UnifieR: A Unified Retriever for Large-Scale Retrieval

Anonymous ACL submission

Abstract

Large-scale retrieval is to recall relevant documents from a huge collection given a query. It relies on representation learning to embed documents and queries into a common semantic encoding space. According to the encoding space, recent retrieval methods based on pre-trained language models (PLM) can be coarsely categorized into either dense-vector or lexicon-based paradigms. These two paradigms unveil the PLMs' representation capability in different granularities, i.e., global sequence-level compression and local word-level contexts, respectively. Inspired by their complementary global-local contextualization and distinct representing views, we propose a new learning framework, UnifieR, which unifies dense-vector and lexicon-based retrieval in one model with a dual-representing capability. Experiments on passage retrieval benchmarks verify its effectiveness in both paradigms. A uni-retrieval scheme is further presented with even better retrieval quality. We lastly evaluate the model on BEIR benchmark to verify its transferability.

1 Introduction

Large-scale retrieval aims to efficiently fetch all relevant documents for a given query from a large-scale collection with millions or billions of entries[1]. It plays indispensable roles as a prerequisite for a broad spectrum of downstream tasks, e.g., information retrieval (Cai et al., 2021), open-domain question answering (Chen et al., 2017). To make online large-scale retrieval possible, the common practice is to represent queries and documents by an encoder in a Siamese manner (i.e., Bi-Encoder, BE) (Reimers and Gurevych, 2019). So, its success depends heavily on a powerful encoder by effective representation learning.

Advanced by pre-trained language models (PLM), e.g., BERT (Devlin et al., 2019), recent works propose to learn PLM-based encoders for large-scale retrieval, which are coarsely grouped into two paradigms in light of their encoding spaces with different focuses of representation granularity That is, *dense-vector encoding methods* leverage *sequence-level* compressive representations that embedded into dense semantic space (Xiong et al., 2021; Zhan et al., 2021; Gao and Callan, 2021b; Khattab and Zaharia, 2020), whereas *lexicon-based encoding methods* make the best of *word-level* contextual representations by considering either high concurrence (Nogueira et al., 2019) or coordinate terms (Formal et al., 2021b) in PLMs. To gather the powers of both worlds, some pioneering works propose *hybrid* methods to achieve a sweet point between dense-vector and lexicon-based methods for better retrieval quality. They focus on interactions of predicted scores between the two paradigms.

Nonetheless, such surface interactions – score aggregations (Kuzi et al., 2020), direct co-training (Gao et al., 2021b), and logits distillations (Chen et al., 2021b) – cannot fully exploit the benefits of the two paradigms – regardless of their complementary contextual features and distinct representation views. Specifically, as for *contextual features*, the dense-vector models focus more on sequence-level global embeddings against information bottleneck (Lu et al., 2021; Gao and Callan, 2021a,b), whereas the lexicon-based models focus on word-level local contextual embeddings for precise lexicon-weighting (Formal et al., 2021a, 2022; Nogueira et al., 2019). Aligning the two retrieval paradigms more closely is likely to benefit each other since global-local contexts are proven complementary in general representation learning (Shen et al., 2019; Beltagy et al., 2020). As for *representing views*, relying on distinct encoding spaces, the two retrieval paradigms are proven to provide different views in terms of query-document relevance (Kuzi et al., 2020; Gao et al., 2021b,a). Such a sort of 'dual views' has been proven piv-

[1] A collection entry could be *sentence*, *passage*, *document*, etc., and we take *document* for demonstrations.

otal in many previous cooperative learning works (Han et al., 2018; Chen et al., 2021a; Liang et al., 2021; Gao et al., 2021c), which provides a great opportunity to bridge the two retrieval paradigms. Consequently, without any in-depth interactions, neither the single (dense/lexicon) nor the hybrid retrieval model can be optimal.

Motivated by the above, we propose a brand-new learning framework, **Unified Retriever** (UnifieR), for in-depth mutual benefits of both dense-vector and lexicon-based retrieval. On the one hand, we present a neural encoder with dual representing modules for UnifieR, which is compatible with both retrieval paradigms. Built upon an underlying-tied contextualization that empowers consistent semantics sharing, a local-enhanced sequence representation module is presented to learn a dense-vector representation model. Meantime, a global-aware lexicon weighting module considering both the global- and local-context is proposed for a lexicon-based representation. On the other hand, we propose a new self-learning strategy, called dual-consistency learning, upon our unified encoder. Besides a basic contrastive learning objective, we first exploit the unified dual representing modules by mining diverse hard negatives for self-adversarial within the UnifieR. Furthermore, we present a self-regularization method based on list-wise agreements from the dual views for better consistency and generalization.

After being trained, UnifieR performs large-scale retrieval via either its lexicon representation by efficient inverted index or dense vectors by parallelizable dot-product. Moreover, empowered by our UnifieR, we present a fast yet effective retrieval scheme, *uni-retrieval*, to gather the powers of both worlds, where the lexicon retrieval is followed by a candidate-constrained dense scoring. Empirically, we evaluate UnifieR on not only passage retrieval benchmarks to check its effectiveness but the BEIR benchmark (Thakur et al., 2021) with twelve datasets (Natural Questions, HotpotQA, etc.) to verify the transferability of our model.

2 Related Work

PLM-based Retriever. Built upon PLMs, recent works propose to learn encoders for large-scale retrieval, which are coarsely grouped into two paradigms in light of their encoding spaces with different focuses of representation granularity: (i) *Dense-vector encoding methods* directly represent a document/query as a low-dimension sequence-level dense vector $\boldsymbol{u} \in \mathbb{R}^e$ (e is embedding size and usually small, e.g., 768). And the relevance score between a document and a query is calculated by dot-product or cosine similarity (Xiong et al., 2021; Zhan et al., 2021; Gao and Callan, 2021b; Khattab and Zaharia, 2020). (ii) *Lexicon-based encoding methods* make the best of word-level contextualization by considering either high concurrence (Nogueira et al., 2019) or coordinate terms (Formal et al., 2021b) in PLMs. It first weights all vocabulary lexicons for each word of a document/query based on the contexts, leading to a high-dimension sparse vector $\boldsymbol{v} \in \mathbb{R}^{|\mathbb{V}|}$ ($|\mathbb{V}|$ is the vocabulary size and usually large, e.g., 30k). The text is then denoted by aggregating over all the lexicons in a sparse manner. Lastly, the relevance is calculated by lexical-based matching metrics (e.g., BM25 (Robertson and Zaragoza, 2009)). In contrast, we unify the two paradigms into one carefully-designed encoder for better consistency within PLMs, leading to complementary information and superior performance.

Hybrid Retriever. Some works propose to bridge the gap between dense and lexicon for a sweet spot between performance and efficiency. A direct method is to aggregate scores of the two paradigms (Kuzi et al., 2020), but resulting in standalone learning and sub-optimal quality. Similar to our work, CLEAR (Gao et al., 2021b) uses a dense-vector model to complement the lexicon-based BM25 model, but without feature interactions and sophisticated learning. Sharing inspiration with our uni-retrieval scheme, COIL (Gao et al., 2021a) equips a simple lexicon-based retrieval with dense operations over word-level contextual embeddings. UnifieR differs in not only our lexicon representations jointly learned for in-depth mutual benefits but also sequence-level dense operations involved for memory-/computation-efficiency. Lastly, SPARC (Lee et al., 2020) distills ranking orders from a lexicon model (BM25) into a dense model as a companion of the original dense vector, which is distinct to our motivation.

Please see §A for more related works regarding our encoder structures and learning methods.

3 Methodology

Task Definition. Given a collection with numerous documents (i.e., $\mathbb{D} = \{d_i\}_{i=1}^{|\mathbb{D}|}$) and a textual query q from users, a retriever aims to fetch a list of text pieces $\bar{\mathbb{D}}_q$ to contain all relevant ones. Generally, this is based on a relevance score between q and every document d_i in a Siamese manner, i.e., $<\mathrm{Enc}(q), \mathrm{Enc}(d_i)>$, where Enc is an arbitrary representation model (e.g., Bag-of-Words and neural encoders) and $<\cdot,\cdot>$ denotes a lightweight relevance metric (e.g., BM25 and dot-product).

3.1 General Retriever Learning Framework

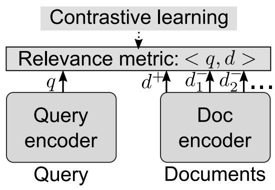

Figure 1: Bi-encoder learning.

To ground a method, we first introduce a contrastive learning framework to train a retrieval model (Figure 1). For supervision data in retriever training, differing from traditional categorical tasks, only query-document tuples (i.e., (q, d_q^+)) are given as positive pairs. Hence, given a q, a method needs to sample a set of negatives $\mathbb{N}_q = \{d_q^-\}_1^M$ from $\mathbb{D}$, and trains the retriever on tuples of $(q, d_q^+, \mathbb{N}_q)$. M is the number of negatives. If no confusion is caused, we omit the subscript 'q' for a specific query in the remaining.

