0.97**)	73.57 (**+0.91**)	79.28 (**+0.58**)	83.88 (**+0.83**)	87.12 (**+0.83**)
RocketQA-retriever	51.75	74.02	80.00	83.99	87.34
RocketQA-retriever + HybRank	56.07 (**+4.32**)	77.04 (**+3.02**)	82.30 (**+2.30**)	85.68 (**+1.69**)	88.17 (**+0.83**)
RocketQA-reranker	54.60	76.59	81.44	85.01	88.17
RocketQA-reranker + HybRank	59.83 (**+5.23**)	78.73 (**+2.14**)	82.83 (**+1.39**)	86.40 (**+1.39**)	88.42 (**+0.25**)
RocketQAv2-retriever	55.57	75.98	81.08	84.46	87.92
RocketQAv2-retriever + HybRank	56.98 (**+1.41**)	76.65 (**+0.67**)	81.94 (**+0.86**)	85.76 (**+1.30**)	88.61 (**+0.69**)
RocketQAv2-reranker	57.17	75.98	81.00	84.71	87.92
RocketQAv2-reranker + HybRank	59.50 (**+2.33**)	78.34 (**+2.36**)	83.24 (**+2.24**)	86.26 (**+1.55**)	88.75 (**+0.83**)

Table 7: The reranking performance of HybRank on MS MARCO. We built HybRank upon DistilBERT-KD (Hofstätter et al., 2021a), ANCE (Xiong et al., 2021), TCT-ColBERT-v1 (Lin et al., 2020), TAS-B (Hofstätter et al., 2021b), TCT-ColBERT-v2 (Lin et al., 2021b), the retriever and reranker of RocketQA (Qu et al., 2021) and RocketQAv2 (Ren et al., 2021b). Improvements brought by HybRank are highlighted in bold.

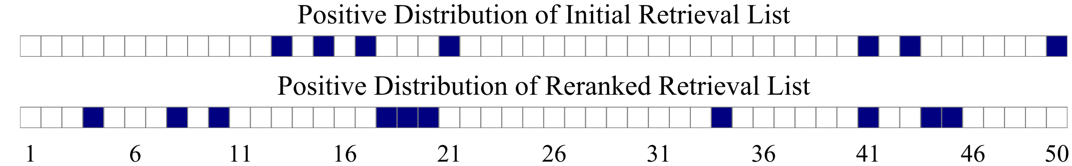
Methods	MRR@10	MRR@100	R@1	R@10	R@50
DistilBERT-KD	32.50	33.61	21.23	58.77	79.24
DistilBERT-KD + HybRank	36.24 (**+3.74**)	37.21 (**+3.60**)	23.98 (**+2.75**)	64.40 (**+5.63**)	82.02 (**+2.78**)
ANCE	33.01	34.16	21.55	59.44	80.10
ANCE + HybRank	36.44 (**+3.43**)	37.45 (**+3.29**)	24.23 (**+2.68**)	64.63 (**+5.19**)	82.79 (**+2.69**)
TCT-ColBERT-v1	33.49	34.62	21.92	60.46	80.67
TCT-ColBERT-v1 + HybRank	36.23 (**+2.74**)	37.25 (**+2.63**)	23.48 (**+1.56**)	64.96 (**+4.50**)	83.44 (**+2.77**)
TAS-B	34.44	35.58	22.06	62.94	83.44
TAS-B + HybRank	36.38 (**+1.94**)	37.41 (**+1.83**)	23.75 (**+1.69**)	65.77 (**+2.83**)	84.71 (**+1.27**)
TCT-ColBERT-v2	35.85	36.95	23.64	63.64	83.31
TCT-ColBERT-v2 + HybRank	37.55 (**+1.70**)	38.58 (**+1.63**)	24.87 (**+1.23**)	66.39 (**+2.75**)	84.97 (**+1.66**)
RocketQA-retriever	35.76	36.84	23.70	64.01	83.41
RocketQA-retriever + HybRank	37.96 (**+2.20**)	38.98 (**+2.14**)	25.21 (**+1.51**)	67.12 (**+3.11**)	85.59 (**+2.18**)
RocketQA-reranker	40.50	41.43	27.22	69.81	86.46
RocketQA-reranker + HybRank	40.98 (**+0.48**)	41.89 (**+0.46**)	27.62 (**+0.40**)	70.40 (**+0.59**)	86.55 (**+0.09**)
RocketQAv2-retriever	37.28	38.29	24.90	65.72	84.04
RocketQAv2-retriever + HybRank	38.69 (**+1.41**)	39.67 (**+1.38**)	25.95 (**+1.05**)	67.92 (**+2.20**)	85.70 (**+1.66**)
RocketQAv2-reranker	41.15	42.08	27.81	69.99	86.55
RocketQAv2-reranker + HybRank	41.40 (**+0.25**)	42.32 (**+0.24**)	28.08 (**+0.27**)	70.37 (**+0.38**)	86.68 (**+0.13**)

