1 INTRODUCTION

Large-scale passage retrieval [2] aims to fetch relevant passages from a million- or billion-scale collection for a given query to meet users’ information needs, serving as an important role in many downstream applications including open domain question answering [20], search engine [56], and recommendation system [54], etc. Recent years have witnessed an upsurge of interest and remarkable performance of dense passage retrievers on first-stage retrieval. Built upon powerful pre-trained language models (PLM) [9, 30, 36], recent works have converted the original text into a latent semantic space are abstractive and condensed, exposing systems to a risk that pivot phrases and mentions may be overlooked and thus leading to sub-optimal efficacy. For example, DPR [20] didn’t regard “Thoros of Myr” as an entity mention in the query “Who plays Thoros of Myr in Game of Thrones?”. Analogously, given the query “What is an active margin”, ANCE [49] overlooked the “active margin” as an entire local salient phrase and hence retrieved passages related to the financial term “margin”. As a remedy, prior works resort to either coupling a dense retriever with the term matching scores (e.g., TF-IDF, BM25) [7, 14, 26, 33] or learning BM25 ranking into a dense model as additional features to complement the original one [4]. But, these approaches are limited by superficial combinations and almost unlearnable BM25 scoring.

To circumvent demerits from the superficial hybrid or learning with inferior lexicon-based representations upon PLM, we propose a brand-new lexicon-enlightened dense (LED) retriever learning framework to inject rich lexicon information into a single
dense encoder, while keeping its sequence-level semantic representation capability. Instead of prevailing BM25 as lexicon-rich sources, we propose to leverage the recently advanced lexicon-centric representation learning model transferred from large-scale masked language modeling (MLM), and attempt to align a dense encoder with two brand-new weakened distilling objectives. On the one hand, we present lexicon-augmented contrastive learning that incorporates the hard negatives provided by lexicon-aware retrievers for contrastive training. Intuitively, the negatives given by the lexicon-aware models could be regarded as adversarial examples to challenge the dense one, so as to transfer lexical knowledge to the dense model. On the other hand, inspired by previous work [1, 8], we propose a pair-wise rank-consistent regularization as a weak supervision to guide dense model’s behavior incline to the lexicon-aware ones. Compared to distribution regularization such as KL-divergence [53] and strict fine-grained distillation like Margin-MSE [16], LED provides weak supervision signals from the lexicon-aware retrievers, leading to desirable partial knowledge injection while maintaining the dense retriever’s own properties.

We evaluate our method on three real-world human-annotated benchmarks. Experimental results show that our methods consistently and significantly improve dense retriever’s performance, even outdoing its teacher. Notably, these significant improvements are brought by the supervision of a performance-comparable lexicon-aware retriever. Besides, a detailed analysis of retrieval results shows that our knowledge distillation strategies indeed equip the dense retriever with lexicon-aware capabilities. Lastly, we show our lexicon-aware distillation strategies are compatible with the standard ranker distillation, achieving further improvement and a new state-of-the-art performance.

Our contributions are three-fold: (1) We consider improving the dense retriever by imitating the retriever based on the lexicon-aware representation model upon PLM; (2) We propose two strategies including lexicon-augmented contrastive training and pair-wise rank-consistent regularization to inject lexical knowledge into the dense retriever; (3) Evaluation results on three benchmarks show that our method brings consistent and significant improvements to the dense retriever with a comparable lexicon-aware retriever as a teacher and a new state-of-the-art performance is achieved.

2 RELATED WORK

Current passage retrieval systems are widely deployed as retrieve-then-rank pipelines [18, 56]. The first-stage retriever (i.e., dual-encoder) [27, 29, 35, 38, 47, 49] selects a small number of candidate passages (usually at most thousands) from the entire collection, and the second-stage ranker (i.e., cross-encoder [55]) scores these candidates again to provide a more accurate passages order. In this paper, we focus on enhancing the first-stage retriever.

**Dense Retriever.** Built upon Pre-trained Language Models [9, 30], dense retriever [20, 35] is to capture the semantic meaning of an entire sequence by encoding sequential text as a continuous representation into a low-dimensional space (e.g., 768). In this way, the dense retriever could handle vocabulary and semantic mismatch issues within the traditional term-based techniques like BM25 [40]. To train a better dense retriever, various techniques are proposed for providing hard negatives including reusing in-batch negatives [20, 31, 35], iteratively sampling [49], mining by a well-trained model or dynamic sampling [52, 53], and denoising by cross-encoder [35]. To build retrieval-specific pre-trained language models, Lee et al. [23] proposed an unsupervised pre-training task, namely Inverse Cloze Task (ICT), Gao and Callan [12] decoupled model architecture during pre-training and further designed corpus-level contrastive learning [13] for better passage representations.

