Complement Lexical Retrieval Model with Semantic Residual Embeddings

Luyu Gao[1(✉)], Zhuyun Dai[1(✉)], Tongfei Chen[2(✉)], Zhen Fan[1(✉)], Benjamin Van Durme[2(✉)], and Jamie Callan[1(✉)]

[1] Carnegie Mellon University, Pittsburgh, USA
{luyug,zhuyund,zhenfan,callan}@cs.cmu.edu
[2] Johns Hopkins University, Baltimore, USA
{tongfei,vandurme}@cs.jhu.edu

Abstract. This paper presents CLEAR, a retrieval model that seeks to complement classical lexical exact-match models such as BM25 with semantic matching signals from a neural embedding matching model. CLEAR explicitly trains the neural embedding to encode language structures and semantics that lexical retrieval fails to capture with a novel residual-based embedding learning method. Empirical evaluations demonstrate the advantages of CLEAR over state-of-the-art retrieval models, and that it can substantially improve the end-to-end accuracy and efficiency of reranking pipelines.

1 Introduction

State-of-the-art search engines adopt a multi-stage retrieval pipeline system: an efficient first-stage *retriever* uses a query to fetch a set of documents from the entire document collection, and subsequently one or more *rerankers* refine the ranking [28]. The retriever needs to run fast with high efficiency in order to scan through the entire corpus with low latency. As a result, retrievers have remained simple and give only mediocre performance. With recent deep neural models like BERT [10] rerankers pushing reranking accuracy to new levels, first-stage retrievers are gradually becoming the bottleneck in modern search engines.

Typical first-stage retrievers adopt a bag-of-words retrieval model that computes the relevance score based on heuristics defined over the *exact word overlap* between queries and documents. Models such as BM25 [32] remained state-of-the-art for decades and are still widely used today. Though successful, lexical retrieval struggles when matching goes beyond surface forms and fails when query and document mention the same concept using different words (*vocabulary mismatch*), or share only high-level similarities in topics or language styles.

An alternative approach for first-stage retrieval is a neural-based, dense embedding retrieval: query words are mapped into a single vector query representation to search against document vectors. Such methods learn an inner product space where retrieval can be done efficiently leveraging recent advances in maximum inner product search (MIPS) [12,15,34]. Instead of heuristics, embedding

D. Hiemstra et al. (Eds.): ECIR 2021, LNCS 12656, pp. 146–160, 2021.
https://doi.org/10.1007/978-3-030-72113-8_10

retrieval learns an encoder to understand and encode queries and documents, and the encoded vectors can softly match beyond text surface form. However, single vector representations have limited capacity [1], and are unable to produce granular token-level matching signals that are critical to accurate retrieval [11,33].

We desire a model that can capture both token-level and semantic-level information for matching. We propose a novel first-stage retrieval model, *Complementary Retrieval Model (*CLEAR *)*, that uses dense embedding retrieval to *complement* exact lexical retrieval. CLEAR adopts a single-stage-multi-retriever design consisting of a lexical retrieval model based on BM25 and an embedding retrieval model based on a Siamese framework that uses BERT [10] to generate query/document embedding representations. Importantly, unlike existing techniques that train embeddings directly for ranking independently [4,40], CLEAR explicitly trains the embedding retrieval model with a *residual* method: the embedding model is *trained* to build upon the lexical model's exact matching signals and to fix the mistakes made by the lexical model by supplementing semantic level information, effectively learning semantic matching not captured by the lexical model, which we term the un-captured residual.

Our experiments on large-scale retrieval data sets show the substantial and consistent advantages of CLEAR over state-of-the-art lexical retrieval models, a strong BERT-based embedding-only retrieval model, and a fusion of the two. Furthermore, CLEAR's initial retrieval provides additive gains to downstream rerankers, improving end-to-end accuracy and efficiency. Our qualitative analysis reveals promising improvements as well as new challenges brought by CLEAR.

2 Related Work

Traditionally, first-stage retrieval has relied on bag-of-words models such as BM25 [32] or query likelihood [19], and has augmented text representations with n-grams [25], controlled vocabularies [30], and query expansion [20]. Bag-of-words representations can be improved with machine learning techniques, e.g., by employing machine-learned query expansion on bag-of-sparse-features [5,39], adjusting terms' weights [8] with BERT [10], or adding terms to the document with sequence-to-sequence models [29]. However, these approaches still use the lexical retrieval framework and may fail to match at a higher semantic level.

