Data Augmentation for Sample Efficient and Robust Document Ranking

Anand, Abhijit; Leonhardt, Jurek; Singh, Jaspreet; Rudra, Koustav; Anand, Avishek

DOI
10.1145/3634911

Publication date
2024

Document Version
Final published version

Published in
ACM Transactions on Information Systems

Citation (APA)
Anand, A., Leonhardt, J., Singh, J., Rudra, K., & Anand, A. (2024). Data Augmentation for Sample Efficient and Robust Document Ranking. *ACM Transactions on Information Systems*, *42*(5), Article 119. https://doi.org/10.1145/3634911

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Data Augmentation for Sample Efficient and Robust Document Ranking

ABHIJIT ANAND, L3S Research Center, Germany
JUREK LEONHARDT, Delft University of Technology, The Netherlands and L3S Research Center, Germany
JASPREET SINGH, Independent Researcher, Germany
KOUSTAV RUDRA, Indian Institute of Technology Kharagpur, India
AVISHEK ANAND, Delft University of Technology, Netherlands

Contextual ranking models have delivered impressive performance improvements over classical models in the document ranking task. However, these highly over-parameterized models tend to be data-hungry and require large amounts of data even for fine-tuning. In this article, we propose data-augmentation methods for effective and robust ranking performance. One of the key benefits of using data augmentation is in achieving *sample efficiency* or learning effectively when we have only a small amount of training data. We propose supervised and unsupervised data augmentation schemes by creating training data using parts of the relevant documents in the query-document pairs. We then adapt a family of contrastive losses for the document ranking task that can exploit the augmented data to learn an effective ranking model. Our extensive experiments on subsets of the MS MARCO and TREC-DL test sets show that data augmentation, along with the ranking-adapted contrastive losses, results in performance improvements under most dataset sizes. Apart from sample efficiency, we conclusively show that data augmentation results in robust models when transferred to out-of-domain benchmarks. Our performance improvements in in-domain and more prominently in out-of-domain benchmarks show that augmentation regularizes the ranking model and improves its robustness and generalization capability.

CCS Concepts: • **Information systems → Retrieval models and ranking;**

Additional Key Words and Phrases: Information retrieval, IR, ranking, document ranking, contrastive loss, data augmentation, interpolation, ranking performance

J. Leonhardt, K. Rudra, and A. Anand research was primarily conducted while affiliated to L3S Research Center.
This work is supported by the European Union - Horizon 2020 Program under the scheme "INFRAIA-01-2018-2019 "Integrating Activities for Advanced Communities" Grant Agreement n.871042, "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (http://www.sobigdata.eu).
This work is supported in part by the Science and Engineering Research Board, Department of Science and Technology, Government of India, under Project SRG/2022/001548. Koustav Rudra is a recipient of the DST-INSPIRE Faculty Fellowship [DST/INSPIRE/04/2021/003055] in the year 2021 under Engineering Sciences.
Authors' addresses: A. Anand, L3S Research Center, Hannover, Germany; e-mail: aanand@L3S.de; J. Leonhardt, Delft University of Technology, Delft, The Netherlands and L3S Research Center, Hannover, Germany; e-mail: L.J.Leonhardt@tudelft.nl, leonhardt@L3S.de; J. Singh, Independent Researcher, Berlin, Germany; e-mail: jaz7290@gmail.com; K. Rudra, Centre of Excellence in Artificial Intelligence, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India; e-mail: krudra@cai.iitkgp.ac.in; A. Anand, Delft University of Technology, Delft, Netherlands; e-mail: avishek.anand@tudelft.nl.

1046-8188/2024/04-ART119 $15.00
https://doi.org/10.1145/3634911

ACM Reference format:
Abhijit Anand, Jurek Leonhardt, Jaspreet Singh, Koustav Rudra, and Avishek Anand. 2024. Data Augmentation for Sample Efficient and Robust Document Ranking. *ACM Trans. Inf. Syst.* 42, 5, Article 119 (April 2024), 29 pages.
https://doi.org/10.1145/3634911

1 INTRODUCTION

Recent approaches for ranking documents have focused heavily on contextual transformer-based models for both retrieval [29, 39] and re-ranking [11, 26, 35, 45, 77]. To further improve the effectiveness of contextual ranking models, earlier works have explored negative sampling techniques [72], pre-training approaches [6], and different architectural variants [26, 29]. One largely under-explored area is the use of *data augmentation* in neural **information retrieval** (**IR**) to learn effective and robust ranking models.

Data augmentation helps improve the generalization and robustness of highly-parameterized models by creating new training examples through transformations applied to the original data. A key benefit of data augmentation is that it can improve *sample efficiency*, meaning that a model can achieve improved performance with limited amounts of training data. This is because data augmentation effectively increases the size of the training dataset, allowing a model to learn from a wider range of examples. Additionally, data augmentation methods result in robust models allowing for better zero-shot transfer [53, 57]. Augmentation techniques have been successfully used to help train more robust models, particularly when using smaller datasets in computer vision [60], speech recognition [48], spoken language understanding [52], and dialog system [84]. However, the use of data augmentation for document ranking has not been investigated in detail to the best of our knowledge until recently [4, 12].