**Lexicon-Aware Retriever.** Another paradigm of work [11, 14, 26] takes advantage of strong PLMs to build lexicon-aware sparse retrievers by term-importance [7, 26] and top coordinate terms [10, 11]. These models have lexical properties and could be coupled with inverted indexing techniques. Based on contextualized representation generated by PLMs [9], Dai and Callan [7] learned to estimate individual term weights, Mallia et al. [33] further optimized the sum of query terms weights for better term interaction, COIL-series works used token-level interactions on weight vector [14] or scalar [26] to obtain exact word matching scores, and Formal et al. [10, 11] trained a retriever encoding passages as vocabulary-size highly sparse embeddings. Recently, Chen et al. [4] trained a PLM-based retriever from scratch with data generated by BM25. The trained lexicon-aware retriever could encode texts as low-dimensional embeddings and have identical lexical properties and comparable performance with BM25.

**Hybrid Retriever.** Arguably, dense sequence-level retrievers and lexicon-aware retrievers have distinctive pros and are complementary to each other. This fact triggered researchers to investigate how to combine their advantages, such as direct score aggregation [22], weighted sum [25, 26, 28, 31, 46], multiplication [50], or concatenation [4, 32, 43] in an ensemble system. The above hybrid retrievers require two dense and lexical retrievers to first compute individually and then combine their results in feature-level [4, 32, 43] or relevance-score-level [22, 25, 26, 28, 31, 46, 50] to obtain a final result. In contrast, our method only needs one model to achieve both dense and lexicon-aware retrieval behaviors, significantly decreasing the memory footprint and inference speed meanwhile providing a more in-depth fusion of lexicon-aware and dense retrieval views.

**Knowledge Distillation.** Cross-encoder empirically outperforms dual-encoder for it inputs query and passage as a whole, so that attention mechanism will be applied between them, leading to in-depth token-level interactions. Its superior performance motivates many works [16, 53] to enhance dual-encoders by knowledge distillation from cross-encoder. KL-Divergence, which minimizes the distances of distributions between teacher and student, has proven effective in many works [42, 53]. Margin-MSE [16] aims to minimize the difference of margins in two passages, and it’s been applied in later works [10, 17]. Reddi et al. [37] used the teacher’s top passages as positive examples to teach students point-wisely. ListNet [3, 48] ensured the consistency of list-wise ranking order by minimizing the difference in score distributions over passages.

The above methods are designed for the cross-encoder teacher that is much stronger than the dense student. But in our situation, the lexicon-aware teacher can only achieve comparable performance with a dense retriever. In practice (i.e., Tab. 2), we find that existing distillation methods may not be perfect choices in our
setting. Therefore, we propose two novel strategies, namely lexicon-augmented contrastive training and pair-wise rank-consistent regularization to transfer lexical knowledge.

3 METHODOLOGY

We first introduce the task formalization, general training framework, and retrieval architectures in Sec. 3.1. Then we present our Lexical-Enlightened Dense (LED) retriever in Sec. 3.2.

3.1 Preliminary

Task Definition. In the first-stage retrieval, given a query \( q \), a retriever is required to fetch top-k relevant passages from a million-even billion-scale passage collection \( C \). Due to the efficiency requirement, dual-encoder architecture is widely applied in this task for its lightweight metric calculation. Formally, dual-encoder represents text \( x \) (could be query \( q \) or passage \( p \)) to \( d \)-dimensional embeddings, i.e.,

\[
x = \text{Dual-Enc}(x; \theta) \in \mathbb{R}^d,
\]

(1)

where \( \theta \) could be dense retriever (\( \theta_{\text{den}} \)) or lexicon-aware retriever (\( \theta_{\text{lex}} \)). With separately encoded query \( q \) and passage \( p \), we could calculate the relevance score via dot product for retrieval, i.e.,

\[
R(q; p; \theta) = q^T p.
\]

(2)

The dual-encoder architecture and lightweight dot product evaluation enable us to encode and index all passages in the collection \( C \) beforehand, so we only need to encode the given query for online retrieval, achieving more efficiency.

Learning Framework for Retriever. To train the dual-encoder \( \theta \), we utilize contrastive learning following previous works [15, 49]. Specifically, with a given query \( q \), a labeled positive passage \( p^* \), and negative passages \( N \), contrastive loss can be applied to optimize the dual-encoder \( \theta \) by maximizing the relevance of the \( q \) and \( p^* \) while minimizing that of \( q \) and \( p \in N \), i.e.,

\[
\mathcal{L}_\theta^{cl} = - \log \frac{\exp(R(q, p^*; \theta))}{\sum_{p \in \{p^*\} \cap N} \exp(R(q, p; \theta))},
\]

(3)

where negative passage set \( N \) can be generated from top-ranked non-answer passages in retrieval results of BM25 model [34] or a trained retrievers [52, 53], i.e.,

\[
N = \{ p \mid p \sim P(C \setminus \{p^*\} \mid q; \theta_{\text{samp}}) \},
\]

(4)

where \( P \) is a probability distribution over \( C \), which can be defined as non-parametric (e.g., \( \theta_{\text{samp}} = \emptyset \)) or parametric (e.g., \( \theta_{\text{samp}} \neq \emptyset \)).