Formally, given q and $\forall d \in \{d^+\} \cup \mathbb{N}$, an encoder, $\mathrm{Enc}(\cdot;\theta)$, is applied to them individually to produce their embeddings, i.e., $\mathrm{Enc}(q;\theta)$ and $\mathrm{Enc}(d;\theta)$, where the encoder is parameterized by θ if applicable. It is noteworthy we tie the query encoder with the document encoder in our work for simplicity. Then, a relevance metric is applied to each pair of the embeddings of the query and each document. Thus, a probability distribution over the documents $\{d^+\}\cup\mathbb{N}$ can be defined as

$$\boldsymbol{p} := P(\mathrm{d} \mid q, \{d^+\}\cup\mathbb{N}; \theta) = \frac{\exp(<\mathrm{Enc}(q;\theta), \mathrm{Enc}(d;\theta)>)}{\sum_{d' \in \{d^+\}\cup\mathbb{N}} \exp(<\mathrm{Enc}(q;\theta), \mathrm{Enc}(d';\theta)>)}, \quad (1)$$

where $\forall d \in \{d^+\} \cup \mathbb{N}$. Lastly, a contrastive learning loss to optimize the encoder θ is

$$L_\theta = -\log P(\mathrm{d} = d^+ \mid q, \{d^+\}\cup\mathbb{N}; \theta). \quad (2)$$

3.2 Neural Encoder in UnifieR

We present an encoder (see Figure 2) for UnifieR for dense-vector and lexicon-based retrieval.

Underlying-tied Contextualization. We first propose to share the low-level textual feature extractor between both representing paradigms. Although the two paradigms are focused on different representation granularities, sharing their underlying contextualization module can still facilitate semantic knowledge transfer between the two paradigms. As such, they can learn consistent semantic and syntactic knowledge towards the same retrieval targets, especially the salient lexicon-based features transferred to dense vectors. Formally, we leverage a multi-layer Transformer (Vaswani et al., 2017) encoder to produce word-level (token-level) contextualized embeddings, i.e.,

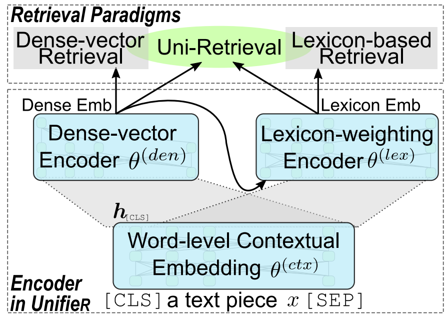

Figure 2: The encoder in UnifieR.

$$\boldsymbol{H}^{(x)} = \text{Transfm-Enc}(\texttt{[CLS]}x\texttt{[SEP]};\theta^{(ctx)}) \quad (3)$$

where $\forall x \in \{q\} \cup \{d^+\} \cup \mathbb{N}$, and [CLS] & [SEP] are special tokens by following PLMs (Devlin et al., 2019; Liu et al., 2019), $\boldsymbol{H}^{(x)} = [\boldsymbol{h}^{(x)}_{\texttt{[CLS]}}, \boldsymbol{h}^{(x)}_1, \ldots, \boldsymbol{h}^{(x)}_n, \boldsymbol{h}^{(x)}_{\texttt{[SEP]}}]$ are resulting embeddings, and n is the number of words in x.

Local-enhanced Sequence Representation. On top of the embeddings with enhanced local contexts, we then present a representing module to produce sequence-level dense vectors. For this purpose, we apply another multi-layer Transformer encoder to $\boldsymbol{H}^{(x)}$, followed by a pooler to derive a sequence-level vector. This can be written as

$$\boldsymbol{u}^{(x)} = \mathrm{Pool}(\text{Transfm-Enc}(\boldsymbol{H}^{(x)};\theta^{(den)})), \quad (4)$$

where this module is parameterized by $\theta^{(den)}$ untied with $\theta^{(ctx)}$, $\mathrm{Pool}(\cdot)$ gets a sequence-level dense vector by taking the embedding of special token [CLS], and the resulting $\boldsymbol{u}^{(x)} \in \mathbb{R}^e$ denotes a global dense representation of the input text x, which is used for dense-vector retrieval.

Global-aware Lexicon Weighting. Lastly, to achieve lexicon-based retrieval, we adapt a recent SParse Lexical AnD Expansion Model (SPLADE) (Formal et al., 2021a) into our neural encoder. SPLADE is a lexicon-weighting retrieval model which learns sparse expansion for each word in

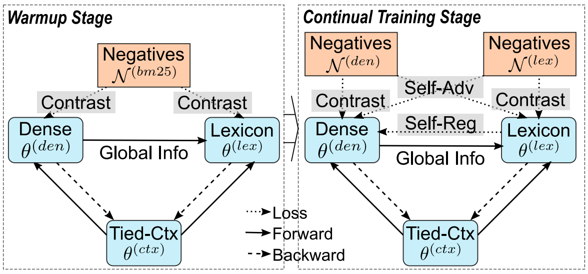

Figure 3: The two-stage self-learning strategy for UnifieR.

query/document x via the MLM head of PLMs and sparse regularization. Differing from the original SPLADE, our lexicon-based representing module not only shares its underlying feature extractor with a dense model but strengthens its hidden states by the global vector $\boldsymbol{u}^{(x)}$ above. The intuition is that, similar to text decoding with a bottleneck hidden state, the global context serves as high-level constraints (e.g., concepts/topics) to guide word-level operations (Sutskever et al., 2014; Lu et al., 2021; Gao and Callan, 2021a). In particular, the word-level contextualization embeddings passed into this module are manipulated as $\hat{\boldsymbol{H}}^{(x)} = [\boldsymbol{u}^{(x)}, \boldsymbol{h}_1^{(x)}, \ldots, \boldsymbol{h}_{\texttt{[SEP]}}^{(x)}]$. Then, a lexicon-weighting representation for x can be derived by

$$\boldsymbol{v}^{(x)} = \log(1 + \text{Max-Pool}(\text{ReLU}(\boldsymbol{W}^{(e)} \, \text{Transfm-Enc}(\hat{\boldsymbol{H}}^{(x)}); \theta^{(mlm)}))), \quad (5)$$

where, $\theta^{(mlm)}$ parameterizes a multi-layer Transformer encoder, $\boldsymbol{W}^{(e)} \in \mathbb{R}^{|\mathbb{V}| \times e}$ denotes the transpose of word embedding matrix as the MLM head, $|\mathbb{V}|$ denotes the vocabulary size, $\theta^{(lex)} = \{\boldsymbol{W}^{(e)}, \theta^{(mlm)}\}$ parameterizes this module, and $\boldsymbol{v}^{(x)} \in \mathbb{R}^{|\mathbb{V}|}$ is a sparse lexicon-based representation of x. And its sparsity is regularized by FLOPS (Paria et al., 2020) as in (Formal et al., 2021a). Here, the saturation function $\log(1+\text{Max-Pool}(\cdot))$ prevents some terms from dominating.

In summary, given a text x, UnifieR produces two embeddings via its dual representing modules:

$$\boldsymbol{u}^{(x)} := \text{Uni-Den}(x; \Theta^{(den)}),$$
$$\boldsymbol{v}^{(x)} := \text{Uni-Lex}(x; \Theta^{(lex)}), \quad (6)$$

where $\Theta^{(den)} = \{\theta^{(ctx)}, \theta^{(den)}\}$ and $\Theta^{(lex)} = \{\theta^{(ctx)}, \theta^{(den)}, \theta^{(lex)}\}$. Hence, $\boldsymbol{u}^{(x)} \in \mathbb{R}^{e}$ denotes a dense vector and $\boldsymbol{v}^{(x)} \in \mathbb{R}^{|\mathbb{V}|}$ denotes a sparse lexicon-based embedding.

3.3 Dual-Consistency Learning for UnifieR

To maximize our encoder's representing capacity, we propose a self-learning strategy, called dual-consistency learning (Figure 3). The 'dual-consistency' denotes learning the dual representing modules to achieve consistency in a unified model via *negative samples* and *module predictions*.

