On Complementarity Objectives for Hybrid Retrieval

Dohyeon Lee
Seoul National University
waylight3@snu.ac.kr

Seung-won Hwang*
Seoul National University
seungwonh@snu.ac.kr

Kyungjae Lee†
LG AI Research
kyungjae.lee@lgresearch.ai

Seungtaek Choi†
Riiid
seungtaek.choi@riiid.co

Sunghyun Park†
LG AI Research
sunghyun.park@lgresearch.ai

Abstract

Dense retrieval has shown promising results in various information retrieval tasks, and hybrid retrieval, combined with the strength of sparse retrieval, has also been actively studied. A key challenge in hybrid retrieval is to make sparse and dense complementary to each other. Existing models have focused on dense models to capture "residual" features neglected in the sparse models. Our key distinction is to show how this notion of residual complementarity is limited, and propose a new objective, denoted as RoC (Ratio of Complementarity), which captures a fuller notion of complementarity. We propose a two-level orthogonality designed to improve RoC, then show that the improved RoC of our model, in turn, improves the performance of hybrid retrieval. Our method outperforms all state-of-the-art methods on three representative IR benchmarks: MSMARCO-Passage, Natural Questions, and TREC Robust04, with statistical significance. Our finding is also consistent in various adversarial settings.

1 Introduction

Representing and matching queries and documents (or answers) is crucial for designing models for Information Retrieval (IR) and open-domain Question Answering (QA). Existing approaches have been categorized into **sparse** and **dense** retrieval.

Classic sparse (or symbolic) retrieval such as BM25 (Robertson and Zaragoza, 2009), quantifies the lexical overlaps (or exact matches) between query q and document d, weighted by term frequency (tf) and inverse document frequency (idf). Such computation can be efficiently localized to a few high-scoring q-d pairs with an inverted index, may fail to match pairs with term mismatches. For example, a text pair with identical intent—"facebook change password" and "fb modify passwd"—does not share any common word, so the pair cannot be matched by lexical retrieval.

To overcome such mismatches, dense retrieval models, such as BERT-based DPR (Karpukhin et al., 2020) or coCondenser (Gao and Callan, 2021), aim to support soft "semantic matching", by encoding queries and documents into low-dimensional embedding vectors. Dense representation is trained so that "password" and "passwd" are located close in the space even though they have different lexical representations.

These complementary advantages of each model have naturally motivated hybrid models (Gao et al., 2020; Yadav et al., 2020; Ma et al., 2021), which we denote as BM25+DPR, extracting scores from both models and selecting documents with the highest linearly combined scores.

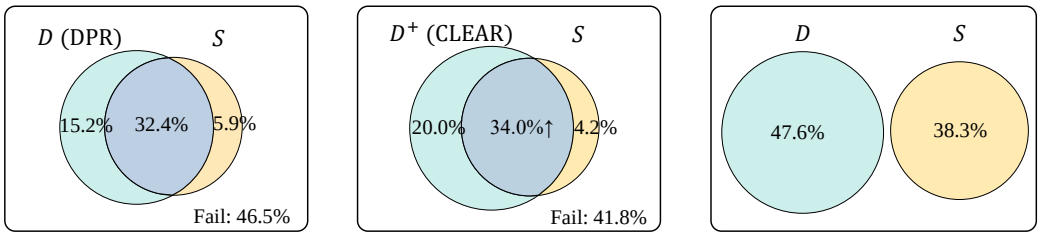
To illustrate how we advance BM25+DPR baseline, Figure 1(a) shows Recall@10 of BM25+DPR on Natural Questions, where a yellow circle, represents questions answerable by BM25, or **S**, and a blue circle, represents those answerable by DPR, or **D**. Desirably, two retrievers together should cover all questions in the universe **U**, but failure is 46.5%, which corresponds to $U - D \cup S$.

To improve, there are two directions: (1) enlarging $|D|$ and (2) making it more complementary to S. Figure 1(b) plots CLEAR (Gao et al., 2020), aiming to emphasize "residual" features neglected in sparse model, or, increase $|D - S|$. Though $|D - S|$ increased from 15.2% (Figure 1a) to 20.0% (Figure 1b), as intended, failure did not decrease significantly, from 46.5% (Figure 1a) to 41.8% (Figure 1b). We argue this decrease in failure cases, is confounded by enlarging $|D|$ from 47.6 (Figure 1a) to 54% (Figure 1b), by comparing with a hypothetical scenario keeping D fixed, but reducing failure cases significantly from 41.8% to 14.1% when the intersection is reduced.