Dense & Lexicon-Aware Retrievers. Both dense retriever (\( \theta_{\text{den}} \)) and lexicon-aware retriever (\( \theta_{\text{lex}} \)) follow dual-encoder architecture and the encoders are built upon PLMs like BERT [9]. Precisely, a PLM (\( \theta_{\text{plm}} \)) encodes a given text (i.e., query \( q \) or passage \( p \)), \( x = \{t_1, t_2, ..., t_n\} \), to contextualized embeddings, i.e.,

\[
H^F = \text{PLM}(x; \theta_{\text{plm}})
\]

\[
= \text{PLM}\left([\text{CLS}], t_1, t_2, ..., t_n, [\text{SEP}], \theta_{\text{plm}}\right),
\]

(5)

eventually \( H^F = [h^F_{\text{CLS}}, h^F_1, ..., h^F_n, h^F_{\text{SEP}}] \). [CLS] and [SEP] are special tokens designed for sentence representation and separation by recent PLMs [9, 30]. Dense retriever [35, 49] represents text by using the embedding of special token [CLS] (i.e., \( h^F_{\text{CLS}} \)) as follows,

\[
x^{\text{den}} = \text{Dual-Enc}(x; \theta_{\text{den}}) = \text{CLS-Pool}(H^F),
\]

(6)

where \( \theta_{\text{den}} = \theta_{\text{plm}} \) with no additional parameters.

For lexicon-aware retriever, we adopt SPLADE [10] which learns to predict the weights of terms in PLM vocab for each token in the input \( x \) by the Masked Language Modeling (MLM) layer and sparse regularization, then max-pooling these weights into a discrete text representation after log-saturation. Formally, with \( H^F \) encoded by the PLM (\( \theta_{\text{plm}} \)), a MLM layer (\( \theta_{\text{mlm}} \)) linearly transform it into \( \tilde{H}^F \),
then term lex weight representation of x could be obtained as follows,
\[ x^{\text{lex}} = \text{Dual-Enc}(x; \theta^{\text{lex}}) = \text{MAX-Pool}(\log(1 + \text{ReLU}(W^x \cdot \hat{H}^x))), \] (7)

where \( W^x \in \mathbb{R}^{V \times e} \) is the transpose of the input embedding matrix in PLM as the MLM head, and the \( \hat{\theta}^{\text{lex}} = \{(\theta^{\text{lm}}, \theta^{\text{mlm}}, W^x)\} \).

The dense encoder represents texts as global sequence-level embeddings and is good at global semantic matching, while the lexicon-aware encoder represents local term-level embeddings and handles salient phrases and entity mentions well. Both encoders can be optimized with the Eq. 3.

### 3.2 Lexical Enlightened Dense Retriever

Fig. 1 illustrates the training workflow of our LED retriever. Specifically, we follow a two-stage training procedure. In the Warmup stage, we independently train the dense and lexicon-aware retrievers by Eq. 3, both with BM25 negatives (\( N^{\text{bm25}} \)). This stage ends up with two retrievers, namely the Lexical Warm-up (\( \theta^{\text{lex1}} \)) and the Dense Warm-up (\( \theta^{\text{den1}} \)).

Then, we sample negative passages (\( N^{\text{lex1}} \)) with the Lexical Warm-up checkpoint (\( \theta^{\text{lex1}} \)) for the second stage, namely Continual Training stage. With the fixed negative passages (\( N^{\text{lex1}} \)), we continually train the lexical retriever initialized from the warm-up checkpoint (\( \theta^{\text{lex1}} \)) by Eq. 3. After the second stage, we could obtain the model named Lexical (\( \theta^{\text{lex2}} \)), which plays a role of a teacher for later lexical knowledge teaching.

With a well-trained lexicon-aware teacher (\( \theta^{\text{lex2}} \)) and dense student after warming up (\( \theta^{\text{den1}} \)), we enlighten the student by transferring knowledge from the teacher. The knowledge transfer is achieved from two perspectives – 1) a lexicon-augmented contrastive objective to challenge the dense encoder and 2) a rank-consistent regularization to make the dense model’s behavior inclined to its lexicon-aware teacher. We will detail the two objectives in the following paragraphs.