Basic Training Objective. To learn the encoder, a straightforward way is applying the contrastive learning loss defined in Eq.(1-2) to our dual representing modules. That is,

$$L^{(\text{con})} = -\log P(\mathrm{d} = d^{+} | q, \{d^{+}\} \cup \mathbb{N}; \Theta^{(den)}) - \log P(\mathrm{d} = d^{+} | q, \{d^{+}\} \cup \mathbb{N}; \Theta^{(lex)}), \quad (7)$$

where the former is for dense-vector retrieval while the latter is for lexicon-based retrieval. Towards the same retrieval target, the model is prone to learn consistent semantic and syntactic features via complementing the global-local granularity of the two retrieval paradigms. Due to the non-differentiability of lexicon-based metrics, we follow (Formal et al., 2021b) to use dot-product of lexicon-weighting representation during training but resort to a lexicon matching system (Yang et al., 2017) with quantization during indexing&retrieval. (see Appx. B for details) Note that $\theta^{(den)}$ would not be optimized w.r.t. the losses on top of the lexicon-based module. As for the query's negatives $\mathbb{N}$ of in Eq.(7), they are initially sampled by a BM25 retrieval system at *the warmup stage* (Zhan et al., 2021; Gao and Callan, 2021b), denoted as $\mathbb{N}^{(bm25)} = \{d | d \sim P(\mathrm{d} \mid q, \mathbb{D}_{\backslash\{d^{+}\}}; \text{BM25})\}$, where $\mathbb{D}_{\backslash\{d^{+}\}}$ denotes all documents in the collection $\mathbb{D}$ except the positive d^{+} for the query q.

Negative-bridged Self-Adversarial. However, it is verified that learning a retriever based solely on BM25 negatives cannot perform competitively (Xiong et al., 2021; Zhan et al., 2021). Thereby, previous works propose to sample hard negatives by the best-so-far retriever for continual training (Zhan et al., 2021; Gao and Callan, 2021b), a.k.a. self-adversarial learning (Sun et al., 2019). In our pilot experiments, we found the two retrieval paradigms can provide distinct hard negatives ($>40\%$ top-retrieved candidates are different) to ensure diversity after a combination. This motivates us to make the best of the hard negatives sampled by our dual representing modules: hard negatives sampled from one module can be applied to both itself and its counterpart in one unified framework. This can be regarded as a sort of self-distillation as both distilling samples (i.e., document mined from the collection) and distilling

Method	Pre-trained model	Reranker taught	Hard negs	Mul Repr	MS-Marco Dev			TREC DL 19	
					MRR@10	R@100	R@1k	R@100	nDCG@10
Dense-vector Retriever									
ANCE (Xiong et al., 2021)	RoBERTa$_{\text{base}}$				33.8	86.2	96.0	44.5	65.4
ADORE (Zhan et al., 2021)	RoBERTa$_{\text{base}}$		✓		34.7	87.6	-	47.3	68.3
TAS-B (Hofstätter et al., 2021)	DistilBERT	✓			34.7	-	97.8	-	71.2
TCT-ColBERT (Lin et al., 2021)	BERT$_{\text{base}}$	✓	✓		35.9	-	97.0	-	71.9
coCondenser (Gao and Callan, 2021b)	coCon$_{\text{base}}$		✓		38.2	-	**98.4**	-	-
ColBERTv1 (Khattab and Zaharia, 2020)	BERT$_{\text{base}}$			✓	36.0	-	96.8	-	-
ColBERTv2 (Santhanam et al., 2021)	BERT$_{\text{base}}$	✓	✓	✓	39.7	-	**98.4**	-	-
RocketQAv2 (Ren et al., 2021b)	ERNIE$_{\text{base}}$	✓	✓		38.8	-	-†	-	-
AR2 (Zhang et al., 2022)	coCon$_{\text{base}}$	✓	✓		39.5	-	-†	-	-
Lexicon-base or Sparse Retriever									
DeepCT (Dai and Callan, 2019)	BERT$_{\text{base}}$				24.3	-	91.3	-	55.1
SPLADE-max (Formal et al., 2021a)	DistilBERT				34.0	-	96.5	-	68.4
DistilSPLADE-max (Formal et al., 2021a)	DistilBERT	✓			36.8	-	97.9	-	72.9
SelfDistil (Formal et al., 2022)	DistilBERT	✓	✓		36.8	-	98.0	-	72.3
EnsembleDistil (Formal et al., 2022)	DistilBERT	✓	✓		36.9	-	97.9	-	72.1
Co-SelfDistil (Formal et al., 2022)	coCon$_{\text{base}}$	✓	✓		37.5	-	**98.4**	-	73.0
Co-EnsembleDistil (Formal et al., 2022)	coCon$_{\text{base}}$	✓	✓	✓	38.0	-	98.2	-	73.2
Hybrid Retriever									
CLEAR (Gao et al., 2021b)	BERT$_{\text{base}}$			✓	33.8	-	96.9	-	69.9
COIL-full (Gao et al., 2021a)	BERT$_{\text{base}}$			✓	35.5	-	96.3	-	70.4
UnifieR$_{\text{lexicon}}$ (*warmup*)	coCon$_{\text{base}}$				37.2	90.1	97.8	50.1	69.7
UnifieR$_{\text{dense}}$ (*warmup*)	coCon$_{\text{base}}$				36.1	87.7	96.6	44.6	63.9
UnifieR$_{\text{uni-retrieval}}$ (*warmup*)	coCon$_{\text{base}}$			✓	38.3	90.8	98.0	50.6	70.2
UnifieR$_{\text{lexicon}}$	coCon$_{\text{base}}$		✓		39.7	91.2	98.1	53.2	73.3
UnifieR$_{\text{dense}}$	coCon$_{\text{base}}$		✓		38.8	90.3	97.6	50.2	71.1
UnifieR$_{\text{uni-retrieval}}$	coCon$_{\text{base}}$		✓	✓	**40.7**	**92.0**	**98.4**	**53.8**	**73.8**

Table 1: Passage retrieval results on MS-Marco Dev and TREC Deep Learning 2019. †Refer to Table 2. 'coCon': coCondenser that continually pre-trained BERT in unsupervised manner. 'Reranker taught': distillation from a reranker (see §A).

labels (i.e., negative label only) are sourced from one unified model. So, we first sample two sets of negatives from the dual-representing modules:

$$\mathbb{N}^{(den)} = \left\{ d | d \sim P(\mathrm{d} \mid q, \mathbb{D}_{\backslash \{d^+\}}; \Theta^{(den)}) \right\},$$
$$\mathbb{N}^{(lex)} = \left\{ d | d \sim P(\mathrm{d} \mid q, \mathbb{D}_{\backslash \{d^+\}}; \Theta^{(lex)}) \right\}, \quad (8)$$

where our UnifieR was trained with $\mathbb{N}^{(bm25)}$ at *warmup stage*. Next, we upgrade $\mathbb{N}$ in Eq.(7) from $\mathbb{N}^{(bm25)}$ at *warmup stage* to $\mathbb{N}^{(den)} \cup \mathbb{N}^{(lex)}$, and then perform a *continual learning stage*.

Agreement-based Self-Regularization. We lastly present a self-regularization method for UnifieR. Its goal is to achieve an agreement from different views through our dual representing modules. Such an agreement-based self-regularization has been proven effective in both retrieval model training (via retriever-reranker agreements for consistent results (Ren et al., 2021b; Zhang et al., 2022)) and general representation learning (via agreements from various perturbation-based views for better generalization (Chen et al., 2021a; Liang et al., 2021; Gao et al., 2021c)). It is stronger than the contrastive learning in Eq.(7) as the agreement is learned by a KL divergence, i.e.,

$$L^{(\mathrm{reg})} = D_{\mathrm{KL}}(P(\mathrm{d}|q, \{d^+\} \cup \mathbb{N}; \Theta^{(den)}) \quad (9)$$
$$\| P(\mathrm{d}|q, \{d^+\} \cup \mathbb{N}; \Theta^{(lex)})).$$

Overall Training Pipeline. In line with (Gao and Callan, 2021b), we lastly follow a simple three-step pipeline to learn our retriever on the basis of the proposed training objectives and hard negatives: (i) ***Warmup Stage***: Initialized by a pre-trained model, UnifieR is updated w.r.t. Eq.(7) + λ FLOPS (by following (Formal et al., 2021a) for sparsity), with BM25 negatives $\mathbb{N}^{(bm25)}$. (ii) ***Hard Negative Mining***: According to the warmup-ed UnifieR, static hard negatives, $\mathbb{N}^{(den)}$ and $\mathbb{N}^{(lex)}$, are sampled by Eq.(8). (iii) ***Continual Learning Stage***: Continual with the warmup-ed UnifieR, the model is finally optimized on $\mathbb{N}^{(den)} \cup \mathbb{N}^{(lex)}$ w.r.t. a direct addition of Eq.(7&9)+λ FLOPS.