Based on these observations, we propose a novel

*Corresponding Author
†Work done before joining current affiliation.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13357–13368



(a) D + S baseline (b) Hybrid Learning (CLEAR) (c) Hypothetical Scenario

Figure 1: Recall@10 on Natural Questions. In the venn diargram, (a) shows BM25+DPR baseline and (b) shows CLEAR using residual margin. (c) is a hypothetical scenario, identical to (a) but without the intersection

complementarity metric, considering the residual complementarity $|D - S|$, but relatively to $|D|$, denoted as Ratio of Complementarity (RoC). RoC is designed to be 1 when two models are disjoint (Figure 1c), and 0 when D is subsumed to S.

$$RoC = \frac{|D - S|}{|D|} = 1 - \frac{|D \cap S|}{|D|} \quad (1)$$

RoC has the following two advantages:

- RoC is backwardly compatible with existing residual complementary notions. We later derive (Section 3) that optimizing RoC can be divided into two sub-goals of increases $|D - S|$ and $|D \cap S|$, where the former is compatible with residual complementarity.

- RoC, when multiplied by $|D \cup S|$, directly approximates the number of questions that can be answered by hybrid models. As a result, increasing RoC as an objective, naturally correlates to the improved performance of a hybrid retriever.

With these advantages, we use our metric in Table 1 to explain the limitation of CLEAR building on residual complementarity alone. CLEAR increases $|D - S|$ as intended, but increases $|D \cap S|$ as its byproduct (see Figure 1b), due to their correlation. In contrast, we propose a simple but effective two-level orthogonality to resolve this correlation, and achieves both sub-goals, which significantly improves RoC. Table 1 shows that the improved RoC correlates to the improved recall as well *.

We verify that enhancement in RoC leads to improvement in hybrid retrieval performance on three IR datasets: MS MARCO, Natural Questions and TREC Robust04.

*This table is a motivational preview, and detailed setting and results can be found in Section 4.2

Model	Increase $\lvert D - S\rvert$	Decrease $\lvert D \cap S\rvert$	RoC	R@100 on NQ
CLEAR	✓		+0.05	+0.51
Ours	✓	✓	+0.12	+1.19

Table 1: Relative increases in two sub-goals, with respect to BM25+DPR (Figure 1a)

2 Related Work

2.1 Sparse and dense retrieval

Sparse (or symbolic) space is generally independent such that data structures, such as inverted index or bitmap, can efficiently identify matching candidates with exact matches, and ranking can also be efficiently computed. BM25 (Robertson and Zaragoza, 2009) is a well-known lexical ranking model using bag-of-words representation.

Meanwhile, dense retrieval models (Shen et al., 2014; Guo et al., 2016; Zhai et al., 2016; Nogueira and Cho, 2019; Zhan et al., 2020) have been proposed to tackle the term mismatch problem, which can be categorized as two groups (Guo et al., 2016): (1) embedding-based and (2) interaction-based models. Our target scenario is the former, representing query q and document d into two independent dense vectors and match q-d by using the vector similarity. However, we also discuss how our idea can apply to interaction-based ranking approaches (Nogueira and Cho, 2019; McDonald et al., 2018), capturing word-by-word interactions without vectors, which we discuss as non-embedding models in Section 3.3.

2.2 Complementarity

To leverage complementarity, there have been approaches to either combine the two spaces, or transfer knowledge from one space to another.

First, for combining, a naive approach is aggregating the scores from two spaces (Ma et al., 2021), which is advanced to a more sophisticated model,

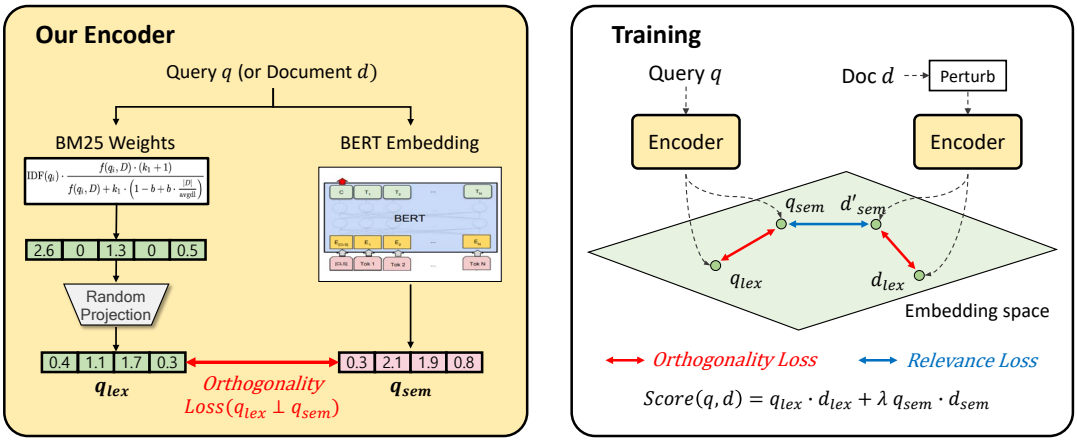


Figure 2: Architecture of our model for query-document matching. Red arrow indicates the proposed orthogonality loss and blue arrow indicates the relevance loss as query-document pairs.