Contextual models are first *pre-trained* on large amounts of language data followed by task-specific *fine-tuning*. However, popular contextualized models are over-parameterized with more than *100 million* parameters and might over-fit the training data when the task-specific fine-tuning data is small. Many real-world ranking tasks can have smaller query workloads and therefore necessitate sample efficient training like data augmentation [3, 5, 47, 62]. However, simply augmenting training data with existing *point-* or *pairwise ranking* losses does not lead to performance improvements. We show that our data augmentation techniques using existing pointwise ranking losses, i.e., cross-entropy losses, result in *degradation of performance* (cf. Figure 3). This can be attributed to a known lack of robustness to noisy labels [82] and the possibility of poor margins [41], leading to reduced generalization performance.

1.1 Contrastive Learning for Rankings with Data Augmentation

Toward improving the ranking performance in *limited data setting* we first propose both unsupervised and supervised data augmentation methods (Figure 1). Both cases involve creating new query-document pairs from existing instances. We do not perturb the query, only the document. Unsupervised data augmentation methods include adding new query document pairs where documents are relevant extractive pieces of text from an existing relevant document determined by lexical (BM25-based) or semantic (embedding-based) similarity. For supervised augmentation, we use rationale-selection [33] approach specifically devised for document ranking. This approach selects relevant portions of an existing relevant document in a supervised manner given a query.

Secondly, we propose *contrastive learning objectives* for document ranking that can exploit the newly augmented training instances (Figure 1). A key idea in contrastive learning is to learn the input representation of an instance or *anchor* such that its positive instances are embedded

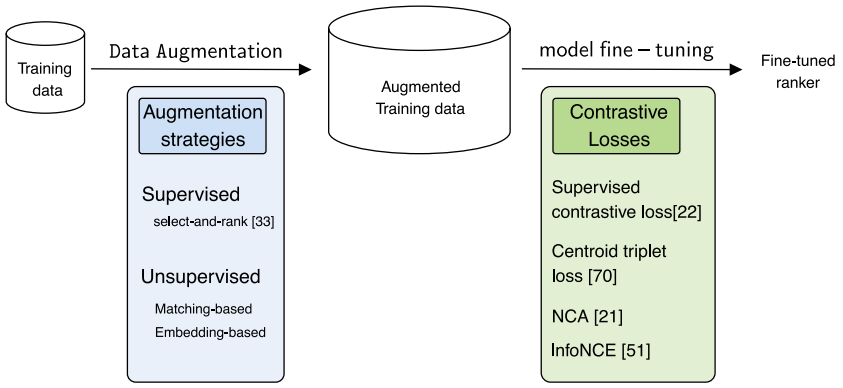

Fig. 1. Training a ranking model with augmented data using different contrastive loss objectives.

closer to each other, and the negative samples are farther apart. In this work, we construct **augmented query-document** pairs from existing positive instances by multiple augmentation strategies. We extend the idea of contrastive learning to the document ranking task by considering query-document pairs belonging to the same query as positive instances, unlike in vision and NLP tasks, where all instances with the same class label can potentially become positive pairs.

Our key contribution to this work is the effective combination of data augmentation and contrastive learning to improve sample efficiency and robustness.

1.2 Results and Key Takeaways

To this end, we systematically explore existing contrastive learning objectives and augmentation strategies on a host of contextual language models – BERT, RoBERTa and DistilBERT in multiple low-data ranking settings—from 100 to 100, 000 training instances. We do not intend to engineer a state-of-art ranking model for document ranking but instead focus on optimization strategies that work well in low-data settings. We find that using the right combination of augmentation technique and loss objective even when you have only 1k training instances leads to a 83% improvement in ndcg@10 for DistilBERT. We find that even larger models like BERT which tend to be more sample efficient in comparison see a 9% improvement in low data settings. When transferring augmented models to out-of-domain datasets, we once again see drastic improvements—RoBERTa sees sample efficiency gains ranging from 18.8% to sometimes a high of 134% on various BEIR datasets with no additional fine-tuning.

1.3 Contributions

In sum, we make the following contributions in this work:

- We propose and study different data augmentation and contrastive loss approaches for document ranking task.
- We also show the impact of model size on ranking performance using augmented data of different sizes.
- We show the performance of different data augmentation and ranking losses in in-domain and out-of-domain (BEIR) settings.

2 RELATED WORK

The related work on the topic at hand can be broadly categorized into three main areas of research, document ranking using contextual models, different data augmentation techniques on text-based

tasks, and finally, we investigate different loss functions used in text ranking and analyse their relationship with contrastive loss.