3.4 Retrieval Schemes

As in Figure 2, our model is fully compatible with the previous two retrieval paradigms. In addition, we present a *uni-retrieval* scheme for fast yet effective large-scale retrieval. Instead of adding their scores (Kuzi et al., 2020; Formal et al., 2022) from

Method	M@10	*R@50*	*R@1K*
RocketQA (Qu et al., 2021)	37.0	85.5	97.9
PAIR (Ren et al., 2021a)	37.9	86.4	98.2
RocketQAv2	38.8	86.2	98.1
AR2	39.5	87.8	**98.6**
UnifieR$_{\text{lexicon}}$	39.7	87.6	98.2
UnifieR$_{\text{dense}}$	38.8	86.3	97.8
UnifieR$_{\text{uni-retrieval}}$	**40.7**	**88.2**	98.5

Table 2: MS-Marco retrieval on *one-positive-enough* recall.

twice-retrieval with heavy overheads, we pipelinelize the retrieval procedure: given q, our lexicon-based retrieval under an inverted file system is to retrieve top-K documents from $\mathbb{D}$. Then, our dense-vector retrieval is then applied to the constrained candidates for dense scores. The final retrieval results are according to a simple addition of the two scores. We use 'addition' as our combination baseline for its generality and explore more advanced methods in §4.4. And, due to fast dense-vector dot-product calculations on top-K documents, uni-retrieval's latency is almost equal to single lexicon-based retrieval. Please see §B for details about lexicon-based inference in large-scale retrieval, especially what's the difference with the dense one.

4 Experiment

Datasets & Metrics. In line with (Formal et al., 2021a), we use popular passage retrieval datasets, MS-Marco (Nguyen et al., 2016), with official queries (no augmentations (Ren et al., 2021b)), and report for MS-Marco Dev set and TREC Deep Learning 2019 set (Craswell et al., 2020). Following previous works, we report MRR@10 (M@10) and Recall@1/50/100/1K[2] for MS-Marco Dev, and report nDCG@10 and R@100 for TREC Deep Learning 2019. Besides, we also transfer our model trained on MS-Marco to the BEIR benchmark (Thakur et al., 2021) to evaluate its generalizability, where nDCG@10 is reported. We take 12 datasets (i.e., TREC-COVID, NFCorpus, NQ, HotpotQA, FiQA, ArguAna, Tóuche-2020, DBPedia, Scidocs, Fever, Climate-FEVER, and SciFact) in the BEIR benchmark as they are widely-used across most previous papers. Please refer to §D for our pre-training and fine-tuning setups.

[2]We follow official evaluation metrics at `https://github.com/castorini/anserini`. But, we found 2 kinds of Recall@N on MS-Marco in recent papers, i.e., official *all-positive-macro recall* and *one-positive-enough recall* (see §C for details). Thereby, we report the former by default but list the latter separately for fair comparisons.

	Method	Avg	Best	In-Dm
Lexicon -based	BM25 (Thakur et al., 2021)	41.1	1	22.8
	DocT5Query	42.4	0	33.8
	UniCOIL (Lin and Ma, 2021)	40.0	0	-
Dense -vector	ColBERT	41.8	2	40.8
	ANCE (Xiong et al., 2021)	37.7	0	38.8
	GenQ (Thakur et al., 2021)	39.8	1	40.8
	TAS-B (Hofstätter et al., 2021)	40.4	0	40.8
	Contriever (Izacard et al., 2021)	44.3	**4**	-
	UnifieR$_{\text{uni-retrieval}}$	**44.5**	**4**	**47.1**
Reranker taught	ColBERT-v2	47.0	N/A	42.5
	DistilSPLADE	47.0	N/A	43.3
Huge models	GTR-XXL (Ni et al., 2021)	45.9	N/A	44.2
	SGPT-5.8B (Muennighoff, 2022)	49.4	N/A	39.9

Table 3: Retrieval nDCG@10 results on BEIR with 12 out-of-domain datasets, as well as 1 in-domain dataset. **Avg** is mean nDCG over 12 datasets and **Best** is how many datasets a method achieves best. DocT5Query (Nogueira et al., 2019), ColBERT (Khattab and Zaharia, 2020), ColBERT-v2 (Santhanam et al., 2021), DistilSPLADE (Formal et al., 2021a).

4.1 Main Evaluation

MS-Marco Dev. As in Table 1&2, our framework achieves new state-of-the-art metrics on most metrics. Our dense-vector retrieval surpasses previous methods without distillations from rerankers, while our lexicon-based retrieval pushes the best sparse method to a new level, especially in MRR@10 (+1.4%). Empowered by our unified structure, the uni-retrieval scheme can achieve 40.7% MRR@10. Although R@1K is approaching its ceiling across recent works, we notice UnifieR is less competitive than AR2 (-0.2%) in Table 2, as the latter involves a costly reranker in training for better generalization. And please see §4.4 for our rerank-taught results.

TREC Deep Learning 2019. As listed in Table 1, our retrieval method, with either single (dese/lexicon) or unified representation, achieves a state-the-of-art or very competitive retrieval quality. Specifically, compared to the previous best method, called TAS-B, our model lifts MRR@10 and nDCG@10 by 6.9% and 2.6%, respectively.

BEIR Benchmark. Table 3 shows in-domain evaluation and zero-shot transfer on BEIR (see §E.1). It is observed that, with outstanding in-domain inference ability, our model also delivers comparable transferability among the retrievers with similar training settings (i.e., comparable models o/w reranker distillations). But, as shown in the table, we found our model suffers from inferior generalization ability compared to the models with MSE-based reranker distillation (Santhanam et al., 2021; Formal et al., 2021a). And a small model with distillation (e.g., DistilSPLADE) even beats

Method	M@10	R@1
UnifieR$_{\text{uni-retrieval}}$	40.7	26.9
Uni-scheme of Best[1]	40.3	26.1
Ensemble of Best[1]	40.4	26.5
Ensemble of SPLADE[2]	40.0	-
COIL-full (hybrid)	35.5	-

Table 4: Comparison with ensemble and hybrid retrievers. [1]We operate on the best SPLADE model (MRR@10=38.5) with the best coCondenser (MRR@10=38.2). [2]An ensemble of four SPLADE models.

Methods	Lexicon-based		Dense-vector	
	M@10	R@100	M@10	R@100
UnifieR *(warmup)*	37.2	90.1	36.1	87.7
◇ w/o sharing Global	36.1	89.8	35.2	87.2
◇ w/o in-depth Interact	36.1	89.3	35.7	89.7

Table 5: Ablation of the encoder on MS-Marco Dev.

the models with billions of parameters (e.g., GTR-XXL). The potential reasons are two-fold: i) distilling a reranker to the retriever has been proven to produce more generalizable scores than a bi-encoder (Menon et al., 2021) and ii) the initialization of UnifieR, coCondenser, has been pre-trained on Marco collection, reducing its generalization.

4.2 Further Analysis

Comparison to Ensemble Models. As in Table 4, we report the numbers to compare our uni-retrieval scheme with ensemble models. Even if we only need once large-scale retrieval followed by a small amount of dot-product calculation, the model still surpasses its competitors. Meantime, we found both uni-retrieval and ensemble are bounded by the worse participant. For example, even if we use a SPLADE with MRR@10 of 39.3 for 'Ensemble/Uni-scheme of Best', the performance did not show a remarkable gain. This suggests us to look for a better aggregation method in the future.

Ablation of Neural Structure. To verify the correctness of each module design, we conduct an ablation study on the neural structure of the encoder (§3.2) in Table 5. This must be performed at the warmup step as the second stage is continual from the warmup. It is observed that, either removing the global information from the lexicon-based module or discarding in-depth inter-paradigm interactions (i.e., learning independently) degrades the model dramatically. Surprisingly, removing the global also diminishes dense performance. A potential reason is that, such a change makes the fine-tuning inconsistent with its initializing pre-trained model, coCondenser, leading to corrupted representing capability. Please refer to §E.2 for additional ablations about learning and data.

Top-N	QPS	M@10	R@100	Remark
BM25	449	19.3	69.0	
ORG	50	41.3	92.3	Not sparsified
75	129	40.8	91.5	↓ Index as Sparse as BM25
50	188	40.4	91.1	
25	343	38.4	89.0	
20	446	37.5	87.6	↓ Infer as Faster than BM25
15	537	36.2	86.0	
10	693	33.6	82.2	
8	911	31.9	79.8	
4	954	25.5	70.0	↑ Better than BM25
2	1144	16.2	53.2	
1	1376	1.8	22.3	

Table 6: UnifieR-lex v.s. QPS by Top-N lexicon sparsifying. The QPS is calculated on a CPU machine with pre-embedded queries, and ORG denotes non-sparsified UnifieR.

Evaluation of Learning Consistency. To verify if the dual representing modules depend on consistent semantic/syntactic features for the common target, we conduct an experiment to train one of the dual modules but leave the other unchanged at *continual training* stage. As in Figure 4, the left-most one is warmup-ed UnifeR (warmup), whereas the rightmost one is the full UnifeR (dual-trn) as an upper bound of performance. Interestingly, optimizing for each of the representing modules can improve both retrieval paradigms (i.e., lexicon and dense). This confirms that optimizing one module can benefit the other, attributed to complementary representations and the consistent learning target.