#### Lexicon-Augmented Contrastive

Following previous work [52], we use the dense negatives (\( N^{\text{den1}} \)) sampled by the Dense Warm-up (\( \theta^{\text{den1}} \)) to boost dense retrieval. Meanwhile, inspired by Chen et al. [4] who trained a lexical retriever with negatives provided by term-based techniques such as BM25, we try to enhance the dense retriever from the negatives augmentation perspective.

Considering the similar backbone (i.e., both are fine-tuned on PLMs) and the same optimization objectives (i.e., Eq. 3) of both dense and lexicon-aware retrievers, their significant differences in retrieval behaviors may partially stem from training with different negative passages. So intuitively, dense one can use the lexical negatives (\( N^{\text{lex1}} \)) to partially imitate the training process of the lexical teacher (\( \theta^{\text{lex2}} \)), thus learning lexicon-aware ability. One step further, we use the lexical negatives (\( N^{\text{lex2}} \)) for learning more and harder lexical knowledge. Meanwhile, compared to the dense negatives (\( N^{\text{den1}} \)), these lexical negatives (\( N^{\text{lex1}} \) and \( N^{\text{lex2}} \)) can provide more diverse examples and could be regarded as adversarial examples to challenge the dense retriever for robust retriever training. Formally, the lexicon-augmented contrastive loss for LED is,
\[ L^{\text{cl}}_{\theta^{\text{led}}} = -\log \frac{\exp(R(q, p^{*}; \theta^{\text{led}}))}{\sum_{p \in [p^{*}] \cap N^{\text{mix}}} \exp(R(q, p; \theta^{\text{led}}))}, \] (8)

where \( N^{\text{mix}} = \{N^{\text{lex1}} \cap N^{\text{lex2}} \cap N^{\text{den1}} \} \).

#### Rank-Consistent Regularization

From the retrieval behavior perspective, for given query-passage pairs, we utilize the lexicon-aware teacher (\( \theta^{\text{lex2}} \)) to generate ranking pairs to regularize and guide LED’s retrieval behavior.

Specifically, given a query \( q \) and passages from \( \mathcal{D} = \{p \in \mathcal{P} \cap N^{\text{mix}} \} \), the Lexical (\( \theta^{\text{lex2}} \)) scores each query-passage pair (abbr. \( R(p; \theta^{\text{lex2}}) \)) with Eq. 2 and generate ranking pairs as follows,
\[ \mathcal{K}^{\text{led}} = \{(p_i, p_j) \mid p_i, p_j \in \mathcal{D}, R(p_i; \theta^{\text{lex2}}) > R(p_j; \theta^{\text{lex2}})\}; \] (9)

Then, a pair-wise rank-consistent regularization is employed to make the dense model’s behavior incline to the lexicon-aware one by minimizing the following margin-based ranking loss,
\[ L^{\text{ll}}_{\theta^{\text{led}}} = \frac{1}{|\mathcal{K}^{\text{led}}|} \sum_{(p_i, p_j) \in \mathcal{K}^{\text{led}}} \max[0, R(p_i; \theta^{\text{led}}) - R(p_j; \theta^{\text{led}})], \] (10)

where \( R(p; \theta^{\text{led}}) \) is the abbreviation of relevance \( R(q, p; \theta^{\text{led}}) \) calculated by LED (\( \theta^{\text{led}} \)) with Eq. 2. Compared to logits distillation with a list-wise KL-divergence loss, our training objective provides a weak supervision signal pair-wisely, thus keeping the effects of injecting lexical knowledge on dense properties at a minimum level.

Especially since we don’t punish the dense student as long as its ranking of a given pair is the same as the teacher, without strict requirements on the score gap like Margin-MSE [16]. Experiments in Tab. 2 demonstrate the merit of our method.

#### Training and Inference

To incorporate lexicon-aware ability while keeping its sequence-level semantic representation ability for passage retrieval, we combine contrastive loss (\( L^{\text{cl}}_{\theta^{\text{led}}} \)) in Eq. 8 and lexical learning loss (\( L^{\text{ll}}_{\theta^{\text{led}}} \)) in Eq. 10 to train our LED retriever (\( \theta^{\text{led}} \)) as follows,
\[ L_{\theta^{\text{led}}} = L^{\text{cl}}_{\theta^{\text{led}}} + \lambda L^{\text{ll}}_{\theta^{\text{led}}}, \] (11)

where \( \lambda \) is a hyperparameter to control how intensive the training inclines to transfer lexical-ware knowledge from the lexicon-aware teacher (\( \theta^{\text{lex2}} \)).

For inference, LED pre-computes the embeddings of all passages in the entire collection \( \mathcal{C} \) and builds indexes with FAISS [19]. Then LED encodes queries online and retrieves top-ranked \( k \) passages based on the relevance score.