2.1 Contextual Models for Ad-hoc Document Retrieval

The task of text ranking often involves two steps: a fast retrieval step followed by a more involved re-ranking step. The re-ranking step is particularly important because it can significantly improve the performance of the text ranking task. In this article, the focus is on improving the performance of the re-ranking stage, which typically involves the use of contextual models.

Contextual models, such as BERT [13] and RoBERTa [42], have shown promising improvements in the document ranking task. The input, i.e., query document pairs, can be encoded using two major paradigms for training a contextual re-ranker: joint encoding and independent encoding. In the joint encoding paradigm, the most common way to apply contextual models for document re-ranking is to jointly encode the query and document using an over-parameterized language model [45, 49]. On the other hand, the second paradigm encodes the document and the query independently of each other. Models that implement independent query and document encoding are referred to as dual encoders, bi-encoders, or two-tower models.

While dual encoders are typically used in the retrieval phase, recent proposals have used them in the re-ranking phase as well [34]. It is important to note that a common problem in both approaches is the upper bound on the acceptable input length of contextual models, which restricts their applicability to shorter documents. When longer documents do not fit into the model, they are chunked into passages or sentences to fit within the token limit, either by using transformer-kernels [25, 26], truncation [11], or careful pre-selection of relevant text [33, 58]. As pre-training is an important part of these models, several approaches and pre-training objectives have been proposed specifically for ranking [6, 17, 18, 32].

Ranking using LLM's: Recently a lot of work has been done on re-ranking using LLM's in a zero-shot setup. There have been works using list-wise approach [44, 65], Pairwise Ranking Prompting [54] for re-ranking passages. All the above approaches use either Flan-UL2 model with 20B parameters or ChatGPT (GPT-3.5-turbo, GPT-4) with more than 175B parameters. To best of our knowledge there have been no works on document ranking using LLM's in a zero-shot or instruct tuned setup.

In this work, the focus is on joint encoding models for document ranking, and simple document truncation is employed whenever longer documents exceed the overall input upper bound. By doing so, the aim is to improve the performance of the re-ranking stage and the text ranking task overall.

2.2 Data Augmentation

Data augmentation is a powerful technique that has a significant impact on different segments, including text, speech, image, vision, and more. Researchers have proposed new data augmentation strategies and explored their influence on deep learning models in different fields, such as speech recognition, spoken language understanding, dialog systems, image recognition, and text classification. Some of the proposed techniques include GridMask [7], AutoAugment [10], and [19, 43, 56, 78, 83].

For text-related tasks, various data augmentation techniques have been proposed, such as named entity recognition, sentiment analysis, text classification, and text generation. Data augmentation has been shown to help boost the performance of several downstream **natural language processing** (**NLP**) and text-related tasks. For instance, data augmentation using pre-trained transformer models has been shown to improve the performance of named entity recognition, language inference, text categorization, classification, and query-based multi-document summarization

tasks [31, 61, 64]. A proposed framework called Text Attack [46] combines data augmentation, adversarial attacks, and training in NLP. One of the most common data augmentation techniques in NLP is the use of word embeddings, which involves converting words into numerical vectors. Different techniques can be used to augment word embeddings, such as word substitution, word deletion, and word insertion. Another proposed approach for text classification involves replacing words with synonyms [80] and inserting or deleting words in the input text.

In recent times, data augmentation methods have gained popularity in the context of retrieval tasks. Such techniques have shown promising results for question retrieval [50], query translation [76], question-answering [74, 75], cross-language sentence selection [8], machine reading [68], and query expansion [38]. For instance, Yang et al. [73] proposed a cross-momentum contrastive learning [23] based scheme for open-domain question answering. Dense retriever models [28, 72] have recently been proposed, which sample negative documents to train the dense retrievers in a contrastive way. However, these methods do not pay attention to the uniform nature of contrastive learning [69]. In contrast, a contrastive dual learning based method [37] for dense retrieval takes care of uniformity. Most of these approaches concentrate on negative samples and aim to train an effective dense retriever framework. Works like InPars [4] and PromptAgator [12] focus on using large generative models to generate queries from sampled documents to increase training data for dense retrieval tasks.

2.3 Contrastive Learning

Contrastive losses with data augmentation have been widely studied in unsupervised learning settings, with augmentation of instances treated as positive samples and other random instances serving as negative samples. However, recent research has explored ways to incorporate label information for more precise supervision signals during data augmentation [30]. Various methods have used this approach to learn representations from unsupervised data [24, 51, 63, 71], achieving superior performance compared to other approaches [14, 20]. By generating training instances from original ones using different data augmentation strategies, a contrastive loss can help bring the representation of related entities closer together in the embedding space. For a comprehensive overview, a recent survey on supervised and self-**supervised contrastive learning** (**SCL**) is recommended [27]. Recent research has applied SCL to fine-tuning pre-trained language models, but with limited success [22]. There are various other contrastive losses used in text retrieval, which include InfoNCE loss[51], N-Pair loss [63], centroid triplet loss [70], and lifted structured loss [63]. We discuss in detail the contrastive losses in Section 5.