4.3 Efficiency Analysis

FLOPS analysis. To view sparsity-efficacy trade-off, we vary the loss weight λ for FLOPS sparse regularization (Paria et al., 2020). As in Figure 5, with λ exponentially increasing, document FLOPS decreases linearly, improving the efficiency of our framework. Meantime, the descending of lexicon-based efficacy is not remarkable when FLOPS > 4 and then becomes notable with the growth of λ. Fortunately, this will not affect the dense representation in terms of dense-vector retrieval.

Uni-Retrieval Hyperparameter. In uni-retrieval scheme, a hyperparam K is used to control computation overheads of dense dot-product. As illustrated in Figure 6, 'K=0' denotes lexicon-only retrieval in UnifieR. The table shows that UnifieR reaches an MRR@10 ceiling when K is set to a de facto number, i.e., 1000. Then, the upper bound of R@1000 is reached when K=2048. After that, the two metrics cannot be observed with any changes.

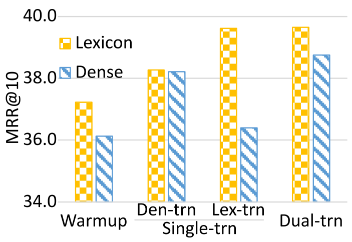

Figure 4: Verifying consistency of dual representing modules. 'trn' denotes 'training'.

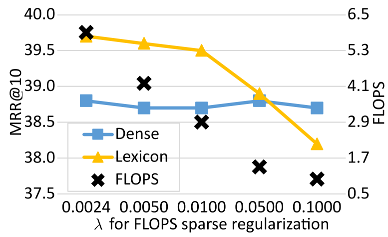

Figure 5: Effects of the loss weight λ of FLOPS sparse regularization on the our performance.

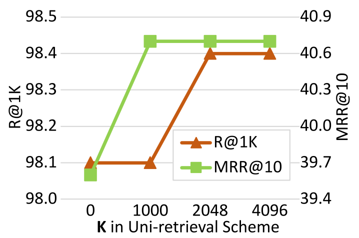

Figure 6: Effects of the hyperparam K in our uni-retrieval scheme on MS-Marco Dev.

Method	M@10	R@100
UnifieR-uni (warmup)	38.3	90.8
+ **query-side gating**	39.2	91.2

Table 7: Stage 1 of UnifieR with query-side gating.

Methods	Dense-vector		Lexicon-based		Uni/Multi-Vec	
	M@10	**R@100**	**M@10**	**R@100**	**M@10**	**R@100**
Previous SoTA	39.5	-	37.5	-	39.7	-
UnifieR	38.8	90.3	39.7	91.2	40.7	92.0
UnifieR (**distill**)	40.5	91.6	41.3	92.3	42.0	93.0

Table 8: Reranker-taught UnifieR v.s. previous state-of-the-art (SoTA) models (i.e., Dense (Zhang et al., 2022), Lexicon(Formal et al., 2022), Multi-Vec (Santhanam et al., 2021)).

Latency Analysis. Besides the un-intuitive FLOPS numbers, we also exhibit the latency (measured by 'query-per-second' – QPS) of UnifieR. Basically, our UnifieR is bottlenecked by its lexicon head in terms of inference speed as aforementioned, so we would like to dive into the controllable sparsity of UnifieR-lex. Note that, to reserve a large room for further sparsifying, we leverage the reranker-taught UnifieR-lex as shown in Table 8, whose MRR@10 is 41.3%. Then, we adopt a simple but effective sparsifying method – top-N (Yang et al., 2021) – but in the index-building process only. As a result, we show the performance of our UnifieR-lexicon with N decreasing in Table 6. It is shown that only the Top-4 tokens kept for each passage can still deliver very competitive results with faster speed than BM25.

4.4 Exploration of Advanced Architecture

Query-side Gating Mechanism. As it is too rough to directly add the scores of the two retrieval paradigms, we incorporate a recent inspiration of mix-of-expert (MoE) to enhance the combination of the two paradigms. As in illustrated in §E.4, we leveraged a gating mechanism to switch UnifieR between dense and lexicon, based solely on the semantics of queries. The reasons for "solely on queries" are two-fold: i) the analyses in §F show that the type of queries affects models a lot and ii) the dependency on queries only will not affect the indexing procedure for large-scale collections, leading to zero extra inference overheads. After this gating mechanism in the warmup stage of UnifieR training where the gate's optimization is based on the relevance score of uni-retrieval. As listed in Table 7, a remarkable improvement is observed with such a query-side gating mechanism (+0.9% MRR@10 and +0.4% R@100).

Reranker-taught UnifieR. Although the UnifieR in Table 1 & 2 seems significant in terms of performance improvement, it's noteworthy that the comparisons are unfair because UnifieR didn't use a re-ranker (a strong but heavy cross-encoder) as a teacher for knowledge distillation (see 'Reranker taught' in Table 1). To make the comparisons fairer, we first trained a re-ranker based on UnifieR's hard negatives and then used a KL loss for distillation in the Continual Training Stage (as illustrated in Figure 8 of §E.5). As listed in Table 8, it is shown that i) our proposed UnifieR is compatible with 'Reranker taught' scheme and consistently brings 1%+ improvement, and ii) UnifieR outperforms its strong competitors by large margins (2.0%+).

5 Conclusion

We present a brand-new learning framework, dubbed UnifieR, to unify dense-vector and lexicon-based representing paradigms for large-scale retrieval. It improves the two paradigms by a carefully designed neural encoder to fully exploit the representing capability of pre-trained language models. Its capability is further strengthened by our proposed dual-consistency learning with self-adversarial and -regularization. Moreover, the uni-retrieval scheme and the advanced architectures upon our encoder are presented to achieve more. Experiments on several benchmarks verify the effectiveness and versatility of our framework.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Yinqiong Cai, Yixing Fan, Jiafeng Guo, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2021. Semantic models for the first-stage retrieval: A comprehensive review. *CoRR*, abs/2103.04831.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Jiaao Chen, Dinghan Shen, Weizhu Chen, and Diyi Yang. 2021a. Hiddencut: Simple data augmentation for natural language understanding with better generalizability. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4380–4390. Association for Computational Linguistics.

Xilun Chen, Kushal Lakhotia, Barlas Oguz, Anchit Gupta, Patrick S. H. Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021b. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *CoRR*, abs/2110.06918.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James R. Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. *CoRR*, abs/2204.10298.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *CoRR*, abs/1910.10687.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural IR models more effective. *CoRR*, abs/2205.04733.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. SPLADE: sparse lexical and expansion model for first stage ranking. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2288–2292. ACM.

Luyu Gao and Jamie Callan. 2021a. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 981–993. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2021b. Unsupervised corpus aware language model pre-training for dense passage retrieval. *CoRR*, abs/2108.05540.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3030–3042. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021b. Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 146–160. Springer.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021c. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR*, abs/2010.02666.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 113–122. ACM.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *CoRR*, abs/2112.09118.

Feihu Jin, Jinliang Lu, Jiajun Zhang, and Chengqing Zong. 2022. Instance-aware prompt learning for language understanding and generation. *CoRR*, abs/2201.07126.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. 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 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach. *CoRR*, abs/2010.01195.

Jinhyuk Lee, Min Joon Seo, Hannaneh Hajishirzi, and Jaewoo Kang. 2020. Contextualized sparse representations for real-time open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 912–919. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 10890–10905.

Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *CoRR*, abs/2106.14807.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Shuqi Lu, Chenyan Xiong, Di He, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is more: Pre-training a strong siamese encoder using a weak decoder. *CoRR*, abs/2102.09206.

Aditya Krishna Menon, Sadeep Jayasumana, Seungyeon Kim, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. 2021. In defense of dual-encoders for neural ranking.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *CoRR*, abs/2202.08904.

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 *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 (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large dual encoders are generalizable retrievers. *CoRR*, abs/2112.07899.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint*, 6.

Biswajit Paria, Chih-Kuan Yeh, Ian En-Hsu Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing flops to learn efficient sparse representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5835–5847. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. 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, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. PAIR: leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2173–2183. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2825–2835. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *CoRR*, abs/2112.01488.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Tensorized self-attention: Efficiently modeling pairwise and global dependencies together. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1256–1266. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *CoRR*, abs/2104.08663.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, V. G. Vinod Vydiswaran, and Hao Ma. 2022. IDPG: an instance-dependent prompt generation method. *CoRR*, abs/2204.04497.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jheng-Hong Yang, Xueguang Ma, and Jimmy Lin. 2021. Sparsifying sparse representations for passage retrieval by top-k masking. *CoRR*, abs/2112.09628.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. 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, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1253–1256. ACM.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1503–1512. ACM.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker for dense text retrieval. In *International Conference on Learning Representations*.