#### Remark

Our framework injects lexicon-aware capability into a sequence-level representation model, showing two-fold advantages in comparison to previous methods superficially combining dense and lexicon-aware retrievers: 1) Compared to using two separate PLM-based dense and lexicon-aware retrievers [4, 26, 28, 44], our LED retriever could achieve hybrid retrieval results and comparable performances with only one model. We no longer need to maintain multiple index systems for both retrieval models in an ensemble system or to encode an online query twice with different retrievers, reducing memory footprint and inference time. 2) Compared to fusing PLM-based dense and traditional term-matching retrievers like BM25 [4, 24–26, 28, 31, 32, 43, 46], our single method could
achieve better results since the lexicon-aware capability of LED is learned from a strong retriever (shown in Tab. 3).

4 EXPERIMENTS

We evaluate our retriever on three public human-annotated real-world benchmarks, namely MS MARCO [34], TREC Deep Learning 2019 [6], and TREC Deep Learning 2020 [5]. MS-MARCO Dev has 6980 queries. TREC 2019 has 43 queries, and TREC 2020 includes 54 queries. In all three benchmarks, first-stage retrievers are required to fetch relevant passages from an 8-million-scale collection. We report MRR@10, Recall@50, and Recall@1000 for MS MARCO Dev, as well as NDCG@10 for both TREC Deep Learning 2019 and TREC Deep Learning 2020. For all three datasets, we use the official TREC evaluation files to conduct the evaluation protocol.5

4.1 Baselines

We compare with previous state-of-the-art baselines including traditional term-based techniques like BM25 [40], and dense [13, 17, 21, 28, 38, 39, 42, 49, 51–53] as well as lexicon-aware retrievers [4, 7, 10, 14, 26]. More details about the baselines are provided in Appendix A. We report the models used during the two-stage training pipeline for a detailed comparison. For lexicon-aware retrievers, we report the models after the warm-up training, namely LEX (Warm-up), and continual training, namely LEX (Continue). Note that LEX (Continue) is the lexical teacher used for teaching. Similarly, for dense retrievers, we show DEN (Warm-up) and DEN (Continue). Note that the DEN (Warm-up) is the dense student where our LED model starts from and DEN (Continue) is independently trained with the hard negatives provided by DEN (Warm-up) by Eq. 3. We also report the DEN (Continue) further enhanced with a strong ranker distillation (i.e., DEN (w/ RT)).

4.2 Implementation Details

All experiments run on 1 NVIDIA Tesla A100 GPU having 80GB memories with a fixed random seed. We train our models with mixed precision to speed up the training and meet the huge memory need. The training time will last about 32 hours. For the lexical teacher, we train a DistilBERT [41] following SPLADE-max [10].3 Following previous work [55], in the warm-up stage, we train the lexical retriever with batch size 48, 5 negatives for each query randomly sampled from BM25 negatives, and a learning rate $3e^{-5}$ for three epochs. In the second stage, we remain all hyperparameters unchanged except lower the learning rate to $2e^{-5}$ and use negative passages randomly sampled from the top 200 self-mining hard negatives. For dense retriever, we train coCondenser [13] checkpoint with batch size 16, 7 negatives per query, and a learning rate of $1e^{-3}$ for three epochs.4

Particularly, in the LED training stage, with other hyperparameters unchanged, we set the learning rate $5e^{-6}$ and randomly select 32 hard negatives from the mixture of each top 200 negatives mined by the warm-up dense student, warm-up lexical teacher, and final lexical teacher. The number of negatives per query 32 is selected from $\{8, 16, 24, 32\}$. The higher number of negatives per query indicates the more pair-wise ranks constructed by the lexical teacher, leading to more lexical knowledge transfer.

For rank-consistent regularization, we set loss weight $\lambda = 1.2$ after searching from $\{1.0, 1.2, 1.5, 1.8, 2.0\}$.5

4.3 Main Results

As shown in Tab. 1, we present the evaluation results on the aforementioned three public benchmarks.

Firstly, our LED retriever achieves comparable performance with state-of-the-art methods CoBERTv2 [42] and AR2 [53] on MS MARCO Dev, although both baselines are taught by the powerful ranking model (i.e., cross-encoder). After coupling with a similar ranker distillation, our LED retriever (i.e., LED (w/ RT)) can be further improved and meanwhile outperforms state-of-the-art baselines on all three datasets, showing the compatibility of distillation from lexicon-aware sparse retriever.5 Note that we neither use heavy ranker teacher in AR2 [53] nor multiple vector representation applied in CoBERTv2 [42].