D Reranking Cases

We present reranking cases in Figure 3 and Figure 4. The first line in the figure is the query sentence. We illustrate the distribution of positives in the passage list before and after reranking. Blue squares indicate positive passages while white squares stand for negative passages in the retrieval list. We only show top-50 out of 100 passages in these lists due to the space limitation. Following the positive distribution, we list several raw texts of reranked passages for the question. The titles of each passages and the answers for each questions are bold and blue, respectively.

Observed from the distribution visualization and rank changes of passages, the positive distributions shift toward the front of the lists as the quantitative analysis in Section 4.3. Ranks of many positive passages are raised by a large margin. Besides, it is apparent that positive passages tend to describe the same entities, events and relations as discussed in Section 1. Case 1 in Figure 4 involves "the king of England" while case 2 in Figure 4 is about "Where's Waldo".

Query: ***Who was the king of England in 1756?***



Positive Passages	Rank Changes
George II of Great Britain. **George II** of Great Britain **George II** (George Augustus; ; 30 October / 9 November 1683 – 25 October 1760) was King of Great Britain and Ireland, Duke of Brunswick-Lüneburg (Hanover) and a prince-elector of the Holy Roman Empire from 11 June 1727 (O.S.) until his death in 1760. George was the last British monarch born outside Great Britain: he was born and brought up in northern Germany. His grandmother, Sophia of Hanover, became second in line to the British throne after about 50 Catholics higher in line were excluded by the Act of Settlement 1701 and the Acts of	$15 \rightarrow 4$ (11 $\uparrow$)
George II of Great Britain. by his grandson, George III. For two centuries after **George II**'s death, history tended to view him with disdain, concentrating on his mistresses, short temper, and boorishness. Since then, most scholars have reassessed his legacy and conclude that he held and exercised influence in foreign policy and military appointments. George was born in the city of Hanover in Germany, and was the son of George Louis, Hereditary Prince of Brunswick-Lüneburg (later King George I of Great Britain), and his wife, Sophia Dorothea of Celle. His sister, Sophia Dorothea, was born when he was three years old. Both of George's parents	$74 \rightarrow 8$ (66 $\uparrow$)
Monarchy of the United Kingdom. Britain was now in personal union. Power shifted towards George's ministers, especially to Sir Robert Walpole, who is often considered the first British prime minister, although the title was not then in use. The next monarch, **George II**, witnessed the final end of the Jacobite threat in 1746, when the Catholic Stuarts were completely defeated. During the long reign of his grandson, George III, Britain's American colonies were lost, the former colonies having formed the United States of America, but British influence elsewhere in the world continued to grow, and the United Kingdom of Great Britain and Ireland was created	$17 \rightarrow 10$ (7 $\uparrow$)
Duke of Cumberland. of Wales, the eldest son and heir apparent of King **George II** and the father of King George III. He died without legitimate issue, when the dukedom again became extinct. This double dukedom, in the Peerage of Great Britain, was bestowed on Prince Ernest Augustus (1771–1851) (later King of Hanover), the fifth son and eighth child of King George III of the United Kingdom and King of Hanover. In 1919 it was suspended under the Titles Deprivation Act 1917 and has not been restored to its titular heir. A historic fixed bridge hand is known as the Duke of Cumberland	$67 \rightarrow 18$ (49 $\uparrow$)
George II of Great Britain. the Hanoverian quarter differenced overall by a label of three points argent. The crest included the single arched coronet of his rank. As king, he used the royal arms as used by his father undifferenced. Caroline's ten pregnancies resulted in eight live births. One of their children died in infancy, and seven lived to adulthood. **George II** of Great Britain **George II** (George Augustus; ; 30 October / 9 November 1683 – 25 October 1760) was King of Great Britain and Ireland, Duke of Brunswick-Lüneburg (Hanover) and a prince-elector of the Holy Roman Empire from 11 June 1727 (O.S.) until	$13 \rightarrow 19$ (6 $\downarrow$)