such as CLEAR (Gao et al., 2020) learning by a residual margin of BM25. Specifically, when q-d pair is hard to match lexically, the margin becomes larger, then the loss for semantic matching is emphasized. In other words, the model is trained

Second, for transferring, SparTerm (Bai et al., 2020) learns sparse representations by distilling contextualized knowledge of BERT into bag-of-words space. Specifically, based on BERT encoders, SparTerm first produces a dense distribution of semantic importance for the vocabulary terms, then controls the activation of each term, ensuring the sparsity of the final representations. It means that, the representation capacity of term-based matching methods can be improved up to semantic-level matching. Ours falls into the first category that combines the two spaces, but we propose a new metric named RoC to evaluate complementarity in hybrid retrieval directly. Table 1 shows how existing complementarity metrics only partially cover RoC. Meanwhile, we also discuss how ours can be combined with the second category approaches, *i.e.*, SparTerm, to achieve further gains (Section 4.3).

3 Proposed Method

We propose RoC in Section 3.1 and then discuss how S (Section 3.2) and D (Section 3.3) are implemented. Section 3.4 discusses how D and S can be combined to optimize RoC.

3.1 Ratio of Complementarity (RoC)

RoC is a metric that can measure complementarity and directly approximates failure cases in Figure 1, i.e., $RoC \propto |U - Fail|$ where U indicates all answer documents. Based on this hypothesis, our goal is to maximize $|D \cup S| \cdot RoC$. We describe this goal into two sub-goals as follows:

$$
\begin{aligned}
&|D \cup S| \cdot RoC \\
&= |D \cup S| \cdot |D - S| \,/\, |D| \\
&= (|D - S| + |S|) \cdot |D - S| \,/\, |D| \\
&\simeq |D - S|^2 \,/\, |D| \quad {}^{*\dagger} \\
&= |D - S| \,/\, (1 + \frac{|D \cap S|}{|D - S|})
\end{aligned}
\tag{2}
$$

The first sub-goal is optimizing $|D \cap S|$, for which we separate the features captured by the sparse and dense models from each other. The second sub-goal is to maximize $|D - S|$, for which we propose to capture residual features of the sparse model. We describe how to achieve each sub-goal in Section 3.3.

3.2 Lexical Representation (S)

While any lexical retriever can be used, we describe our approach with BM25 (Robertson and Zaragoza, 2009) to construct symbolic representation (green vector q_{lex} in Figure 2(left)), to capture lexical matches. BM25 score can be written as an inner product between bag-of-words representations of the query and document. We define q and d representation from BM25 as q_{bm25} and $d_{\text{bm25}} \in \mathbb{R}^{|V|}$, respectively, where the i-th element of the representations q_{bm25} and d_{bm25} can be writ-

[†]since $|S|$ is a constant.

ten as follows:

$$q_{\text{bm25}}(i) = \begin{cases} \text{IDF}(q_i) & q_i \in q \\ 0 & q_i \notin q \end{cases}$$

$$d_{\text{bm25}}(i) = \frac{\text{TF}(d_i)(k_1+1)}{\text{TF}(d_i) + k_1 \cdot (1 - b + b\frac{|d|}{\text{avgdl}})},$$

where $\text{IDF}(\cdot)$ is inverse document frequency of term i, and $\text{TF}(\cdot)$ is frequency of term i in a given document. Thus, BM25(q, d) can be denoted as an inner product between q_{bm25} and d_{bm25}.

Given that the lexical representations have much larger dimensionality than semantic representations, it is required to compress q_{bm25} and d_{bm25} into a low-dimensional space. For such compression, random projection (Vempala, 2004) was found effective for preserving document ranking in Luan et al. (2020), which we adopt in this work. Though compression loss exists, this loss can be bounded by changing embedding dimension k. In our experiments on NQ, we follow the protocol from Luan et al. (2020), to set k as 715, which guarantees errors to be lower than 0.038, for the 768 dimension BERT embedding. Random projection is a linear transformation via matrix A, and each element of the matrix $A \in \mathbb{R}^{768\times|V|}$ is randomly sampled from a Rademacher distribution with equal probability from the two values: $\{-\frac{1}{\sqrt{768}}, \frac{1}{\sqrt{768}}\}$. The final lexical representation, q_{lex} and d_{lex}, can be obtained as follows:

$$q_{\text{lex}} = A \cdot q_{\text{bm25}}, \quad d_{\text{lex}} = A \cdot d_{\text{bm25}} \tag{3}$$

With Eq. (3), q_{lex} and d_{lex} are in the same dimensional space as semantic vectors from BERT, while preserving the ranking. In addition, our goal is to enforce complementarity with the semantic vectors in Section 3.3.