Secondly, LED (w/ RT) achieves better performance than DEN (w/ RT) on all three datasets, demonstrating that our training method can transfer some complementary lexicon-aware knowledge not covered by the cross-encoder. The weak intensity of the supervision signal makes our lexical-enlightened strategy a promising plug-and-play technique for other dense retrievers.

Thirdly, our LED retriever taught by a smaller lexicon-aware retriever is similarly performant as the dense retriever taught by a strong cross-encoder (i.e., DEN (w/ RT)), showing the effectiveness of injecting lexical knowledge into the dense retriever. The reasons are two-fold: (1) The dual-encoder architecture of the lexicon-aware teacher enables the relevance calculation can be easily integrated into in-batch techniques to scale up the teaching amount. (2) More importantly, lexicon-aware retriever could provide self-mining hard negatives for more direct supervision while cross-encoder can only provide score distribution over given passages.

4.4 Further Analysis

Comparison of Teaching Strategies. Tab. 2 shows the comparison of our proposed pair-wise rank-consistent regularization with other teaching strategies. Filter means the negatives with high scores (i.e., false negatives) are filtered by LEX (Continue). The other three strategies (e.g., Margin-MSE, ListNet, and KL-Divergence) are borrowed from knowledge distillation in IR domain. From the table, we can find that all strategies can bring performance gain, even in an indirect way like Filter. This observation proves that learning from the lexicon-aware representation model leads to a better dense retriever. Also, our rank-consistent regularization outperforms other baselines on MRR@10 metric by a large margin, showing the superiority of our method. Besides, we can find that the point-wise objective (i.e., Margin-MSE) brings the least gain, followed by the list-wise objectives (i.e., ListNet and KL-Divergence) and our pairwise rank-consistent regularization brings the most significant gain.

We use KLDivergence to distill the ranker’s scores into the LED model, but we use ERNIE-2.0-base [45] instead of ERNIE-2.0-large in AR2. The KL loss is directly added to Eq. 11 during the training of LED.
Table 1: Experimental results on MS MARCO, TREC DL 2019 (DL’19), and TREC DL 2020 (DL’20) datasets (%). We mark the best results in bold and the second-best underlined. Numbers marked with ‘*’ mean that the improvement is statistically significant compared with the baseline (t-test with \( p \)-value < 0.05).

| Methods                  | PLM  | Ranker Taught | Multi Vector | MS MARCO Dev MRR@10 | R@50 | R@1k | NDCG@10 | DL’19 NDCG@10 | DL’20 NDCG@10 |
|--------------------------|------|---------------|--------------|---------------------|------|------|---------|----------------|----------------|
| BM25 [40]                | -    |               |              | 18.7               | 59.2 | 85.7 | 50.6    | 48.0           |                |
| DeepCT [7]               | BERT\_base |           |              | 24.3              | 69.0 | 91.0 | 55.1    | 55.6           |                |
| COIL-full [14]           | BERT\_base |           |              | 35.5              | -    | 96.3 | 70.4    |                |                |
| UniCOIL [26]             | BERT\_base |           |              | 35.2              | 80.7 | 95.8 | -       | -              |                |
| SPLADE-max [10]          | DistilBERT |         |              | 34.0              | -    | 96.5 | 68.4    | -              |                |
| DistilSPLADE-max [10]    | DistilBERT | ✓         |              | 36.8              | -    | 97.9 | 72.9    | -              |                |
| UniCOIL \_\_\_\_ [4]     | BERT\_base |           |              | 34.1              | 82.1 | 97.0 | -       |                |                |

**Lexicon-Aware Retriever**

**Dense Retriever**

**Our Models**

Table 2: Evaluation results of different teaching strategies on MS MARCO Dev (%). ‘***’ refers to statistical significance.

| Methods                  | MRR@10 | R@1k  |
|--------------------------|--------|-------|
| No Distillation          | 38.1   | 98.4  |
| Filter [35]              | 38.4   | 98.4  |
| Margin-MSE [16]          | 38.5   | 98.3  |
| ListNet [48]             | 38.7   | 98.2  |
| KL-Divergence [53]       | 39.0   | 98.4  |
| Ours                     | 39.6*  | 98.3  |

Comparison of Ensemble Retrievers. We are also curious whether our LED can improve the performance of ensemble retrievers. With LEX (Continue) (\( \theta_{lex2} \)) and LED (\( \theta_{led} \)), we simply use the summation of the normalized relevance scores of two retrievers, and then return a new order of retrieval results. Tab. 3 gives the evaluation results of our systems and other strong baselines reported in previous work [4, 26, 28]. Note that previous work [26, 28] utilized weighted score sum after hyper-parameter searching while we directly sum the normalized scores of two retrievers without any tuning. From the results in Tab. 3, we can observe:

1. Aligned with results in SPAR [4], the ensemble of two dense retrievers (i.e., DEN (Continue) + LED and DEN (Continue) + DEN (w/ RT)) is not as performant as that of one dense and one lexicon-aware retriever. In particular, the ensemble of two dense retrievers is even less competitive than a single LED or DEN (w/ RT). The results are rational because two base models have similar retrieval
Table 3: Comparison with Ensemble Systems on MS MARCO Dev (%). The first block results are copied from [4, 26, 28]. \( \Lambda \) [4] refers to a dense retriever trained with data generated by lexicon-based methods such as BM25 and UniCOIL. \( \ast \ast \) indicates statistical significance compared to their counterparts without our training strategies.

| Ensemble Systems                  | MRR@10 | R@50 | R@1k |
|----------------------------------|--------|------|------|
| TCT-ColBERT + BM25 [28]          | 36.9   | -    | -    |
| TCT-ColBERT + UniCOL [26]        | 38.2   | -    | -    |
| TCT-ColBERT + UniCOL [26]        | 37.8   | -    | -    |
| ANCE + BM25 [4]                  | 34.7   | 81.6 | 96.9 |
| RocketQA + BM25 [4]              | 38.1   | 85.9 | 98.0 |
| RocketQA + UniCOIL [4]           | 38.8   | 86.5 | 97.3 |
| RocketQA + BM25 \( \Lambda \) [4] | 37.9   | 85.7 | 98.0 |
| RocketQA + UniCOIL \( \Lambda \) [4] | 38.6 | 86.3 | 98.5 |
| DEN (Continue) + BM25            | 30.4   | 87.1 | 98.6 |
| DEN (Continue) + LED             | 39.3   | 86.9 | 98.5 |
| DEN (Continue) + DEN (w/ RT)     | 39.4   | 87.0 | 98.5 |
| DEN (Continue) + LEX (Continue)  | 40.4   | 88.4 | 98.7 |
| DEN (w/ RT) + LEX (Continue)     | 40.7   | 88.4 | 98.7 |
| LED + LEX (Continue)             | 40.9\( \ast \) | 88.3 | 98.6 |
| LED (w/ RT) + LEX (Continue)     | 41.1\( \ast \) | 88.5 | 98.7 |

Table 4: Ablation Study on MS MARCO Dev (%). Negs is short for negatives. \( \ast \ast \) indicates statistical significance.

| Retrievers                      | MRR@10 | R@1k |
|---------------------------------|--------|------|
| w/o Rank Regularization        | 39.6\( \ast \) | 98.3 |
| w/o LEX Continue Negs (\( \Lambda_{lex1} \)) | 37.9 | 98.5 |
| w/o LEX Warm-up Negs (\( \Lambda_{lex2} \)) | 39.4 | 98.3 |
| w/o LEX Mixed Negs (\( \Lambda_{lex1} \land \Lambda_{lex2} \)) | 39.2 | 98.4 |

Figure 2: Distributions of model prediction for DEN (Continue), LEX (Continue), and LED retrievers over MS MARCO Dev. For visual clarity, we use the query-passage pairs which the LEX and DEN predict discrepantly as data samples. The discrepancy is determined by that there is a \( a > 0.2 \) margin between their predicted scores normalized over passages retrieved for a query. To ensure diversity, we consider two normalization cases, LEX-biased pairs (i.e., LEX’s top-100) and LEX-unbiased pairs (i.e., LEX’s bottom-100 out of 1000).

Figure 3: (a) Effects of the regularization weight \( \lambda \) on MS MARCO Dev.

Figure 4: (a) Effects of \# Negatives per query on MS MARCO Dev. (b) Effects of the regularization weight \( \lambda \) on MS MARCO Dev.

**Impact of Different Components.** We conduct an ablation study to further investigate the impact of lexical hard negatives and rank-consistent regularization method. Tab. 4 reports the results of removing each component. We can observe that pair-wise rank-consistent regularization plays an important role in lexical learning because removing it will bring significant performance degradation on MRR@10 metric. In addition, we can find that both negatives provided by LEX (Warm-up) and LEX (Continue) are both helpful for the contrastive training of the dense retriever, and removing both of them results in a more obvious performance drop.
**Impacts of Hyperparameters.** We conduct extra experiments to explore the impact of hyperparameters on LED retriever training. Fig. 3(a) illustrates the impact of changing negative passages on the LED. We can observe that as the number of negative passages increases, the MRR@10 performance goes up and the R@1k performance reaches the peak when 16 and decreases gradually. The main table shows that Lexical is less performant than Dense at R@1k metric (98.0 < 98.4). So the trend of increasing the number of negatives proves that improving too much the lexical retriever will also be negatively influenced by the weakness of the teacher. These two trends indicate that, with more negatives, the teacher will construct more rank pairs for more lexical knowledge transfer. Fig. 3(b) shows the performance with regard to different regularization weights $\lambda$. It is observed that the performances don’t fluctuate significantly as the weight $\lambda$ changes, demonstrating the robustness of lexical enhancement strategies. Interestingly, the increase in MRR@10 comes with the drop in R@1k to some extent, once again showing that a well-enhanced LED also inherits the weakness of Lexical.