Another line of work that frequently utilizes contrastive loss functions is *dense passage retrieval*. Karpukhin et al. [28] introduced the DPR model, which uses dual-encoders to independently compute representations of queries and documents. The loss used to train DPR models is similar to the ones mentioned above; it makes use of *in-batch negatives*, i.e., it takes the documents from *all* instances in a given training batch into account and *contrasts* them with the relevant document. Several other approaches [55, 85] have adopted this training objective. Subsequent works have focused on providing better *negative sampling* techniques to replace the in-batch negatives [40, 72, 79]. In principle, our data augmentation techniques can be applied in the context of dual-encoder models for retrieval, although we focus on cross-encoders for re-ranking in this work.

2.4 Extension from Previous Work

This work is an extension of our previous work [2] titled ‘SCL Approach for Contextual Ranking”. In Reference [2], we predominantly explored simple data augmentation techniques with SCL for document ranking. In this work, we considerably increase the scope of our investigations to include more involved supervised data augmentation schemes like [33]. Also included in this article is a

detailed investigation of other metric losses like centroid triplet loss, noise contrastive losses, and neighborhood-based unsupervised losses. Different from Reference [2], we also study how model size impacts model performance when trained on different sizes of augmented and non-augmented data. Finally, we showcase the benefits of data augmentation in zero-shot transfer settings to test the robustness and generalization of the rankers learned using augmented training data.

3 DOCUMENT (RE-)RANKING USING CONTEXTUAL LANGUAGE MODELS

Our task is to train a model for *document re-ranking*. Ranking models usually provide a relevance score when given a query-document pair (q, d) as input. This score can then be used to rank documents based on their relevance to the given query.

Formally, the training set comprises pairs $q_i, d_i{}_{i=1}^{N}$, where q_i is a query and d_i is a document that is either relevant or irrelevant based on its label y_i. The aim is to train a ranker R that predicts a relevance score $\hat{y} \in [0; 1]$ given a query q and a document d:

$$R : (q, d) \mapsto \hat{y}$$

After training, the ranking model R can be used to re-rank a set of documents obtained in the initial retrieval process by a lightweight, typically term-frequency-based, retriever with respect to a query. This is a common practice for ranking tasks, where the documents are initially retrieved and then re-ranked by a more sophisticated and computationally expensive model. Recent studies have shown that pre-trained contextual language models have exhibited promising performance in document ranking tasks [11, 49, 58, 77]. These cross-attention models jointly model queries and documents. In this study, three different joint modeling approaches based on BERT[13], RoBERTa[42] and DistilBERT [59] are considered, and their performance is evaluated under different contrastive loss setup with different amounts of data augmentation. All three models share the same input format: a pair of query q and document d is fed into the model as

$$\texttt{[CLS]}\ q\ \texttt{[SEP]}\ d\ \texttt{[SEP]}$$

To account for the input length limitations of the models, long documents may need to be truncated to fit the sequence length the model is pre-trained with.

Traditionally, there are two main methods to train ranking models, which are pointwise and pairwise. Let us assume of a mini batch of N training examples $\{x_i, y_i\}_{i=1,\ldots,N}$. The pointwise training method considers the document ranking task as a binary classification problem, where each training instance $x_i = (q_i, d_i)$ is a query-document pair and $y_i \in 0, 1$ is a relevance label. The predicted score of x_i is denoted as $\hat{y}i$. The cross-entropy loss function is defined as follows:

$$\mathcal{L}_{\texttt{Point}} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$

In the pairwise training method, each training example contains a query and two documents, $x_i = (q_i, d^+i, d^-i)$, where the former is more relevant to the query than the latter. The pairwise loss function is defined as follows:

$$\mathcal{L}_{\texttt{Pair}} = \frac{1}{N}\sum_{i=1}^{N}\max\{0, m - \hat{y}_i^+ + \hat{y}_i^-\}$$

where $\hat{y}_i^+$ and $\hat{y}_i^-$ are the predicted scores of d_i^+ and d_i^-, respectively, and m is the loss margin. The pairwise method is commonly used for ranking tasks as it takes into account the relative ordering between documents.

ALGORITHM 1: Training Data Augmentation

Input: training batch B, number of sentences per augmented document k_a
Output: augmented training batch B'

1 $B' \leftarrow$ empty list
2 **foreach** (q, d^+, d^-) **in** B **do**
    // keep the original example
3   append (q, d^+, d^-) to B'
    // create augmented example
4   $d_a^+ \leftarrow \texttt{augment}(d^+, q, k_a)$
5   $d_a^- \leftarrow$ random irrelevant document
6   append (q, d_a^+, d_a^-) to B'
7 **end**
8 **return** B'

4 DATA AUGMENTATION FOR DOCUMENT RANKING

We intend to use data augmentation in the context of a document ranking task to improve the quality of the training data and increase the diversity of the examples presented to the model. In this section, we propose extractive methods to create new query-document pairs from the instances already in the training dataset.