Figure 3: Reranking case 1. Blue squares indicate positive passages and white squares stand for negative passages. The titles of passages are bold and put in front of passages. These blue texts are the answers for the question.

Query: ***What kind of book is Where's Waldo?***

Positive Distribution of Initial Retrieval List

Positive Distribution of Reranked Retrieval List

1 6 11 16 21 26 31 36 41 46 50

Positive Passages	Rank Changes
Where's Waldo? (video game). Where's Waldo? (video game) Where's Waldo? (Where's Wally? in the UK, Australia and South America) is a **puzzle** game developed by Bethesda Softworks and published in 1991 by THQ for the Nintendo Entertainment System. It was the first video game loosely based on Martin Handford's book of the "same name". Mostly similar to the books, players must help Waldo get to the moon by finding him in each of the eight levels in the game. The game was panned by critics, who criticized the game for its graphics, which made it more difficult to find Waldo in each of the	24 → 1 (23 ↑)
Where's Waldo? (video game). takes advantage of superior pointer-based motion controls to easily locate search targets and supports versus multiplayer. Where's Waldo? (video game) Where's Waldo? (Where's Wally? in the UK, Australia and South America) is a **puzzle** game developed by Bethesda Softworks and published in 1991 by THQ for the Nintendo Entertainment System. It was the first video game loosely based on Martin Handford's book of the "same name". Mostly similar to the books, players must help Waldo get to the moon by finding him in each of the eight levels in the game. The game was panned by critics, who criticized the	31 → 5 (26 ↑)
Activity book. and does not fall neatly into one of these more specific categories. Activity books are typically centred around a particular theme. This may be a generic theme, e.g. dinosaurs, or based on a toy, television show, book, or game. For example, the Where's Wally? series of books (known as Where's Waldo? in the USA) by Martin Handford consists of both **puzzle** books, wherein the reader must search for characters hidden in pictures, and activity books such as "", which include a wider range of games and activities as well as **puzzles**. In 2018, Nintendo announced its intention to publish activity	28 → 6 (22 ↑)
Where's Waldo? The Fantastic Journey (video game). Where's Waldo? The Fantastic Journey (video game) Where's Waldo? The Fantastic Journey is a video game published by Ubisoft and developed by Ludia based on the book of the same name. It is a **puzzle** adventure game released for the Nintendo DS, Wii, Microsoft Windows, and the iPhone, and is also a remake of "The Great Waldo Search", released in 1992. Like the other games in the series, the object of the game is to search for hidden characters and items within a time limit. Hints are awarded to the player through Woof, Waldo's pet dog. Woof alerts the players	7 → 8 (1 ↓)
Where's Wally? was turned into a Sunday newspaper comic/**puzzle**, distributed by King Features Syndicate. The comics were also released in book form in the US, using the regional name 'Waldo'. In the early 1990s Quaker Life Cereal in the US carried various "Where's Wally?" scenes on the back of the boxes along with collector's cards, toys and send-away prizes. This was shown in "The Simpsons" episode "Hello Gutter, Hello Fadder" where Homer shouts "WALDO, WHERE ARE YOU?!" after looking at the scene on the cereal box as Waldo walks by the kitchen window. On 1 April 2018 Google Maps added a minigame	61 → 23 (38 ↑)

Figure 4: Reranking case 2.