Neural models excel at semantic matching with the use of dense text representations. Neural models for IR can be classified into two groups [11]: *interaction-based* and *representation-based* models. Interaction-based models model interactions between word pairs in queries and documents. Such approaches are effective for reranking, but are cost-prohibitive for first-stage retrieval as the expensive document-query interactions must be computed online for all ranked documents.

Representation-based models learn a single vector representation for the query or the document and use a simple scoring function (e.g., cosine or dot product) to measure their relevance. Representation-based neural retrieval models can be traced back to efforts such as LSI [9], Siamese networks [2], and MatchPlus [3]. Recent research investigated using modern deep learning techniques to

build vector representations: [21] and [13] used BERT-based retrieval to find passages for QA; [4] proposes a set of pre-training tasks for sentence retrieval. Representation-based models enable low-latency, full-collection retrieval with a dense index. By representing queries and documents with dense vectors, retrieval is reduced to a maximum inner product search (MIPS) [34] problem. In recent years, there has been increasing effort on accelerating maximum inner product and nearest neighbor search, which led to high-quality implementations of libraries for nearest neighbor search such as hnsw [24], FAISS [15], and SCaNN [12]. Notably, with these technologies, nearest neighbor search can now scale to millions of candidates with millisecond latency [12,15], and has been successfully used in large-scale retrieval tasks [13,21]. They provide the technical foundation for fast embedding retrieval of our proposed CLEAR model.

The effectiveness of representation-based neural retrieval models for standard ad-hoc search is mixed [11,40]. All of the representation-based neural retrieval models share the same limitation – they use a fixed number of dimensions, which incurs the specificity vs. exhaustiveness trade-off as in all controlled vocabularies [33]. Most prior research on hybrid models has focused on the reranking stage [26]. Some very recent research begins to explore hybrid lexical/embedding models. Its focus is mainly on improving the embedding part with weak-supervision [18] for low-resource setups, or new neural architectures that use multiple embedding vectors to raise model capacity [23]. In these works, embedding models are all trained independently from the lexical models and rely on simple post-training fusion to form a hybrid score. To the best of our knowledge, ours is the first work that investigates jointly training latent embeddings and lexical retrieval for first-stage ad hoc retrieval.

3 Proposed Method

CLEAR consists of a lexical retrieval model and an embedding retrieval model. Between these two models, one's weakness is the other's strength: lexical retrieval performs exact token matching but cannot handle vocabulary mismatch; meanwhile, the embedding retrieval supports semantic matching but loses granular (lexical level) information. To ensure that the two types of models work together and fix each other's weakness, we propose a *residual*-based learning framework that teaches the neural embeddings to be complementary to the lexical retrieval.

3.1 Lexical Retrieval Model

Lexical retrievers are designed to capture token level matching information. They heuristically combine token overlap information, from which they compute a matching score for query document pairs. Decades of research have produced many lexical algorithms such as vector space models, Okapi BM25 [32], and query likelihood [19]. We use BM25 [32] given its popularity in existing systems.

Given a query q and document d, BM25 generates a score based on the overlapping words statistics between the pair.

$$s_{\text{lex}}(q,d) = \text{BM25}(q,d) = \sum_{t \in q \cap d} \text{rsj}_t \cdot \frac{\text{tf}_{t,d}}{\text{tf}_{t,d} + k_1 \left\{(1-b) + b\frac{|d|}{l}\right\}}. \tag{1}$$

t is a term, $tf_{t,d}$ is t's frequency in document d, rsj_t is t's Robertson-Spärck Jones weight, and l is the average document length. k_1 and b are parameters.

3.2 Embedding Retrieval Model

The embedding retrieval model encodes either the query or document text sequence into a dense embedding vector, and matches queries and documents softly by comparing their vector similarity. Generally, the embedding retrieval model can take various neural architectures that encode natural language sequences such as CNN [16], or LSTM [14], as long as the model outputs can be pooled effectively into a single fixed-length vector for any input. A model capable of deeper text understanding is usually desired to produce high-quality embedding.

This work uses a Transformer [35] encoder. We start with pretrained BERT [10] weights and fine-tune the model to encode both queries and documents into vectors in a d-dimension embedding space, i.e., $\mathbf{v}_q, \mathbf{v}_d \in \mathbb{R}^d$. The model has a Siamese structure, where the query and document BERT models share parameters θ in order to reduce training time, memory footprint, and storthe special token $\langle \text{QRY} \rangle$ to queries and $\langle \text{DOC} \rangle$ to documents. For a given query or document, the embedding model computes the corresponding query vector $\mathbf{v}_q$ or document vector $\mathbf{v}_d$, following SentenceBERT [31], by average pooling representations from the encoder's last layers.