To enhance the training data, we utilize augmentation techniques to form d_a^+ from d^+ for each triple (q, d^+, d^-) in the training set. Creating an augmented instance is extractive since it involves *selecting* relevant sentences to the corresponding query, followed by random sampling of an irrelevant document d_a^-. The resulting augmented training instances are then appended to their respective batch, effectively doubling the size of each batch.

The document is treated as a sequence of sentences s_i, denoted as $d = (s_1, s_2, \ldots, s_{|d|})$. A query-specific selector is employed to choose a fixed number of sentences from the document, based on the distribution $p(s \mid q, d)$, encoding the relevance of the sentence given the input query q. This distribution is used to select an extractive, query-dependent summary, denoted as $d' \subseteq d$. The augmentation process is detailed in Algorithm 1. The function that creates the augmented document (line 4) is defined as

$$\texttt{augment}(d, q, k) = k\text{-argmax}_{1 \le i \le |d|} \, \text{score}(q, s_i). \tag{1}$$

The sentences in the augmented documents are ordered by score, i.e., the original order is not preserved. Note that the scoring function $\text{score}(q, s)$ in Equation (1) represents an augmentation strategy. We present both unsupervised (heuristic) and supervised (predictive) augmentation strategies in the following sections.

4.1 Unsupervised Augmentation

4.1.1 Term-Matching-Based (BM25). BM25 (Best Matching 25) is a variant of the popular tf-idf weighting scheme that is commonly used to rank documents by relevance to a query. BM25 works by computing a score for each document in a corpus based on its relevance to a query. The score is calculated by combining several factors, including the frequency of the query terms in the document, the inverse document frequency of the query terms, and the length of the document, and is then normalized by the average document length in the corpus. We use tf-idf scores between the query q and sentences s_i to determine the best sentences:

$$\text{score}_{\text{BM25}}(q, s_i) = \text{BM25}(q, s_i).$$

Inverse document frequencies are computed over the complete corpus.

4.1.2 Semantic-Similarity-Based (GloVe). GloVe works by constructing a co-occurrence matrix of word pairs based on their co-occurrence frequency in a corpus. The co-occurrence matrix is then factorized using matrix factorization techniques, such as **singular value decomposition (SVD)**, to generate low-dimensional embeddings for each word. Unlike other word embedding methods, such as Word2Vec, GloVe is designed to capture both the global co-occurrence statistics and the local context of the words. It does this by weighting the importance of word co-occurrences based on their frequency and using a logarithmic function to down weight the importance of highly frequent co-occurrences. We use GloVe to find the representation for query q and sentence s_i. Both the query and sentence are represented as average over their constituent word embeddings. We use semantic (cosine) similarity scores between the query q and sentences s_i to determine the best sentences for a given query, i.e.,

$$\text{score}_{\text{GloVe}}(q, s_i) = \langle \text{E}_{\text{GloVe}}(q), \text{E}_{\text{GloVe}}(s_i) \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the dot product and $\text{E}_{\text{GloVe}}(\cdot)$ computes the average of the embeddings of all tokens in a sequence.

4.2 Supervised Augmentation

Unlike unsupervised augmentation where sentence selections were based on lexical- and semantic-similarities, we also consider sentence selections based on supervised training data. Recently, in the area of explainable IR, select-and-rank approaches have been proposed that given a query-document pair, select an extractive piece of text from the document as a potentially relevant signal [33, 36, 81]. Specifically, in the selection phase, relevant sentences given a query is extracted. This is followed by the ranking phase, where the relevance estimation is performed just on the extracted sentences. The key idea here is that typically a small part of the document is relevant and the selection phase filters out non-relevant text. The supervision signal is obtained by the training data and a combination of the gumbel-max trick and reservoir sampling is used to train the selector network [33]. The output of the selection phase can be considered as a query-based extractive summary and can be utilized for our data augmentation needs. We considered the two supervised sentence selection approaches—`Linear` sentence selection, and `Attention` based sentence selection.

Let a query $q = (t_1^q, \ldots, t_{|q|}^q)$ and document $d = (t_1^d, \ldots, t_{|d|}^d)$ be sequences of (embedded) tokens, respectively. Furthermore, let s_{ij} be a sentence within the document, such that $s_{ij} = (t_i^d, \ldots, t_j^d)$. In the following, we describe how the two selection approaches score a sentence s_{ij} w.r.t. a query q.