**Zoom-in Study of Retrieval Ranking.** Tab. 5 shows how the average rank of golden passages varies across different rank ranges, bucketed by LEX-predicted ranks of the golden passages. We can observe that: (1) More than 50% golden passages are ranked in the top 5 by the LEX, paving the way for good lexical teaching. (2) The average ranking of golden passages by LED is consistently improved until the top 100, which means approximately 90% of answers are ranked higher by the retriever after learning lexical knowledge, proving the effectiveness of our lexical knowledge transfer. Meanwhile, similar even more gain can also be observed in LED (w/ RT), once again proving that our method is complementary to distillation from a cross-encoder. (3) In the queries that the LEX performs unfavorably (i.e., ground truth ranked lower than 100), LED and LED (w/ RT) are negatively impacted by the lexicon-aware teacher’s mistakes. Interestingly, their original rankings of these ground truths are not very high, either. So these queries are intractable for both dense and lexicon-aware retrievers, which we leave for future work.

### 4.5 Case Study

Tab. 6 shows the three case queries with rankings of 4 retrievers. With lexicon-aware capability, LED and LED (w/ RT) could retrieve golden passages as the top-1 result like their teacher LEX. In particular, in the first case, DEN mismatches the “benefits” in the query with “advantages” in the passage since they are both positive words. On the contrary, the LEX and LED-series retrievers could exactly match the phrase “benefits of the Internet”. In the second query, the “Pencil tool” refers to a specific GUI prototyping tool (as highlighted in the positive passage). DEN misunderstands the mention “Pencil tool” in the query and returns passages about the vanilla pencil, which is non-relevant to the user’s intention. The above two cases show that retrievers with lexicon-aware capability (i.e., LEX, LED, LED (w/ RT)) could well capture the salient phrases and entity mentions, providing more precise retrieval results.

### 5 CONCLUSION

In this paper, we consider developing a lexicon-enlightened dense retriever by transferring knowledge from a lexicon-aware sparse representation model into a dense one. To achieve this end, we propose to enlighten a dense retriever from two aspects, namely the lexicon-augmented contrastive objective and the pair-wise rank-consistent regularization. Experimental results on three real-world retrieval benchmarks show that with a performance-comparable lexicon-aware representation model as the teacher, our strategies can improve a dense retriever consistently and significantly, even outdoing its teacher. Further extensive analysis and discussions demonstrate the effectiveness and compatibility of our training strategies, as well as the interpretability of the LED retriever.
A BASELINES

We compare with previous state-of-the-art baselines including traditional term-based techniques like BM25 [40], and dense as well as lexicon-aware retrievers. For lexicon-aware retrievers, DeepCT [7] was trained to predict term weights. COIL [14] used contextualized representation for exact term matching and UniCOIL [26] compressed vectors in COIL into scalars. DistilSPLADE-max and SPLADE-max [10] were both trained with Eq. 7 and the latter one was further enhanced by a cross-encoder. The UniCOIL Λ was a lexicon-aware model trained with UniCOIL’s top-ranked passages and negatives [4]. For dense retrievers, ANCE [49] selected hard training negatives from the entire collection. ADORE [52] used self-mining static negatives and then dynamic negatives. TAS-B [17] proposed balanced topic-aware negative sampling strategies for effective teaching. CL-DRD [51] taught the retriever in a curriculum learning fashion, starting from coarse-grained pair examples and progressing to fine-grained ones. ColBERTv1 [21] and ColBERTv2 [42] utilized late-interaction and the latter one further incorporates ranker distillation. TCT-ColBERT [28] utilized ColBERTv1 as the tightly-coupled teacher to enable in-batch distillation. The coCondenser [13] augmented MLM loss with contrastive learning and based a model architecture [12] with a decoupled sentence and token interaction. PAIR [38] introduced passage-centric loss to assist the contrastive loss and combine cross-encoder teaching. RocketQAv2 [39] utilized K-L divergence to align the list-wise distributions between retriever and ranker and proposed hybrid data augmentation. AR2-G [53] used an adversarial framework to train the retriever and ranker simultaneously. Notably, AR2 used a different Recall@N evaluation from the official TREC Recall@N. Therefore, we don’t report their Recall@N performances in Tab. 1.

6https://github.com/microsoft/AR2/tree/main/AR2