$$\mathbf{v}_q = \text{AvgPool}[\text{BERT}_\theta(\langle \text{QRY} \rangle \ ; \ \text{query})] \tag{2}$$

$$\mathbf{v}_d = \text{AvgPool}[\text{BERT}_\theta(\langle \text{DOC} \rangle \ ; \ \text{document})] \tag{3}$$

The embedding matching score $s_{\text{emb}}(q,d)$ is the dot product of the two vectors. We use dot product as the similarity metric as it allows us to use MIPS [12,15] for efficient first-stage retrieval.

$$s_{\text{emb}}(q,d) = \mathbf{v}_q^{\text{T}} \ \mathbf{v}_d \ . \tag{4}$$

3.3 Residual-Based Learning

We propose a novel residual-based learning framework to ensure that the lexical retrieval model and the embedding retrieval model work well together. While BM25 has just two trainable parameters, the embedding model has more flexibility. To make the best use of the embedding model, we must avoid the embedding model "relearning" signals already captured by the lexical model. Instead, we focus its capacity on semantic level matching missing in the lexical model.

In general, the neural embedding model training uses hinge loss [36] defined over a triplet: a query q, a relevant document d^+, and an irrelevant document d^- serving as a negative example:

$$\mathcal{L} = [m - s_{\text{emb}}(q, d^+) + s_{\text{emb}}(q, d^-)]_+ \tag{5}$$

where $[x]_+ = \max\{0, x\}$, and m is a static loss margin. In order to train embeddings that complement lexical retrieval, we propose two techniques: sampling negative examples d^- from lexical retrieval errors, and replacing static margin m with a variable margin that conditions on the lexical retrieval's residuals.

Error-Based Negative Sampling. We sample negative examples (d^- in Eq. 5) from those documents mistakenly retrieved by lexical retrieval. Given a positive query-document pair, we uniformly sample irrelevant examples from the top N documents returned by lexical retrieval with probability p. With such negative samples, the embedding model learns to differentiate relevant documents from confusing ones that are lexically similar to the query but semantically irrelevant.

Residual-Based Margin. Intuitively, different query-document pairs require different levels of extra semantic information for matching on top of exact matching signals. Only when lexical matching fails will the semantic matching signal be necessary. Our negative sampling strategy does not tell the neural model the degree of error made by the lexical retrieval that it needs to fix. To address this challenge, we propose a new residual margin. In particular, in the hinge loss, the conventional static constant margin m is replaced by a linear residual margin function m_r, defined over $\text{s}_{\text{lex}}(q, d^+)$ and $\text{s}_{\text{lex}}(q, d^-)$, the lexical retrieval scores:

$$m_r(\text{s}_{\text{lex}}(q, d^+), \text{s}_{\text{lex}}(q, d^-)) = \xi - \lambda_{\text{train}}(\text{s}_{\text{lex}}(q, d^+) - \text{s}_{\text{lex}}(q, d^-)), \tag{6}$$

where ξ is a constant non-negative bias term. The difference $\text{s}_{\text{lex}}(q, d^+) - \text{s}_{\text{lex}}(q, d^-)$ corresponds to a residual of the lexical retrieval. We use a scaling factor λ_{train} to adjust the contribution of residual. Consequently, the full loss becomes a function of both lexical and embedding scores computed on the triplet,

$$\mathcal{L} = [m_r(\text{s}_{\text{lex}}(q, d^+), \text{s}_{\text{lex}}(q, d^-)) - s_{\text{emb}}(q, d^+) + s_{\text{emb}}(q, d^-)]_+ \tag{7}$$

For pairs where the lexical retrieval model already gives an effective document ranking, the residual margin m_r (Eq. 6) becomes small or even becomes negative. In such situations, the neural embedding model makes little gradient update, and it does not need to, as the lexical retrieval model already produces satisfying results. On the other hand, if there is a vocabulary mismatch or topic difference, the lexical model may fail, causing the residual margin to be high and thereby driving the embedding model to accommodate in gradient update. Through the course of training, the neural model learns to encode the semantic patterns that are not captured by text surface forms. When training finishes, the two models will work together, as CLEAR.

3.4 Retrieval with CLEAR

CLEAR retrieves from the lexical and embedding index respectively, taking the union of the resulting candidates, and sorts using a final retrieval score: a weighted average of lexical matching and neural embedding scores:

$$s_{\text{CLEAR}}(q, d) = \lambda_{\text{test}} s_{\text{lex}}(q, d) + s_{\text{emb}}(q, d) \tag{8}$$

We give CLEAR the flexibility to take different λ_{train} and λ_{test} values. Though both are used for interpolating scores from different retrieval models, they have different interpretations. Training λ_{train} serves as a global control over the residual based margin. On the other hand, testing λ_{test} controls the contribution from the two retrieval components.