4.2.1 Linear Sentence Selection. The linear sentence selector is a simple non-contextual model, i.e., each sentence within the document is scored independently. This approach is similar to the GloVe-based augmentation, however, this model has been trained specifically on a ranking task. The query and each sentence are first represented as the average of their token embeddings, respectively. After averaging, each representation is passed through a single-layer feed-forward network. The final score of a sentence is then computed as the dot product of its representation and the query representation. Formally, a sequence of tokens, $t = (t_1, \ldots, t_{|t|})$, is represented as

$$\text{Enc}(t) = \frac{\sum_{s_i \in t} (W t_i + b)}{|t|},$$

where W and b are trainable parameters of the feed-forward layer. The score of a sentence s_{ij} w.r.t. the query q is then computed as

$$\text{score}_{\text{Lin}}(q, s_{ij}) = \langle \text{Enc}(q), \text{Enc}(s_{ij}) \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the dot product.

4.2.2 Attention-Based Sentence Selection. The Attention-based selector computes sentence-level representations based on the QA-LSTM model [66]. Query and document are first contextualized by passing their token embeddings through a shared bi-directional LSTM:

$$q^{\text{LSTM}} = \text{Bi-LSTM}(q),$$
$$d^{\text{LSTM}} = \text{Bi-LSTM}(d).$$

The query representation $\hat{q}$ is obtained by applying element-wise max-pooling over q^{LSTM}:

$$\hat{q} = \text{Max-Pool}(q^{\text{LSTM}})$$

For each hidden representation d_i^{LSTM}, attention to the query is computed as

$$m_i = W_1 h_1 + W_2 \hat{q},$$
$$h_i = d_i^{\text{LSTM}} \exp\left(W_3 \tanh\left(m_i\right)\right),$$

where W_1, W_2 and W_3 are trainable parameters. For s_{ij}, let h_{ij} denote the corresponding attention outputs. The sentence representation is computed similarly to the query representation, i.e.,

$$\hat{s}_{ij} = \text{Max-Pool}(h_{ij}).$$

The final score of a sentence is computed as the cosine similarity of its representation and the query representation:

$$\text{score}_{\text{Att}}(q, s_{ij}) = \cos(\hat{q}, \hat{s}_{ij}).$$

4.3 Creating Augmented Training Batches

To preserve the randomness in the data while augmenting the training set, the creation of mini-batches is a crucial step in our approach. It has been demonstrated in prior studies that the quality of the augmented data plays an important role in the performance of self-SCL [22].

Our approach begins with retrieving the top-k documents per query using a first stage retrieval method. From this top-k set, we create the training dataset by collecting all positive query-document training instances. For each positive pair, we randomly sample one irrelevant document to serve as the negative instance. To ensure randomness, we shuffle the resulting set of (q, d^+, d^-) triples. It should be noted that, for pointwise training, we create two query-document pairs from each triple.

5 CONTRASTIVE LEARNING WITH AUGMENTED DATA FOR DOCUMENT RANKING

Augmented training data can be used to train a model with no alteration to the pointwise and pairwise cross entropy loss objective. We instead propose the usage of contrastive learning to better leverage augmented data. Augmented query document pairs are not clean training samples by definition and should be treated accordingly during training.

In this section, we detail different contrastive learning objectives which we used to train our ranking models. We do not propose a new contrastive loss objective but instead focus on combining existing losses correctly for ranking tasks.

5.1 Ranking-Based Supervised Contrastive Loss

Theaim of the SCL loss is to capture the similarities between relevant document parts for a given query and contrast them against examples from non-relevant queries. The ranking model outputs the query-document representation $\Phi(\cdot) \in \mathbb{R}^t$ (e.g., the `[CLS]` output for BERT-based models) which can be used to compute similarities. The SCL loss function includes the adjustable scalar temperature parameter $\tau > 0$ that controls the distance between relevant and non-relevant examples, and the scalar weighting hyper-parameter λ, which is tuned for each downstream task and setting.

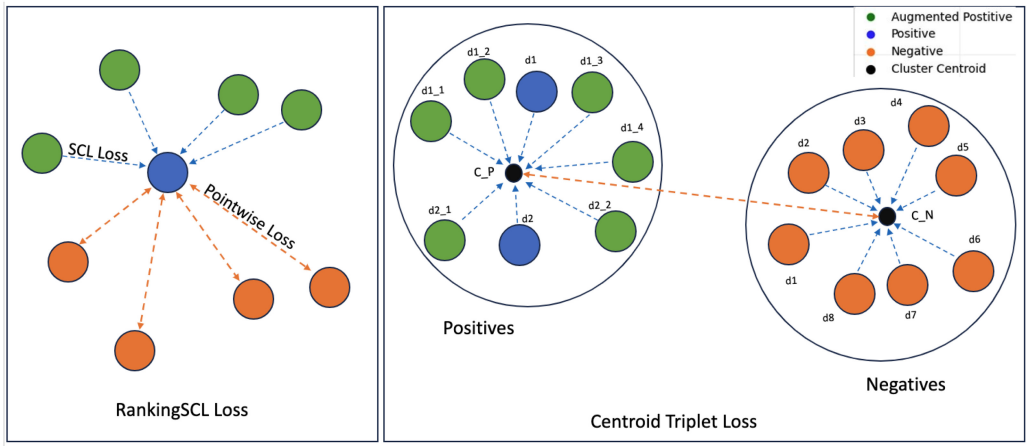

Fig. 2. On the left, we describe RankingSCL loss (Equation (3)) and on the right Centroid triplet Loss (Equation (4)).