From lexical representations, the final relevance score between query q and document d is calculated by an inner product, as follows:

$$\text{Score}_{\text{lex}}(q, d) = q_{\text{lex}} \cdot d_{\text{lex}} \tag{4}$$

This relevance score is approximated to BM25 score, and at the same time, we can handle the two vectors, q_{lex} and d_{lex}, in semantic space.

3.3 Semantic Representation (D)

For semantic representation (pink vectors in Figure 2(left)), we adopt a state-of-the-art (coCondenser; (Gao and Callan, 2021)) for explanation purposes, consisting of a dual-encoder structure based on BERT (Devlin et al., 2019). Thus, our dense retrieval follows BERT's architecture, settings, and hyper-parameters. Following BERT's input style, we apply wordpiece tokenizer to the input document and query, and then add a [CLS] token at the beginning and a [SEP] token at the end, as follows:

$$\text{Input}(\cdot) = \text{[CLS] Tokenizer}(\cdot) \text{ [SEP]} \tag{5}$$

Then, we take the embeddings of queries and documents, from the representation of BERT at [CLS] token. The semantic representations of q and d can be formulated as follows:

$$\begin{aligned} h_q &= \text{BERT}(\text{Input}(q)) \in \mathbb{R}^{|q|\times 768} \\ h_d &= \text{BERT}(\text{Input}(d)) \in \mathbb{R}^{|d|\times 768} \\ q_{\text{sem}} &= \text{Pool}(h_q), \quad d_{\text{sem}} = \text{Pool}(h_d) \in \mathbb{R}^{768} \end{aligned} \tag{6}$$

where $\text{Pool}(\cdot)$ indicates [CLS] pooling extracting the first vector over the hidden states h. Their semantic relevance $\text{Score}_{\text{sem}}$ is calculated by an inner product of q_{sem} and d_{sem}: $\text{Score}_{\text{sem}} = q_{\text{sem}} \cdot d_{\text{sem}}$.

The training loss for DPR is the negative log likelihood of the positive passage:

$$\mathcal{L}_{\text{rel}} = -\log \frac{e^{\text{Score}_{\text{sem}}(q,d^+)}}{e^{\text{Score}_{\text{sem}}(q,d^+)} + \sum_{d^-} e^{\text{Score}_{\text{sem}}(q,d^-)}}, \tag{7}$$

where d^+ and d^- indicate positive and negative documents corresponding to given query q. For selecting the negative documents, we follow the convention in previous works (Karpukhin et al., 2020; Sachan et al., 2021; Gao et al., 2020), *i.e.*, hard negative sampling, of selecting top-ranked documents retrieved from BM25 that do not contain the answer.

3.4 Complementarity Objective

We propose embedding-level and input-level orthogonality constraints, which decrease $|D \cap S|$ and increase $|D - S|$, respectively.

Embedding-level Orthogonality To reduce $|D \cap S|$, we separate the features between the semantic and lexical representation spaces. Specifically, we enforce orthogonality between lexical and semantic representations (*i.e.*, $q_{lex} \perp q_{sem}$). While training BERT by Eq. (7), we impose an additional constraint using cosine similarity, which normalizes

the features, to constrain the direction of the two vectors (lex and sem). We define the loss function of the orthogonality, as follows:

$$\mathcal{L}_{\text{ortho}} = \left(\frac{\langle q_{\text{lex}}, q_{\text{sem}} \rangle}{\|q_{\text{lex}}\| \|q_{\text{sem}}\|} \right)^2 + \left(\frac{\langle d_{\text{lex}}, d_{\text{sem}} \rangle}{\|d_{\text{lex}}\| \|d_{\text{sem}}\|} \right)^2 \tag{8}$$

where $\langle \cdot, \cdot \rangle$ is an inner product. If the two vectors are perfectly perpendicular to each other, the loss is equal to 0; otherwise, it has a positive value. This is compatible with our goal of minimizing common features of the two vectors, resulting in the reduced overlap $|D \cap S|$ of the semantic (D) and lexical (S) model, as in Figure 1(c). Adding this orthogonality loss, the final loss function for BERT-ranker is computed as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rel}} + \mathcal{L}_{\text{ortho}} \tag{9}$$

While we can tune the above aggregation, we empirically found 1:1 aggregation was effective.