CLEAR achieves low retrieval latency by having each of the two retrieval models adopt optimized search algorithms and data structures. For the lexical retrieval model, CLEAR index the entire collection with a typical inverted index. For the embedding retrieval model, CLEAR pre-computes all document embeddings and indexes them with fast MIPS indexes such as FAISS [15] or SCANN [12], which can scale to millions of candidates with millisecond latency. As a result, CLEAR can serve as a first-stage, full-collection retriever.

4 Experimental Methodology

Dataset and Metrics. We use the MS MARCO passage ranking dataset [27], a widely-used ad-hoc retrieval benchmark with 8.8 millions passages. The training set contains 0.5 million pairs of queries and relevant passages, where each query on average has one relevant passage[1]. We used two evaluation query sets with different characteristics:

- **MS MARCO Dev Queries** is the MS MARCO dataset's official dev set, which has been widely used in prior research [8,28]. It has 6,980 queries. Most of the queries have only 1 document judged relevant; the labels are binary. MRR@10 is used to evaluate the performance on this query set following [27]. We also report the Recall of the top 1,000 retrieved (R@1k), an important metric for first-stage retrieval.
- **TREC2019 DL Queries** is the official evaluation query set used in the TREC 2019 Deep Learning Track shared task [6]. It contains 43 queries that are manually judged by NIST assessors with 4-level relevance labels, allowing us to understand the models' behavior on queries with *multiple, graded relevance judgments* (on average 94 relevant documents per query). NDCG@10, MAP@1k and R@1k are used to evaluate this query set's accuracy, following the shared task.

Compared Systems. We compare CLEAR retrieval with several first-stage lexical retrieval systems that adopt different techniques such as traditional BM25, deep learning augmented index and/or pseudo relevance feedback.

[1] Dataset is available at https://microsoft.github.io/msmarco/.

- **BM25** [32]: A widely-used off-the-shelf lexical-based retrieval baseline.
- **DeepCT** [8]: A state-of-the-art first-stage neural retrieval model. It uses BERT to estimate term importance based on context; in turn these context-aware term weights are used by BM25 to replace *tf* in Eq. 1.
- **BM25+RM3**: RM3 [20] is a popular query expansion technique. It adds related terms to the query to compensate for the vocabulary gap between queries and documents. BM25+RM3 has been proven to be strong [22].
- **DeepCT+RM3**: [7] shows that using DeepCT term weights with RM3 can further improve upon BM25+RM3.

In addition, we also compare with an embedding only model, **BERT-Siamese**: This is a BERT-based embedding retrieval model without any explicit lexical matching signals, as described in Subsect. 3.2. Note that although BERT embedding retrieval models have been tested on several question-answering tasks [4,13,21], their effectiveness for ad hoc retrieval remains to be studied.

Pipeline Systems. To investigate how the introduction of CLEAR will affect the final ranking in state-of-the-art pipeline systems, we introduce two pipeline setups.

- **BM25+BERT reranker**: this is a state-of-the-art *pipelined* retrieval system. It uses BM25 for first-stage retrieval, and reranks the top candidates using a BERT reranker [28]. Both the BERT-BASE and the BERT-LARGE reranker provided by [28] are explored. Note that BERT rerankers use a very deep self-attentive architecture whose computation cost limits its usage to only the reranking stage.
- **clear+BERT reranker**: a similar pipelined retrieval system that uses CLEAR as the first-stage retreiever, followed by a BERT reranker (BERT-BASE or BERT-LARGE reranker from [28]).

Setup. Lexical retrieval systems, including BM25, BM25+RM3, and deep lexical systems DeepCT and DeepCT+RM3, build upon Anserini [38]. We set k_1 and b in BM25 and DeepCT using values recommended by [8], which has stronger performance than the default values. The hyper-parameters in RM3 are found through a simple parameter sweep using 2-fold cross-validation in terms of MRR@10 and NDCG@10; the hyper-parameters include the number of feedback documents and the number of feedback terms (both searched over $\{5, 10, \cdots, 50\}$), and the feedback coefficient (searched over $\{0.1, 0.2, \cdots, 0.9\}$).