The SCL loss is formulated as

$$\mathcal{L}_{\text{SCL}} = \sum_{i=1}^{N} -\frac{1}{N_+} \sum_{j=1}^{N_+} \mathbf{1}_{\substack{q_i=q_j,\\ i\neq j,\\ y_i=y_j=1}} \log \frac{\exp(\Phi(x_i)\cdot\Phi(x_j)/\tau)}{\sum_{k=1}^{N} \mathbf{1}_{i\neq k} \exp(\Phi(x_i)\cdot\Phi(x_k)/\tau)} \tag{2}$$

where N_+ is the total number of positive examples (i.e., relevant query-document pairs) in the batch.

The loss function enforces that the positive pair with the same query should be embedded close to each other, rather than a pair of documents that are relevant for different queries. This is important since it ensures that the representations for the "relevant parts" of the same query are close to each other. The final ranking SCL loss is

$$\mathcal{L}_{\texttt{RankingSCL}} = (1-\lambda)\mathcal{L}_{\texttt{Ranking}} + \lambda\mathcal{L}_{\text{SCL}} \tag{3}$$

We illustrate the RankingSCL loss in Figure 2 (left figure) using a pointwise ranking loss. It shows the two components working together; the ranking loss separates the pairs of positive and negative documents, while the contrastive loss moves all positive documents in the batch closer to each other.

5.2 Ranking-Based Centroid Triplet Loss

The triplet loss function [70] enforces the learning of a distance metric that satisfies the following property: for a given triplet of data points (A, P, N), where A is an anchor point (same class), P is a positive example for A, and N is a negative example for A, the distance between A and P should be smaller than the distance between A and N by a certain margin. This property is known as the triplet constraint. The objective is to minimize the distance between $A - P$, while maximizing the distance to the N sample.

The loss function is formulated as follows:

$$\mathcal{L}_{\text{triplet}} = \left[\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha\right]_+$$

where $[z]_+ = \max(z, 0)$, f denotes embedding function learned during training stage and α is a margin parameter.

In the case of CTL, instead of comparing the distance of an anchor A to the positive and negative instances, CTL measures the distance between A and class centroids c_P and c_N representing either the same class as the anchor or a different class, respectively. In case of a document ranking task, for a given query q, let $\{d^+{}_i, d^-{}_i\}_{i=1,\ldots,N}$ be a set of relevant and non-relevant documents, respectively, to that given query. Where $d^+{}_i$ can be positives or augmented positives. Let c_P be the centroid of the relevant class (relevant and augmented relevant documents) and c_N be centroid of the non-relevant class. Then, CTL is, therefore, formulated as

$$\mathcal{L}_{\text{CTriplet}} = \left[\|d_i^+ - c_P\|_2^2 - \|d_i^- - c_N\|_2^2 + \alpha_c\right]_+ \tag{4}$$

$$\mathcal{L}_{\texttt{RankingCTriplet}} = (1-\lambda)\mathcal{L}_{\texttt{Ranking}} + \lambda\mathcal{L}_{\texttt{triplet}}. \tag{5}$$

5.3 Ranking-Based InfoNCE

The InfoNCE loss function [51] encourages similar samples to have similar representations and dissimilar samples to have dissimilar representations. This is achieved by comparing the similarity between positive pairs (similar samples) and negative pairs (dissimilar samples) using the cross-entropy loss. The InfoNCE loss function is a variant of the standard cross-entropy loss that has been modified to account for the varying number of negative samples used in the contrastive learning framework.

InfoNCE is particularly effective in situations where the number of negative samples is large. Additionally, the use of the cross-entropy loss makes the InfoNCE loss easy to optimize using standard optimization algorithms.