Input-level Orthogonality For increasing $|D - S|$ in the input-level, we follow the convention of using residual features, neglected by the sparse model, such as the synonymy between mismatched terms, e.g., "password = passwd". For this objective, we propose a method to perturb a subset of matched tokens for learning mismatched terms, similar to a denoising autoencoder (Hill et al., 2016). In the denoising autoencoder, input text is corrupted by random noise function, then the decoder is trained to recover the original text, learning robust features on variances (Vincent et al., 2008). In contrast, while our token perturbation does not have the recovering decoder, our distinction is corrupting exact matches, focusing on soft matching. Given a q-d pair, we denote a set of exact matched tokens in d as $X_{EM} = \{x_i | x_i \in q \text{ and } x_i \in d\}$. Through random sampling, we replace tokens in X_{em} with the [MASK] token and feed the new sequence d' into BERT. By the token perturbation, we modify d_{sem} in Eq. (6) to d'_{sem}, computed as follows:

$$\begin{aligned} d' &= d \backslash \{\text{Sample}(X_{EM})\} \\ d'_{\text{sem}} &= \text{Pool}(\text{BERT}(\text{Input}(d'))) \end{aligned} \tag{10}$$

where $\text{Sample}(\cdot)$ is random sampling of tokens.[‡] This perturbation is applied for only training process and we do not use this at inference time. This perturbation enables to apply disentanglement ideas, not only to new models, but diverse ranges of existing models (See Section 4.1).

[‡]We sample 15% tokens in X_{em} and this ratio was decided empirically on the dev set.

	MS MARCO	Natural Questions	TREC Robust04
Total # Queries	808K (train) 6.9K (test)	58K (train) 3.6K (test)	200 (train) 50 (test)
Total # Doc	8.8M	250K	528K
Avg Query Length	5.9	9.4	2.7
Avg Doc Length	56.2	91.1	261.0

Table 2: Statistics of three datasets.

Final Relevance Score For aggregating the scores from the two IR models, we follow the convention of major baselines (Karpukhin et al., 2020; Gao et al., 2020; Ma et al., 2021), using a linear combination.

$$\begin{aligned} \text{Score}_{\text{dual}}(q, d) &= \text{Score}_{\text{lex}}(q, d) + \lambda \text{Score}_{\text{sem}}(q, d) \\ &= q_{\text{lex}} \cdot d_{\text{lex}} + \lambda q_{\text{sem}} \cdot d_{\text{sem}} \end{aligned} \tag{11}$$

where λ is the hyper-parameter controlling the weight for the different scales.

4 Experiment

In this section, we describe experimental setting and formulate our research questions to guide our experiments.

4.1 Experimental Setting

Dataset To validate the effectiveness of our method, we conduct query-passage (or, query-document) matching for the following three datasets, which are widely used and statistically diverse (Table 2) as well:

- **MS MARCO-Passage**[§] (Nguyen et al., 2016): This benchmark provides 8.8 million passages, and labels are obtained from the top-10 results retrieved by the Bing search engine. As the relevance labels for the official test set are not publicly available, we evaluate the development set only. We use MRR@10 and R@100 to evaluate the performance for full-ranking retrieval.
- **Natural Questions**[¶] (Kwiatkowski et al., 2019): In this dataset, we aim to find relevant passages that answer the given question from total 250K

[§]https://github.com/microsoft/MSMARCO-Passage-Ranking

[¶]https://ai.google.com/research/NaturalQuestions/download

Model	MS MARCO			Natural Questions		TREC Robust04	
	MRR @10	MAP	R @100	MAP	R @100	MAP	nDCG @20
Reported results							
DPR (Karpukhin et al., 2020)	31.1*	-	-	-	85.4†	-	-
DPR+PAQ (Oğuz et al., 2021)	31.4	-	-	-	88.6	-	-
POSIT-DRMM+MV (McDonald et al., 2018)	-	-	-	-	-	27.0	46.1
CLEAR: DPR+BM25 (Gao et al., 2020)	33.8	-	-	-	-		
COCONDENSER (Gao and Callan, 2021)	38.2	-	-	-	89.0	-	-
Re-implemented Baselines							
(1) SPARTERM	27.94	24.62	72.48	24.54	71.68	19.86	33.48
(2) BM25	19.25	19.57	69.54	26.59	73.70	25.64	41.95
(3) DPR	29.20	25.83	71.42	33.08	85.38	33.36	48.76
(4) DPR + BM25 (Naive sum)	33.75	29.34	77.34	33.56	86.77	33.65	49.32
(5) COCONDENSER	38.19	31.41	80.53	34.32	89.03	34.14	52.29
(6) COCONDENSER + BM25 (Naive sum)	37.85	31.82	80.76	34.47	88.94	34.53	53.06
(7) CLEAR: DPR+BM25	33.46	28.64	77.68	33.17	87.23	34.49	51.63
(8) CLEAR: COCONDENSER+BM25	37.99	32.08	80.42	34.93	89.45[6]	35.86[6]	52.78
OURS: DPR+BM25	34.62	29.27	78.75	34.15	87.89	36.43	53.27
OURS: COCONDENSER+BM25	**38.63**[68]	**32.33**[6]	**80.84**[8]	**35.97**[68]	**90.13**[68]	**36.74**[68]	**53.39**

Table 3: Results of the different models on MS MARCO, Natural Questions, and TREC Robust04 datasets. Best performing results are showin in **bold**. In *Reported results*, we copy the numbers from * (Xiong et al., 2020), † (Karpukhin et al., 2020), and a dash ("-") indicates the baseline methods did not report scores. [6] and [8] indicates the p-value < 0.05 when the result is compared with baseline (6) and (8) with Bonferroni correction.

passages, and labels are minded from spans in Wikipedia articles identified by annotators. Following DPR (Karpukhin et al., 2020), we consider the passages including answers as relevant passages at evaluation regarding R@k.