Our neural models were built on top of the HuggingFace [37] implementation of BERT. We initialized our models with BERT-BASE-UNCASED, as our hardware did not allow fine-tuning BERT-LARGE models. For training, we use the 0.5M pairs of queries and relevant documents. At each training step, we randomly sample one negative document from the top 1,000 documents retrieved by BM25. We train our neural models for 8 epochs on one RTX 2080 Ti GPU; training more steps did not improve performance. We set $\xi = 1$ in Eq. 6. We fixed $\lambda_{\text{train}} = 0.1$ in the experiments. For λ_{test}, we searched over $\{0, 1, 0.2, \cdots, 0.9\}$ on 500 training queries, finding 0.5 to be the most robust. Models are trained using the Adam

optimizer [17] with learning rate 2×10^{-5}, and batch size 28. In pipelined systems, we use BERT rerankers released by Nogueira et al. [28]. Statistical significance was tested using the permutation test with $p < 0.05$.

5 Results and Discussion

We study CLEAR's retrieval effectiveness on a large-scale, supervised retrieval task, its impact on downstream reranking, and its winning/losing cases.

Table 1. First-stage retrieval effectiveness of CLEAR on the MS MARCO dataset, evaluated using two query evaluation sets, with ablation studies. Superscripts 1–6 indicate statistically significant improvements over methods indexed on the left. ↓ indicates a number being statistically significantly lower than CLEAR. *: CLEAR w/ Constant Margin is equivalent to a post-training fusion of BM25 and BERT-Siamese.

Type	Model	MS MARCO Dev		TREC2019 DL		
		MRR @10	R@1k	NDCG @10	MAP @1k	R@1k
Lexical	1 BM25	0.191^{2}	0.864	0.506	0.377^{5}	0.738^{5}
	2 BM25+RM3	0.166	0.861	0.555^{1}	0.452^{135}	0.789^{13}
	3 DeepCT	0.243^{124}	0.913^{12}	0.551^{1}	0.422^{1}	0.756^{1}
	4 DeepCT+RM3	0.232^{12}	0.914^{12}	0.601^{123}	0.481^{123}	0.794^{13}
Embedding	5 BERT-Siamese	0.308^{1-4}	0.928^{123}	0.594^{123}	0.307	0.584
Lexical+ Embedding	6 CLEAR	$\mathbf{0.338}^{1-5}$	$\mathbf{0.969}^{1-5}$	$\mathbf{0.699}^{1-5}$	$\mathbf{0.511}^{1-5}$	$\mathbf{0.812}^{1-5}$
	− w/ Random Sampling	$0.241^{\downarrow}$	$0.926^{\downarrow}$	$0.553^{\downarrow}$	$0.409^{\downarrow}$	$0.779^{\downarrow}$
	− w/ Constant Margin*	$0.314^{\downarrow}$	$0.955^{\downarrow}$	$0.664^{\downarrow}$	$0.455^{\downarrow}$	0.794

5.1 Retrieval Accuracy of CLEAR

In this experiment, we compare CLEAR's retrieval performance with first stage retrieval models described in Sect. 4 and record their performance in Table 1.

Clear vs. Lexical Retrieval. CLEAR outperforms BM25 and BM25+RM3 systems by large margins in both recall-oriented metrics (R@1k and MAP@1k) as well as precision-oriented ones (MRR@10 and NDCG@10). CLEAR also significantly outperforms DeepCT and DeepCT+RM3, two BERT-augmented lexical retrieval models. DeepCT improves over BM25 by incorporating BERT-based contextualized term weighting, but still use exact term matching. The results show that lexical retrieval is limited by the strict term matching scheme, showing CLEAR's advantages of using embeddings for semantic-level soft matching.

Clear vs. BERT-Siamese Retrieval. BERT-Siamese performs retrieval solely relying on dense vector matching. As shown in Table 1, CLEAR outperforms BERT-Siamese with large margins, indicating that an embedding-only retrieval is not sufficient. Interestingly, though outperforming BM25 by a large margin on MSMARCO Dev queries, BERT-Siamese performs worse than BM25 in terms of MAP@1k and recall on TREC DL queries. The main difference between the two query sets is that TREC DL query has multiple relevant documents with graded

relevance levels. It therefore requires a better-structured embedding space to capture this, which proves to be harder to learn here. CLEAR circumvents this full embedding space learning problem by grounding in the lexical retrieval model while using embedding as complement.

Table 2. Comparing CLEAR and the state-of-the-art BM25+BERT Reranker pipeline on the MS MARCO passage ranking dataset with two evaluation sets (Dev: MS MARCO Dev queries; TREC: TREC2019 DL queries). We record the most optimal reranking depth for each initial retriever. Superscripts 1–6 indicate statistically significant improvements over the corresponding methods.