A More Related Work

Bottleneck-based Learning. In terms of neural designs, our encoder is similar to several recent representation learning works, e.g., SEED-Encoder (Lu et al., 2021). Condenser (Gao and Callan, 2021a), coCondenser (Gao and Callan, 2021b), and DiffCSE (Chuang et al., 2022), but they focus on the bottleneck of sequence-level dense vectors. For example, SEED-Encoder, Condenser, and CoCondenser enhance their dense capabilities by emphasizing the sequence-level bottleneck vector and weakening the word-level language modeling heads, while DiffCSE makes the learned sentence embedding sensitive to the difference between the original sentence and an edited sentence by a word-level discriminator. With distinct motivations and targets, we fully exploit both the dense-vector bottleneck and the word-level representation learning in a PLM for their mutual benefits. These are on the basis of not only the shared neural modules but also structure-facilitated self-learning strategies (see the next section). Nonetheless, as discussed in our experiments, our model can still benefit from these prior works via parameter initializations.

Instance-dependent Prompt. Our model also shares high-level inspiration with recent instance-dependent prompt learning methods (Jin et al., 2022; Wu et al., 2022). They introduce a trainable component to generate prompts based on each input example. Such generated prompts can provide complementary features to the original input for a better prediction quality. Analogously, our sequence-level dense vector can be seen as a sort of 'soft-prompt' for the sparse lexicon-based representation module, resulting in the superiority of our lexicon-based retrieval, which will be discussed in experiments. In addition, the 'soft-prompt' in our UnifieR also serves as crucial outputs in a unified retrieval system.

Reranker-taught Retriever. Distilling the scores from a reranker into a retriever is proven promising (Hofstätter et al., 2020; Formal et al., 2021a; Hofstätter et al., 2021) . In light of this, recent works propose to jointly optimize a retriever and a reranker: RocketQAv2 (Ren et al., 2021b) is proposed to achieve their agreements with reranker-filtered hard negatives, while AR2 (Zhang et al., 2022) is to learn them in an adversarial fashion where the retriever is regarded as a generator and the reranker as a discriminator. In contrast to reranker-retriever co-training, we resort to in-depth sharing from the bottom (i.e., features) to the top (i.e., self-learning) merely within a retriever, with no need for extra overheads of reranker training. Meantime, our unified structure also uniquely enables it to learn from more diverse hard negatives mined by its dual representing modules.

B Lexicon-based Inference for Large-Scale Retrieval

During the inference of large-scale retrieval, there are some differences between dense-vector and lexicon-based retrieval methods.

As in Eq.(1), we use the dot-product between the real-valved sparse lexicon-based representations as a relevance metric, where 'real-valved' is a prerequisite of gradient back-propagation and end-to-end learning. However, it is inefficient and infeasible to leverage the real-valved sparse representations, especially for the open-source term-based retrieval systems, e.g., LUCENE and Anserini (Yang et al., 2017). Following Formal et al. (2021a), we adopt 'quantization' and 'term-based system' to complete our retrieval procedure. That is, to transfer the high-dimensional sparse vectors back to the corresponding lexicons and their virtual frequencies, the lexicons are first obtained by keeping the non-zero elements in a high-dim sparse vector, and each virtual frequency then is derived from a straightforward quantization (i.e., $\lfloor 100 \times \boldsymbol{v} \rfloor$).

In summary, the overall procedure of our large-scale retrieval based on a fine-tuned UnifieR-lex is i) generating the high-dim sparse vector for each document and transferring it to lexicons and frequencies, ii) building a term-based inverted index via Anserini (Yang et al., 2017) for all documents in a collection, iii) given a test query, generating the lexicons and frequencies, in the same way, and iv) querying the built index to get top document candidates.

C Explanation of Two Recall Metrics

Regarding R@N metric, we found there are two kinds of calculating ways, and we strictly follow the official evaluation one at `https://github.com/usnistgov/trec_eval` and `https://github.com/castorini/anserini`, which is defined as

$$\text{Marco-Recall@N} = \frac{1}{|\mathbb{Q}|}\sum_{q\in\mathbb{Q}} \frac{\sum_{d_+\in\mathbb{D}_+} \mathbf{1}_{d_+\in\bar{\mathbb{D}}}}{\min(N,|\mathbb{D}_+|)}, \tag{10}$$

where there may be multiple positive documents $\mathbb{D}^+ \in \mathbb{D}$, $\mathbb{Q}$ denotes the test queries and $\bar{\mathbb{D}}$ denotes top-K document candidates by a retrieval system. We also call this metric *all-positive-macro* Recall@N. On the other hand, another recall calculation method following DPR (Karpukhin et al., 2020) is defined as

$$\text{DPR-Recall@N} = \frac{1}{|\mathbb{Q}|}\sum_{q\in\mathbb{Q}} \mathbf{1}_{\exists d\in\bar{\mathbb{D}} \wedge d\in\mathbb{D}^+}. \tag{11}$$

which we call *one-positive-enough* Recall@N. Therefore, The official (*all-positive-macro*) Recall@N is usually less than DPR (*one-positive-enough*) Recall@N, and the smaller N, the more obvious.

D Experimental Setups

As stated in §3.3, we take a 2-stage learning scheme (Gao and Callan, 2021b). We use coCondenser-marco (Gao and Callan, 2021b) (unsupervised continual pre-training from BERT-base (Devlin et al., 2019)) as our initialization as it shares a similar neural structure (see the end of §3.2) and has potential for promising performance (Gao and Callan, 2021b; Formal et al., 2022; Zhang et al., 2022). $\theta^{(ctx)}$, $\theta^{(den)}$, and $\theta^{(lex)}$ correspond to Transformer layers of 6, 6, and 2, respectively, where max length is 128 and warmup ratio is 5%. At warmup stage, batch size of queries is 16, each with 1 positive document and 15 negatives, learning rate is 2×10^{-5}, the random seed is fixed to 42. And loss weight of FLOPS (Paria et al., 2020) is set to 0.0016 since we want make the model sparser than SPLADE (Formal et al., 2021a) (0.0008). At continual learning stage, batch size is 12 to enable each module with 15 negatives. And learning rate is reduced to 1/3 of the original, and the random seed is changed to 22 for a new data feeding order. And the loss weight of FLOPS is lifted to 0.0024. We did not tune the hyperparameters. In retrieval phase, we set K=2048 in our uni-retrieval, and also compare other choices in our analysis. All experiments are run on a single A100 GPU. Our codes will be open-sourced.

E More Experimental Analysis

E.1 BEIR Details

Please refer to Table 9 for detailed results on BEIR benchmark with 12 datasets.

E.2 Ablation of Learning Objectives

Learning of Learning Strategy. Furthermore, we conduct another ablation study on the learning strategies (§3.3) in Table 10. This is performed at the continual training stage. The table shows that, ablating the negative-bridged self-adversarial (self-adv) and the agreement-based self-regularization (self-reg) has a minor effect on lexicon-based retrieval but is remarkable on dense-vector one. This is because the former is already far stronger than the latter. Thereby, both self-adv and self-reg can be regarded as a sort of (self-)distillation from lexicon knowledge from a well-trained language model to dense semantic representation. We will dive into the self-reg in the following to seek for a better learning strategy, especially for the lexicon-based retrieval. In addition, we also observed that the proposed self-learning strategies (i.e., self-adversarial and self-regularization) mainly contribute to dense-vector retrieval (+0.6% and 0.3% MRR@10, respectively) but only bring limited performance improvement for lexicon-based method (+0.1% and 0.1% MRR@10, respectively). The main reasons are two-fold: i) Verified in (Formal et al., 2021a; Hofstätter et al., 2021), lexicon-based methods consistently outperform dense-vector methods in ad-hoc retrieval as lexicon-overlap serves as an important feature in relevance calculations. Therefore, the improvement mainly falls into the dense-vector part via knowledge distillation from the lexicon-based part. ii) Meantime, the common knowledge distillation schema is from a strong teacher to a weak student, e.g., cross-encoder reranker v.s. bi-encoder retriever with a 5∼10% performance gap in ad-hoc retrieval scenarios (Zhang et al., 2022; Ren et al., 2021b). In contrast, the participants (UnifieR-dense & -lexicon) of our self-learning have similar performance (gap <1%), making the improvement limited.