- **TREC Robust04**[‖] (Voorhees et al., 2005): This dataset contains 250 topic queries and 528K documents. As there is no official train/test split published in Robust04, we follow the split setting provided in McDonald et al. (2018) using 5-fold cross-validation.

We honor the metrics used in the original work, which explains different metrics for different datasets. Robust04 is widely used but small in size, so we also follow the convention of studying MS MARCO and Natural Questions with larger sizes.

Implementation For DPR encoder, we use a base version (Uncased) of BERT (Devlin et al., 2019). For training, we set batch size 10 and use Adam (Kingma and Ba, 2015) optimizer with learning rate 0.0002. For stable training, we used gradient clipping (Pascanu et al., 2013) with norm 1.0, and we halve the learning rate for every epoch after 3 epochs of training iteration. We follow DPR (Karpukhin et al., 2020) for the other training details such as hard negative sampling.

As hyper-parameters, we automatically found the best values for λ, based on MAP on development set, where we search λ in a range of $[0, 2]$ with 0.1 step size. The best configuration for λ was 1.5, 1.3 and 2.0 on MARCO-Passage, Natural Questions and Robust04, respectively.

Evaluation Metric For task evaluation, we compute the following metrics and report average performance: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (nDCG), and Recall at top-k ranks (R@k). For Recall, we follow the previous work (Karpukhin et al., 2020), which is computed as the proportion of questions to which the top-k retrieved passages contain answers. For MAP and nDCG, we use the latest TREC evaluation script[**] to compute these metrics. Results of the p-value < 0.05 on the t-test with Bonferroni correction are displayed in bold in Table 3.

Baselines We compare our model with the following baselines which are state-of-the-art retrievals. We use SPARTERM (Bai et al., 2020), COCONDENSER (Gao and Callan, 2021), and BM25

[‖] https://github.com/nlpaueb/deep-relevance-ranking

[**] https://trec.nist.gov/trec_eval/

Model	NQ	Robust04	
	R@20	MAP	nDCG
(5) DPR+BM25	78.92	33.65	49.32
+ Emb-level Ortho (E)	80.87 (+1.95)	35.54 (+1.89)	52.19 (+2.87)
+ Input-level Ortho (I)	79.24 (+0.32)	34.08 (+0.43)	51.42 (+2.10)
OURS:DPR+BM25 (I + E)	81.28 (+2.36)	36.43 (+2.78)	53.27 (+3.95)

Table 4: Ablation study. The number inside the parenthesis indicates the increase from the baseline model.

as our baselines. SPARTERM is a term-based retrieval model using BERT, and gives lexical matching score. BM25 is a well-known lexical matching method using TF and IDF. For BM25, we use Pyserini[††] open-source implementation. For coCondenser, we use open-source implementation[‡‡] to reproduce. On the other hand, we use hybrid space baselines such as COCONDENSER+BM25 (Naive sum) and CLEAR (Gao et al., 2020). Both methods give similarity score by merging the scores of sparse and dense model. Note our implementation of CLEAR performs better than their published results, as we update its base transformer with coCondenser. For fair comparison, both ours and CLEAR build upon the same coCondenser implementation.

4.2 Experimental Results

Research Questions To evaluate the effectiveness of our method, we address the following research questions:

- RQ1: Does the two-level orthogonality improves the RoC?
- RQ2: Does the improved RoC contribute to better complementarity?
- RQ3: Does the improved complementarity improve hybrid retrieval?

4.2.1 RQ1: Effectiveness of Orthogonality

Model	RoC	MAP on NQ
COCONDENSER	0.32	34.47
+ Emb-level Ortho (E)	0.42	34.94
+ Input-level Ortho (I)	0.38	35.62
+ I and E	0.47	35.97

Table 5: Effect of orthogonality objectives on ROC.

[††] https://github.com/castorini/pyserini
[‡‡] https://github.com/luyug/Condenser

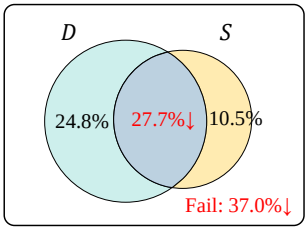


Figure 3: Recall@10 of our proposed hybrid method on Natural Questions compared with Figure 1.