Retriever	Reranker	MSMARCO Dev	TREC DL	Rerank Depth
		MRR@10	NDCG@10	K
1 BM25	–	0.191	0.506	–
2 CLEAR	–	0.338^{1}	0.699^{1}	–
3 BM25	BERT-BASE	0.345^{1}	0.707^{1}	1k
4 CLEAR	BERT-BASE	0.360^{123}	0.719^{12}	20
5 BM25	BERT-LARGE	0.370^{123}	0.737^{123}	1k
6 CLEAR	BERT-LARGE	$\mathbf{0.380}^{1-5}$	$\mathbf{0.752}^{1-5}$	100

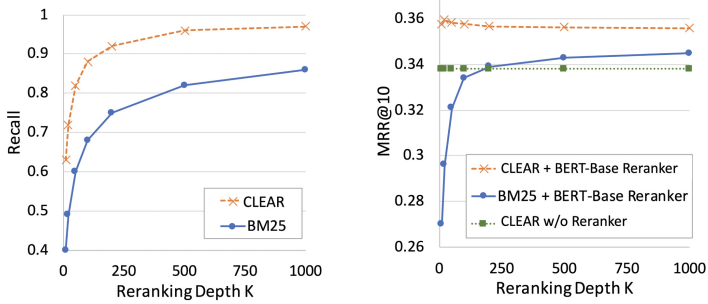
Ablation Studies. We hypothesize that CLEAR's residual-based learning approach can optimize the embedding retrieval to *complement* the lexical retrieval, so that the two parts can generate additive gains when combined. To verify this hypothesis, we run ablation studies by **(1)** replacing the error-based negative samples with random negative samples, and **(2)** replacing the residual margin in the loss function with a constant margin, which is equivalent to a fusion of BM25 and BERT-Siamese rankings. Using random negative samples leads to a substantial drop in CLEAR's retrieval accuracy, showing that it is important to train the embeddings on the mistakenly-retrieved documents from lexical retrieval to make the two retrieval models additive. Using constant margins instead of residual margins also lowers the performance of the original CLEAR model. By enforcing a residual margin explicitly, the embedding model is forced to learn to compensate for the lexical retrieval, leading to improved performance. The results confirm that CLEAR is more effective than a post-training fusion approach where the retrieval models are unaware of each other.

5.2 Impacts of CLEAR on Reranking

Similar to other fist-stage retrievers, CLEAR can be incorporated into the state-of-the-art pipelined retrieval system, where its candidate list can be reranked by a deep neural reranker. To quantitatively evaluate the benefit of CLEAR, in the next experiment, we test reranking CLEAR results with BERT rerankers.

Results are listed in Table 2. Here, we compare CLEAR against the widely-used BM25 in a two-stage retrieval pipeline, using current state-of-the-art BERT

rerankers [28] as the second stage reranking model. The rerankers use the concatenated query-document text as input to BERT to classify the relevance. We experimented with both BERT-BASE and BERT-LARGE reranker variants provided by [28]. We also investigate the reranking depth for each initial retriever and record the most optimal here.



(a) Retrieval Recall (b) Reranking Accuracy

Fig. 1. Comparison between CLEAR and BM25 pipeline systems on MS MARCO Dev queries. The system uses the BERT-BASE reranker to rerank against various depth K.

The performance of CLEAR *without reranking* is already close to that of the two-stage BM25+BERT-BASE reranker. When adding a reranker, CLEAR pipelines significantly outperforms the BM25 pipelines. We also discover that reranking a truncated top list for CLEAR is sufficient, while top 1000 is required for BM25. Concretely, the required re-ranking depth decreased from K=1,000 to K=20 for BERT-BASE reranker and K=100 for BERT-LARGE reranker, reducing the computational cost by 10×–50×. In other words, CLEAR generates strong initial rankings that systematically raise the position of relevant documents across all queries and help state-of-the-art rerankers to achieve higher accuracy with lower computational costs, improving end-to-end accuracy, efficiency, and scalability.

Figure 1 further plots the recall and reranking accuracy at various reranking depth. Figure 1a shows that CLEAR had higher recall values than BM25 at all depths, meaning that CLEAR can provide more relevant passages to the reranker. Figure 1b shows the performance of a BERT reranker [28] applied to the top K documents retrieved from either BM25 or CLEAR. When applied to BM25, the accuracy of the BERT reranker improved as reranking depth K increases. Interestingly for CLEAR, the reranking accuracy was already high with small K. While increasing K improves global recall, the reranking accuracy shows saturation with larger K, indicating that BERT rerankers do not fully exploit the lower portion of CLEAR candidate lists. We investigate this further in Subsect. 5.3.