Narrowing Self-regularization Targets. By default, we apply the self-reg to hard negatives from both representing modules, which intuitively is a compromise choice for both. To explore if the self-reg can push one of them to an extreme, we conduct exploratory settings for the self-reg in Ta-

Methods	Sparse				Dense					
	BM25	DT5Q	UniCOIL	ColBERT	DPR	ANCE	GenQ	TAS-B	Contriever	Ours
TREC-COVID	65.6	71.3	59.7	67.7	33.2	65.4	61.9	48.1	59.6	**71.5**
NFCorpus	32.5	32.8	32.5	30.5	18.9	23.7	31.9	31.9	32.8	**32.9**
NQ	32.9	39.9	36.2	**52.4**	47.4	44.6	35.8	46.3	49.8	51.4
HotpotQA	60.3	58.0	64.0	59.3	39.1	45.6	53.4	58.4	63.8	**66.1**
FiQA	23.6	29.1	27.0	31.7	11.2	29.5	30.8	30.0	**32.9**	31.1
ArguAna	31.5	34.9	35.5	23.3	17.5	41.5	**49.3**	42.9	44.6	39.0
Tóuche-2020	**36.7**	34.7	25.9	20.2	13.1	24.0	18.2	16.2	23.0	30.2
DBPedia	31.3	33.1	30.2	39.2	26.3	28.1	32.8	38.4	**41.3**	40.6
Scidocs	15.8	16.2	13.9	14.5	7.7	12.2	14.3	14.9	**16.5**	15.0
Fever	75.3	71.4	72.3	**77.1**	56.2	66.9	66.9	70.0	75.8	69.6
Climate-FEVER	21.3	20.1	15.0	18.4	14.8	19.8	17.5	22.8	**23.7**	17.5
SciFact	66.5	67.5	67.4	67.1	31.8	50.7	64.4	64.3	67.7	**68.6**
BEST ON	1	0	0	2	0	0	1	0	4	4
AVERAGE	41.1	42.4	40.0	41.8	26.4	37.7	39.8	40.4	44.3	**44.5**

Table 9: Detailed results (NDCG@10) on BEIR benchmark.

Methods	Lexicon-based		Dense-vector	
	M@10	R@100	M@10	R@100
UnifieR	39.7	91.2	38.8	90.3
◇ w/o Self-adv	39.6	91.5	38.2	90.3
◇ w/o Self-adv&-reg	39.5	91.3	37.9	90.1

Table 10: Ablation of our learning strategy at *continual training* stage on MS-Marco Dev.

Methods	Lexicon-based		Dense-vector	
	M@10	R@100	M@10	R@100
UnifieR	39.7	91.2	38.8	90.3
◇ Self-reg on $\mathbb{N}^{(den)}$ only	39.5	91.0	38.3	90.0
◇ Self-reg on $\mathbb{N}^{(lex)}$ only	**39.9**	91.4	38.5	90.3

Table 11: Effect of our self-regularization's targets on MS-Marco.

ble 11. First, applying self-reg to the negatives from dense-vector module even makes the whole framework degenerate. It is likely attributed to the dense-vector receiving less supervisions from the lexicon part, which supports the above claim that the self-reg can be seen as a distillation from lexicons to dense embedding. On the other hand, when applying self-reg only to the negatives by the lexicon part, the lexicon-based model achieves a new level with 39.9% MRR@10, which is superior to a single-representing retriever. This supports the idea of instance-dependent prompt learning (mentioned in §3.2), where all modules work together for better lexicon-weighting representations.

E.3 Comparison to Retrieval&Rerank

Without distillations or co-teaching from a reranker, our retriever can be competitive with some state-of-the-art *retrieval & rerank* methods as in Table 12.

Retriever	**Reranker**	M@10
RepBERT	RepBERT	37.7
ME-HYBRID	ME-HYBRID	39.4
RocketQA	RocketQA	40.9
RocketQAv2	RocketQAv2	41.9
Ours (retriever-only)		40.7

Table 12: Comparisons with *retrieval&rerank* pipelines.

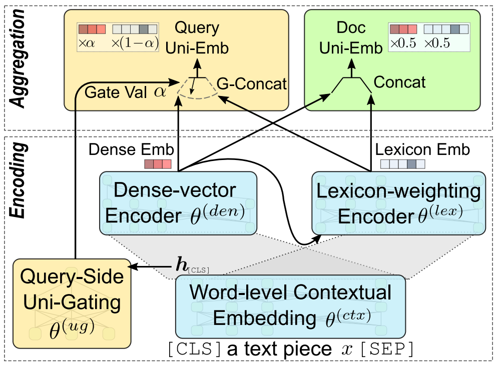

Figure 7: Equipping UnifieR with query-side gating.

Note that the reranker is extremely costly as it is applied to every query-document text concatenation, instead of counterpart-agnostic representations from a bi-encoder.

E.4 Illustration of Query-side Gating

We illustrate the query-side gating mechanism in Figure 7, which leverages a gating mechanism to dynamically combine lexicon and dense embeddings only at the query side.

E.5 Reranker-taught Pipeline

In contrast to the normal two-stage training pipeline in Figure 3, we present our reranker-taught pipeline in Figure 8.

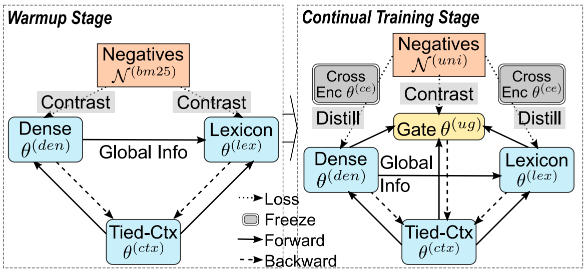

Figure 8: Reranker-taught UnifieR by knowledge distillation.

F Qualitative Analysis

F.1 Case Study

As shown in Table 13, we list two queries coupled with the ranking results from five retrieval systems. Those are from three groups, i.e., i) previous state-of-the-art dense-vector and lexicon-based retrieval models, ii) the dense-vector and lexicon-based retrieval modules from our UnifieR, and iii) uni-retrieval scheme by our UnifieR.

As demonstrated in the first query of the table, 'Indep-lex' achieves a very poor performance, where the positive passage is ranked as 94. Via exhibiting its top-1 passage, the error is possibly caused by the confusion between the 'weather' for a specific day and 'weather' for a period (i.e., climate). This is because the 'weather' as a pivot word in both contexts receives large weights, making the distinguishment very hard. Although our $\text{UnifieR}_{\text{lex}}$ can lift the positive from 94 to 3 by our carefully designed unified model, it still suffers from confusion. Meantime, it is observed that both dense-vector methods perform well since they rely on latent semantic contextualization, less focusing on a specific word.

As shown in the second query of the table, the strange word, 'idiotsguides' makes both dense-vector models less competent. On the contrary, the lexicon-based method can handle this case perfectly. It is still noteworthy that our $\text{UnifieR}_{\text{den}}$ can also outperform the vanilla one, 'Indep-den', by lifting 31 (41→10) ranking position. This is attributed to our consistent feature learning, which bridges the gap of heterogeneity between dense-vector and lexicon-based retrieval.

These two cases also support the previous claim that the two representing ways can provide distinct views of query-document relevance. Furthermore, despite varying performance across different paradigms, our uni-retrieval scheme consistently performs well as it is an aggregation of both.

F.2 Error Analysis.

As shown in Table 14, we show two representative cases which our proposed method cannot handle.

i) *query hubness*: The first case shows a query that cannot be tackled by our UnifieR in any retrieval paradigm. However, it is observed that the top-1 passage retrieved by our model can also be considered as a positive passage, which can answer the query '*what is a dvt*'. These negative passages for the query are false negatives, which are brought by the limited crowd-sourcing labeling procedure. Therefore, the poor performance of our model instead proves that our model is more robust, whereas the independent learning model is overfitting to its false negatives, resulting in seemingly good outputs.

ii) *Insufficient representation ability*: The second case lists the top-retrieved passages for all five retrieval systems. It is shown that compared to independently learned retrieval models (i.e., 'Indep-den' and 'Indep-lex'), our unified models even perform worse and retrieve less relevant passages (refer to $\text{UnifieR}_{\text{den}}$'s 1st). An interesting point is that the 'Ups'-related passage is retrieved by our $\text{UnifieR}_{\text{den}}$ since 'upsell' is tokenized as 'ups' and '##ell'. This is highly likely since one single model is required to serve dual representing modules, compromising its representation ability.

Meantime, our uni-retrieval can still improve the ranking performance by combining both of the representing worlds.

F.3 Limitation

The main limitations of this work are i)*PLM Compatibility*: due to the special encoder design, UnifieR can only be initialized from a limited number of pre-trained models, and ii) *Additional Infrastructure*: in spite of the almost same retriever latency as traditional lexicon-based retrieval, UnifieR requires extra computation infrastructure for indexing and storing both dense and sparse embeddings of all documents in the collection.