We conduct an ablation study to confirm whether the two orthogonality constraints contribute to improving RoC. As Table 5 shows, embedding-level orthogonality improves RoC by 0.1 compared to naive sum and input-level orthogonality improves by 0.06. Applying both of these improves RoC by 0.15, which is a significant improvement compared to CLEAR, while CLEAR improves by only 0.05 from the naive sum.

4.2.2 RQ2: Improved Complementarity with RoC

In this section, we show our method improves complementarity, by using recall of hybrid retrieval, and also adversarial evaluation.

First, ours (Figure 3) shows 0.15 higher RoC than CLEAR because of the improvement in RoC and failure cases are reduced compared to CLEAR (Figure 1b). In other words, RoC is a more reliable predictor of $|D \cup S|$, which directly correlates to performance.

Second, we can also observe complementarity in **adverse scenarios**. We categorize queries into two groups, BM25-Easy and BM25-Hard, following the convention of (Wei and Zou, 2019). Specifically, we define easy and hard set, by sorting MRR@10 scores of BM25 for all queries. Top 50% is BM25-Easy, where BM25 alone is already competitive, and the rest is BM25-Hard, which is adverse for lexical retrievers.

Desirably, we expect a hybrid model to outperform BM25 ranking in the hard set. With this expectation, on Figure 4, we compare the ratio that the hybrid model provides better ranking (with respect to MRR@10). Surprisingly, CLEAR does not improve such ratio of DPR much (+1.5%), which is consistent with the results in Figure 5. In contrast, we significantly improve the ratio by +10.2%.

Alternative way to observe adverse scenarios, is to build an adverse dataset with less matched terms. Specifically, lexically matched terms be-

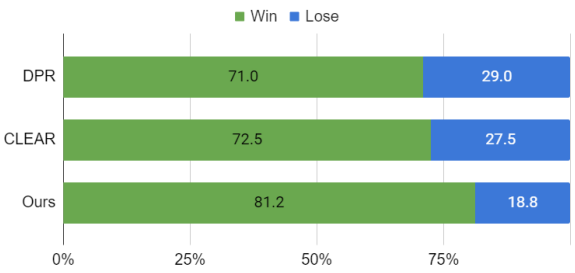


Figure 4: Comparative performances on BM25-Hard, queries of the lower 50% based on MRR@10 of BM25.

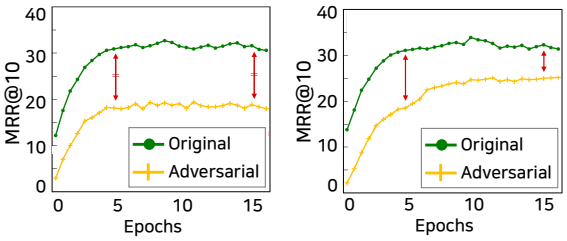


Figure 5: MRR@10 scores on original and adversarial MS MARCO development, of CLEAR (left) and **Ours (right)**.

tween query and document are replaced by their synonyms (as similarly done in (Wei and Zou, 2019)). Figure 5 compares how CLEAR and ours generalize to this adverse set. Both models on the original dataset improve MRR rapidly in early epochs, while not so much on adverse set until epoch 5. However, this gap decreases only in ours (Figure 5b), while stays constant in CLEAR (Figure 5a), showing CLEAR continues to focus on lexical matching, while we learn to leverage semantic matching.

4.2.3 RQ3: Effect of Complementarity in Hybrid Retrieval

In this experiment, we verify the effect of enhanced complementarity on various performance aspects.

We first compare the performance of the hybrid model and ours to show that our two-level orthogonality improves the hybrid retrieval performance as well as the complementarity between the two models. In Table 3, when compared with coCondenser+BM25 (Naive sum), our method improved MRR@10 by 0.78 on MSMARCO, and MAP by 1.50 & 2.21 on NQ & Robust04, respectively, showing the complementarity improves document ranking. When compared with the state-of-the-art model, CLEAR, our method achieved 0.64 gains of MRR@10 on MSMARCO, and 1.04 & 0.88 gains of MAP on NQ & Robust04, respectively. Note that our method has a statistically significant performance improvement, as indicated by superscripts in Table 3.

Model	NQ	Robust04	
	MAP	MAP	nDCG
SPARTERM	26.33	19.86	45.35
+ Input-level Ortho	**26.78** (+0.45)	**20.62** (+0.76)	**46.09** (+0.74)
INTERACTIVE BERT	41.76	30.46	49.63
+ Input-level Ortho	**42.20** (+0.44)	**30.99** (+0.53)	**49.94** (+0.31)

Table 6: Results for non-embedding models.