5.3 Case Study: The Goods and the New Challenges

In this section, we take a more in-depth look into CLEAR through case studies. We first examine how BM25 ranking changes after being complemented by the dense embedding retrieval in CLEAR, then turn to investigate why the lower part of CLEAR's candidates are challenging for BERT rerankers.

Table 3. Example documents retrieved by CLEAR. We show ranking improvements from pure BM25 to CLEAR's complementary setup .

Query	Document retrieved by CLEAR	BM25 → CLEAR
Weather in danville, ca	Thursday:The Danville forecast for Aug 18 is 85 degrees and **Sunny** . There is 24% chance of **rain** and 10 mph **winds** from the West. Friday:...	989 → 10
brief **government** definition	Legal Definition of brief. 1 1 : a concise statement of a client's case written for the instruction of an **attorney** usually by a **law clerk** ...	996 → 7
population of **jabodatek**	The population of **Jabodetabek**, with an area of 6,392 km2, was over 28.0 million according to the Indonesian Census 2010	Not retrieved → 1

Table 4. Challenging non-relevant documents retrieved only by CRM, not by BM25, through semantic matching. We show in CLEAR initial candidate list ranking as well as after BERT reranking.

Query	Document retrieved by CLEAR	CLEAR → Rerank
Who is robert *gray*	*Grey* started ... dropping his Robert Gotobed alias and using his birthname Robert *Grey*.	Rank 496 → rank 7
What is *theraderm* used for	A *thermogram* is a device which measures heat through use of picture	Rank 970 → rank 8
What is the daily life of *thai people*	Activities of daily living include are the tasks that are required to get going in the morning ... 1 walking. 2 bathing. 3 dressing.	Rank 515 → rank 7

In Table 3, we show three example queries to which the CLEAR brings huge retrieval performance improvement. We see that in all three queries, critical query terms, *weather*, *government* and *jabodatek*, have no exact match in the

relevant document, leading to failures in exact match only BM25 system. CLEAR solves this problem, complementing exact matching with high-level semantic matching. As a result, "weather" can match with document content "sunny, rain, wind" and "government" with document content "attorny, law clerk". In the third query, spelling mismatch between query term "jabodatek" and document term "Jabodetabek" is also handled.

While CLEAR improves relevant documents' rankings in the candidate list, it also brings in new forms of non-relevant documents that are not retrieved by lexical retrievers like BM25, and affects downstream rerankers. In Table 4, we show three queries and three corresponding false positive documents retrieved by CLEAR, which are not retrieved by BM25. Unlike in BM25, where false positives mostly share surface text similarity with the query, in the case of CLEAR, the false positives can be documents that are topically related but not relevant. In the first two queries, CLEAR mistakenly performs soft spell matches, while in the third one critical concept "thai people" is ignored.

Such retrieval mistakes further affect the performance of downstream BERT reranker. As BERT also performs semantic level matching without explicit exact token matching to ground, the rerankers can amplify such semantically related only mistakes. As can be seen in Table 4, those false positive documents reside in the middle or at the bottom of the full candidate list of CLEAR. With BERT reranker, however, their rankings go to the top. In general, CLEAR goes beyond exact lexical matching to rely on semantic level matching. While improving initial retrieval, it also inevitably brings in semantically related false positives. Such false positives are inherently more challenging for state-of-the-art neural reranker and require more robust and discriminative rerankers. We believe this also creates new challenges for future research to improve neural rerankers.

6 Conclusion

Classic lexical retrieval models struggle to understand the underlying meanings of queries and documents. Neural embedding based retrieval models can soft match queries and documents, but they lose specific word-level matching information. This paper present CLEAR, a retrieval model that complements lexical retrieval with embedding retrieval. Importantly, instead of a linear interpolation of two models, the embedding retrieval in CLEAR is exactly trained to fix the errors of lexical retrieval.

Experiments show that CLEAR achieves the new state-of-the-art first-stage retrieval effectiveness on two distinct evaluation sets, outperforming classic bag-of-words, recent deep lexical retrieval models, and a BERT-based pure neural retrieval model. The superior performance of CLEAR indicates that it is beneficial to use the lexical retrieval model to capture simple relevant patterns using exact lexical clues, and complement it with the more complex semantic soft matching patterns learned in the embeddings.

Our ablation study demonstrates the effectiveness of CLEAR's residual-based learning. The error-based negative sampling allows the embedding model to be

aware of the mistakes of the lexical retrieval, and the residual margin further let the embeddings focus on the harder errors. Consequently, CLEAR outperforms post-training fusion models that directly interpolate independent lexical and embedding retrieval models' results.