Query	*ID:1088347//* weather in new york city ny
Passage+	*ID:7094280// Title: - Body:* New York, NY - Weather forecast from Theweather.com. Weather conditions with updates on temperature, humidity, wind speed, snow, pressure, etc. for New York, New York Today: Cloudy skies with light rain, with a maximum temperature of 72C and a minimum temperature of 52C.
Rank	Indep-den: 1; Indep-lex: 94; UnifieR$_{\text{den}}$: 1; UnifieR$_{\text{lex}}$: 3; UnifieR$_{\text{uni}}$: 1
Retrieved	**Indep-lex's 1st.** *ID:65839// Title:* New York City - Best Time To Go & When to Go *Body:* Weather: Spring in New York City is the best time to be in the city, without doubt. Spring usually means less humidity and temps between 50-80 degrees, though June occasionally sees a 90 degree day. An occasional humidity soaked heat wave can strike, but it usually feels nice the first time around. **Indep-lex's 2nd.** *ID:4835773// Title:* Climate of New York *Body:* Weather: Unlike the vast majority of the state, New York City features a humid subtropical climate (Koppen Cfa). New York City is an urban heat island, with temperatures 5-7 degrees Fahrenheit (3-4 degrees Celsius) warmer overnight than surrounding areas. In an effort to fight this warming, roofs of buildings are being painted white across the city in an effort to increase the reflection of solar energy, or albedo. **UnifieR$_{\text{lex}}$'s 1st.** *ID:65839// Title:* New York City - Best Time To Go & When to Go *Body:* Weather: Spring in New York City is the best time to be in the city, without doubt. Spring usually means less humidity and temps between 50-80 degrees, though June occasionally sees a 90 degree day. An occasional humidity soaked heat wave can strike, but it usually feels nice the first time around. **UnifieR$_{\text{lex}}$'s 2nd.** *ID:8819213// Title:* New York City - Best Time To Go & When to Go *Body:* Weather: Spring in New York City is the best time to be in the city, without doubt. Spring usually means less humidity and temps between 50-80 degrees, though June occasionally sees a 90 degree day.
Query	*ID:391101//* idiotsguides tai chi
Passage+	*ID:7668258// Title: - Body:* Bill is the author of The Complete Idiot's Guide to T'ai Chi & Qigong (4th edition), and his newest upcoming books, The Tao of Tai Chi, and The Gospel of Science, in which he paints a vision of vast global benefit as mind-body sciences spread across the planet.
Rank	Indep-den: 41; Indep-lex: 1; UnifieR$_{\text{den}}$: 10; UnifieR$_{\text{lex}}$: 1; UnifieR$_{\text{uni}}$: 1
Retrieved	**Indep-den's 1st.** *ID:1603205// Title: - Body:* Tai chi. Tai chi (simplified Chinese: ; traditional Chinese: ; pinyin: chi, an abbreviation of ;is an internal Chinese martial art (Chinese: ; pinyin:) practiced for both its defense training and its health benefits. **Indep-den's 2nd.** *ID:3449438// Title:* Tai chi: A gentle way to fight stress *Body:* Tai chi is an ancient Chinese tradition that, today, is practiced as a graceful form of exercise. It involves a series of movements performed in a slow, focused manner and accompanied by deep breathing. Tai chi, also called tai chi chuan, is a noncompetitive, self-paced system of gentle physical exercise and stretching. **UnifieR$_{\text{den}}$'s 1st.** *ID:2294942// Title:* WHAT IS TAI CHI? *Body:* The Chinese characters for Tai Chi Chuan can be translated as the 'Supreme Ultimate Force'. The notion of 'supreme ultimate' is often associated with the Chinese concept of yin-yang, the notion that one can see a dynamic duality (male/female, active/passive, dark/light, forceful/yielding, etc.) in all things. **UnifieR$_{\text{den}}$'s 2nd.** *ID:3449442// Title:* What is Tai Chi? *Body:* What is Tai Chi? In China, and increasingly throughout the rest of the world, tai chi is recognized for its power to instill and maintain good health and fitness in people of all ages. Tai chi aims to bring balance to body, mind and spirit through specifically designed movements, natural breathing and a calm state of mind. It is easily recognized by its slow, captivating and mesmerizing movements. It represents a way of life, helping people meet day to day challenges while remaining calm and relaxed.

Table 13: Case study on MS-Marco Dev set. 'Passage+' denotes positive passage of the corresponding query. 'Indep-den' denotes a well-trained state-of-the-art dense-vector retrieval model with static hard negatives (i.e., coCondenser (Gao and Callan, 2021b), M@10=38.2) while 'Indep-lex' denotes a well-trained state-of-the-art lexicon-based retrieval model with static hard negatives (i.e., SPLADE (Formal et al., 2022), M@10=38.5).

Query	*ID:682365//* what is a dvt?
Passage+	*ID:7544458// Title:* Deep vein thrombosis *Body:* For other uses, see DVT (disambiguation). Deep vein thrombosis, or deep venous thrombosis (DVT), is the formation of a blood clot (thrombus) within a deep vein, most commonly the legs. Nonspecific signs may include pain, swelling, redness, warmness, and engorged superficial veins.
Rank	Indep-den: 3; Indep-lex: 2; UnifieR$_{\text{den}}$: 12; UnifieR$_{\text{lex}}$: 11; UnifieR$_{\text{uni}}$: 9
Retrieved	**UnifieR$_{\text{den}}$'s 1st.** *ID:5404002// Title:* Definition of 'DVT' *Body:* Definition of 'DVT'. DVT is a serious medical condition caused by blood clots in the legs moving up to the lungs. DVT is an abbreviation for 'deep vein thrombosis'. The results from one of the largest studies yet carried out leave little doubt that DVT is caused by flying. **UnifieR$_{\text{lex}}$'s 1st.** *ID:8492523// Title:* What Is DVT? *Body:* What Is DVT? Deep vein thrombosis is a blood clot that forms inside a vein, usually deep within your leg. About half a million Americans every year get one, and up to 100,000 die because of it. The danger is that part of the clot can break off and travel through your bloodstream. **UnifieR$_{\text{uni}}$'s 1st.** *ID:8492523// Title:* What Is DVT? *Body:* What Is DVT? Deep vein thrombosis is a blood clot that forms inside a vein, usually deep within your leg. About half a million Americans every year get one, and up to 100,000 die because of it. The danger is that part of the clot can break off and travel through your bloodstream.
Query	*ID:1029124//* what is upsell
Passage+	*ID:7220016// Title:* Upselling *Body:* What is Upselling? Upselling is a sales technique aimed at persuading customers to purchase a more expensive, upgraded or premium version of the chosen item or other add-ons for the purpose of making a larger sale. eCommerce businesses often combine upselling and cross-selling techniques in an attempt to increase order value and maximize profit.
Rank	Indep-den: 3; Indep-lex: 1; UnifieR$_{\text{den}}$: 11; UnifieR$_{\text{lex}}$: 9; UnifieR$_{\text{uni}}$: 8
Retrieved	**Indep-den's 1st.** *ID:6288350// Title: - Body:* If you improve inventory turn but pay more. in freight costs for multiple shipments or your warehouse has to increase their variable costs. to process the additional shipments, the net result may be a loss. 4. An upsell feature on the web is a visual reminder of how much money a customer can. spend before the next shipping & handling threshold is met. King Arthur Flour is an. excellent example of how to improve upsell and increase items per order. Showing the. amount available, relevant. choices within the price. **Indep-lex's 1st.** *ID:7220016// Title:* Upselling *Body:* What is Upselling? Upselling is a sales technique aimed at persuading customers to purchase a more expensive, upgraded or premium version of the chosen item or other add-ons for the purpose of making a larger sale. eCommerce businesses often combine upselling and cross-selling techniques in an attempt to increase order value and maximize profit. **UnifieR$_{\text{den}}$'s 1st.** *ID:8487388// Title:* Acronyms &Abbreviations *Body:* Ups is an open source source-level debugger developed in the late 1980s for Unix and Unix-like systems, originally developed at the University of Kent by Mark Russell. It supports C and C++, and Fortran on some platforms. The last beta release was in 2003. **UnifieR$_{\text{lex}}$'s 1st.** *ID:4754301// Title:* Upselling: 75 Strategies, Ideas and Examples *Body:* . Upsell Drip Campaign to upsell B2B/Saas solutions. What is it? The upsell for B2B/Saas solutions email is meant to add to the services. These emails offer premium services or upgrades for users on paying, free or trial accounts. When is it sent? Upsell emails for B2B/Saas solutions are meant to extend the usability and functionality of the software. **UnifieR$_{\text{uni}}$'s 1st.** *ID:4754301// Title:* Upselling: 75 Strategies, Ideas and Examples *Body:* . Upsell Drip Campaign to upsell B2B/Saas solutions. What is it? The upsell for B2B/Saas solutions email is meant to add to the services. These emails offer premium services or upgrades for users on paying, free or trial accounts. When is it sent? Upsell emails for B2B/Saas solutions are meant to extend the usability and functionality of the software.

Table 14: Error analysis on MS-Marco Dev set.