Ablation Study of Embedding- and Input-level Approaches To investigate the isolated effect of two-level orthogonality on hybrid retrieval performance, we conducted an ablation study in NQ and Robust04 as shown in Table 4. For this, we add each component (embedding- or input-level objective) to the baseline model: DPR+BM25 (Naive sum). In both datasets, we observe that the embedding- and input-level methods can achieve significant improvements over the baseline, showing that the enhanced complementarity improves hybrid retrieval performance. Note that the embedding-level objective is more effective than the input-level objective, which is consistent with the complementarity improvement result in Table 5. We can also see in Table 6 that the input-level objective works even for non-embedding models.

Length Generalizability Based on the well-known weakness of BERT showing low accuracy on long documents in the NQ dataset (Luan et al., 2020), we verify the effect of improved complementarity on robustness for long documents. Our proposed model outperforms CLEAR and obtains the best scores over all the lengths except one group. This shows that complementarity plays an essential role in length generalization. Results and details are described in Section A.1.

5 Conclusion

We study the problem of hybrid retrieval, where existing state-of-the-arts have pursued a partial notion of complementarity. In contrast, we propose RoC, a metric that captures a fuller notion of the complementarity between sparse and dense models. We then propose a simple but effective two-level orthogonality objective to enhance RoC and verify that optimizing RoC enhances both com-

plementarity and retrieval, leads to outperforming state-of-the-arts in three representative IR benchmarks, MSMARCO-Passage, Natural Questions, and TREC Robust04, and generalizing to adversarial settings.

6 Limitations

We make use of MS-MARCO, a resource that provides large-scale relevance annotations. However, as with most retrieval datasets, this dataset could contain annotation biases. Given the vast number of documents in the corpus supplied by the dataset, relevance annotations are sparsely distributed, with all other documents assumed to be non-relevant. Consequently, some relevant documents may be inaccurately labeled as non-relevant, leading to false negatives. A notable annotation bias in MS-MARCO is that the relevant label correlates highly with the exact matching term (Xiong et al., 2020). This bias poses a limitation during the training or evaluation stages. To appropriately address this annotation bias, we might need to reorganize the labeling process using either a human or a neural annotator, or we could aim to design and train a model that is resilient to such bias. We reserve this task for future research efforts.

Acknowledgements

This work was supported by the SNU-NAVER Hyperscale AI Center and MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01789) and grants [NO.2021-0-0268, AI Hub, SNU], [No.2022-0-00077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data], and [NO.2021-0-01343, AI Graduate School].

References

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. In *arXiv preprint*, page abs/2010.00768.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. 2020. Complementing lexical retrieval with semantic residual embedding. *arXiv preprint arXiv:2004.13969*.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Rhinehart, 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. In *Transactions of the Association of Computational Linguistics*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. In *arXiv preprint*, page abs/2005.00181.

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

Ryan McDonald, George Brokos, George Brokos, and Ion Androutsopoulos. 2018. Deep relevance ranking using enhanced document-query interactions. In *EMNLP*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. Domain-matched pretraining tasks for dense retrieval. *arXiv preprint arXiv:2107.13602*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. *arXiv preprint arXiv:2101.00408*.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM.

Santosh Vempala. 2004. The random projection method, volume 65 of dimacs series in discrete mathematics and theoretical computer science. *American Mathematical Society*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Ellen M Voorhees et al. 2005. Overview of the trec 2005 robust retrieval track. In *Trec*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Having your cake and eating it too: Training neural retrieval for language inference without losing lexical match. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1625–1628.

Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. 2016. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1295–1304. ACM.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. Repbert: Contextualized text embeddings for first-stage retrieval. In *arXiv preprint*, page abs/2006.15498.

A Appendices

A.1 Length Generalizability

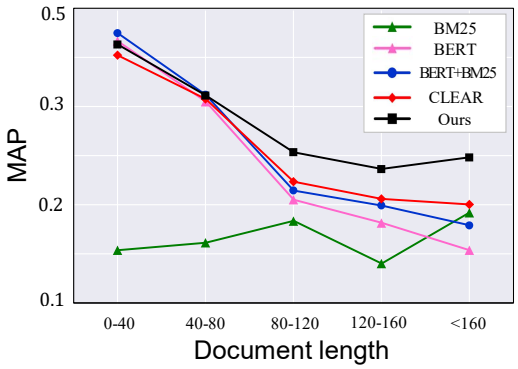


Figure 6: Graph shows MAP of various models by document lengths in NQ dataset.

As shown in Figure 6, we group test set by the length of target documents (per 40 tokens), and report MAP score per each group. From the results, we can confirm the reported weakness in long documents– Precision of DPR decreases as the document length increases, while that of BM25 stays consistent. Meanwhile, hybrid models including both CLEAR and ours show better robustness than DPR and BM25 over the longer documents. Our proposed model outperforms CLEAR and obtains the best scores over all the lengths except a group "0-40". This shows that complementarity plays an essential role in length generalization.

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?
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?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
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)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
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?
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.?
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)?
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?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.