A single-stage retrieval with CLEAR achieves an accuracy that is close to popular two-stage pipelines that uses a deep Transformer BERT reranker. We view this as an encouraging step towards building deep and efficient retrieval systems. When combined with BERT rerankers in the retrieval pipeline, CLEAR's strong retrieval performance leads to better end-to-end ranking accuracy and efficiency. However, we observe that state-of-the-art BERT neural rerankers do not fully exploit the retrieval results of CLEAR, pointing out future research directions to build more discriminative and robust neural rerankers.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2015)
2. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a Siamese time delay neural network. Adv. Neural Inf. Process. Syst. **6**, 737–744 (1993)
3. Caid, W.R., Dumais, S.T., Gallant, S.I.: Learned vector-space models for document retrieval. Inf. Process. Manag. **31**(3), 419–429 (1995)
4. Chang, W., Yu, F.X., Chang, Y., Yang, Y., Kumar, S.: Pre-training tasks for embedding-based large-scale retrieval. In: 8th International Conference on Learning Representations (2020)
5. Chen, T., Van Durme, B.: Discriminative information retrieval for question answering sentence selection. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pp. 719–725 (2017)
6. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.: Overview of the TREC 2019 deep learning track. In: TREC (to appear) (2019)
7. Dai, Z., Callan, J.: Context-aware document term weighting for ad-hoc search. In: WWW 2020: The Web Conference 2020, pp. 1897–1907 (2020)
8. Dai, Z., Callan, J.: Context-aware term weighting for first-stage passage retrieval. In: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (to appear) (2020)
9. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
10. Devlin, J., Chang, M., Lee, K., Toutanova, K.: 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, pp. 4171–4186 (2019)
11. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management, pp. 55–64 (2016)
12. Guo, R., et al.: Accelerating large-scale inference with anisotropic vector quantization. In: Proceedings of the 37th International Conference on Machine Learning (2020)

13. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: REALM: retrieval-augmented language model pre-training. CoRR abs/2002.08909 (2020)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
15. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. CoRR abs/1702.08734 (2017)
16. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP (2014)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations (2015)
18. Kuzi, S., Zhang, M., Li, C., Bendersky, M., Najork, M.: Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach. ArXiv abs/2010.01195 (2020)
19. Lafferty, J.D., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 111–119 (2001)
20. Lavrenko, V., Croft, W.B.: Relevance-based language models. In: SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 120–127 (2001)
21. Lee, K., Chang, M., Toutanova, K.: Latent retrieval for weakly supervised open domain question answering. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, pp. 6086–6096 (2019)
22. Lin, J.: The neural hype and comparisons against weak baselines. In: SIGIR Forum, pp. 40–51 (2018)
23. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Transactions of the Association of Computational Linguistics (2020)
24. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 824-836 (2018)
25. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 472–479 (2005)
26. Mitra, B., Diaz, F., Craswell, N.: Learning to match using local and distributed representations of text for web search. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1291–1299 (2017)
27. Nguyen, T., et al.: MS MARCO: A human generated machine reading comprehension dataset. In: Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches 2016 Co-Located with the 30th Annual Conference on Neural Information Processing Systems (2016)
28. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv:1901.04085 (2019)
29. Nogueira, R., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. CoRR abs/1904.08375 (2019)
30. Rajashekar, T.B., Croft, W.B.: Combining automatic and manual index representations in probabilistic retrieval. J. Am. Soc. Inf. Sci. **46**(4), 272–283 (1995)
31. Reimers, N., Gurevych, I.: Sentence-Bert: Sentence embeddings using Siamese Bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 3980–3990 (2019)

32. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 232–241 (1994)
33. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1984)
34. Shrivastava, A., Li, P.: Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). Adv. Neural Inf. Process. Syst. **27**, 2321–2329 (2014)
35. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, pp. 5998–6008 (2017)
36. Weston, J., Watkins, C.: Support vector machines for multi-class pattern recognition. In: ESANN 1999, 7th European Symposium on Artificial Neural Networks, pp. 219–224 (1999)
37. Wolf, T., et al.: Huggingface's transformers: State-of-the-art natural language processing. CoRR abs/1910.03771 (2019)
38. Yang, P., Fang, H., Lin, J.: Anserini: enabling the use of Lucene for information retrieval research. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1253–1256 (2017)
39. Yao, X., Van Durme, B., Clark, P.: Automatic coupling of answer extraction and information retrieval. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 159–165 (2013)
40. Zamani, H., Dehghani, M., Croft, W.B., Learned-Miller, E.G., Kamps, J.: From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 497–506 (2018)