In document ranking given a query $\mathbf{q_i}$ and corresponding relevant document $\mathbf{d_i^+}$, the positive sample should be drawn from the conditional distribution $p(\mathbf{x} \mid \mathbf{d_i^+})$, while $N-1$ negative samples are drawn from the proposal distribution $p(\mathbf{x})$, independent from the context d_i^+ . For simplicity, let us label all the documents for the query q_i as $D = \{\mathbf{d}_i\}_{i=1}^N$ among which only one of them $\mathbf{d}_{\text{pos}}$ is a positive sample. The probability of we detecting the positive sample correctly is

$$p(C = \text{pos} \mid D, \mathbf{d_i^+}) = \frac{p(d_{\text{pos}} \mid \mathbf{d_i^+}) \prod_{i=1,\ldots,N; i\neq \text{pos}} p(\mathbf{x}_i)}{\sum_{j=1}^N [p(\mathbf{x}_j \mid \mathbf{d_i^+}) \prod_{i=1,\ldots,N; i\neq j} p(\mathbf{x}_i)]} = \frac{\frac{p(\mathbf{x}_{\text{pos}} \mid d_i^+)}{p(\mathbf{x}_{\text{pos}})}}{\sum_{j=1}^N \frac{p(\mathbf{x}_j \mid \mathbf{d_i^+})}{p(\mathbf{x}_j)}} = \frac{f(\mathbf{x}_{\text{pos}}, \mathbf{d_i^+})}{\sum_{j=1}^N f(\mathbf{x}_j, \mathbf{d_i^+})}$$

where the scoring function is $f(\mathbf{x}, \mathbf{d_i^+}) \propto \frac{p(\mathbf{x} \mid \mathbf{d_i^+})}{p(\mathbf{x})}$. The InfoNCE loss optimizes the negative log probability of classifying the positive sample correctly:

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\left[\log \frac{f(\mathbf{x}, \mathbf{d_i^+})}{\sum_{\mathbf{x}' \in X} f\left(\mathbf{x}', \mathbf{d_i^+}\right)}\right] \tag{6}$$

The fact that $f(x, d_i^+)$ estimates the density ratio $\frac{p(x \mid d_i^+)}{p(x)}$ has a connection with mutual information optimization. To maximize the the mutual information between input x and context vector d_i^+, we have:

$$I(\mathbf{x}; \mathbf{d_i^+}) = \sum_{\mathbf{x}, \mathbf{d_i^+}} p(\mathbf{x}, \mathbf{d_i^+}) \log \frac{p(\mathbf{x}, \mathbf{d_i^+})}{p(\mathbf{x}) p(\mathbf{d_i^+})} = \sum_{\mathbf{x}, \mathbf{d_i^+}} p(\mathbf{x}, \mathbf{d_i^+}) \log \frac{p(\mathbf{x} \mid \mathbf{d_i^+})}{p(\mathbf{x})}$$

where the logarithmic term in below is estimated by f.

$$\frac{p(\mathbf{x} \mid \mathbf{d_i^+})}{p(\mathbf{x})}$$

$$\mathcal{L}_{\texttt{RankingInfoNCE}} = (1-\lambda)\mathcal{L}_{\texttt{Ranking}} + \lambda\mathcal{L}_{\texttt{InfoNCE}}. \tag{7}$$

5.4 Neighbourhood Component Analysis (NCA)

NCA [21] is a distance-based classification algorithm that learns a metric for measuring the similarity between data points. The metric is learned based on a training set of labeled examples and is used to classify new, unseen examples. The basic idea behind NCA is to learn a linear transformation of the input data that maximizes the accuracy of a **k-Nearest Neighbor** (**k-NN**) classifier. The transformation is learned by minimizing a loss function that measures the classification error of the k-NN classifier. In context of ranking, given a query q_i and corresponding relevant documents d_i and d_j, the NCA loss function is defined as follows:

$$\mathcal{L}_{\texttt{NCA}} = \sum_i \log\left(\sum_{j \in C_i} p_{ij}\right) = \sum_i \log\left(p_i\right), \tag{8}$$

where p_i is the probability of calculating the document d_i to the relevant class as neighboring point d_j is defined as

$$p_i = \sum_{j \in C_i} p_{ij}$$

We define the p_{ij} using a softmax over Euclidean distances in the transformed space:

$$p_{ij} = \frac{\exp\left(-\|Ad_i - Ad_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|Ad_i - Ad_k\|^2\right)}$$

where A is the transformation matrix, d_i is relevant to the query q_i and d_k is the k-Nearest relevant Neighbor of d_i in the input space.

$$\mathcal{L}_{\texttt{RankingNCA}} = (1-\lambda)\mathcal{L}_{\texttt{Ranking}} + \lambda\mathcal{L}_{\texttt{NCA}}. \tag{9}$$

5.5 Combining Ranking Loss with Contrastive Loss

We propose a simple linear combination of standard ranking losses with contrastive losses described above for the augmented training samples. The overall ranking losses are then given by

$$\mathcal{L}_{\texttt{ContrastiveRanking}} = (1-\lambda)\mathcal{L}_{\texttt{Ranking}} + \lambda\mathcal{L}_{\texttt{Contrastive}},$$

where

$$\mathcal{L}_{\texttt{Ranking}} \in \{\mathcal{L}_{\texttt{Point}}, \mathcal{L}_{\texttt{Pair}}\} \text{ and}$$
$$\mathcal{L}_{\texttt{Contrastive}} \in \{\mathcal{L}_{\texttt{SCL}}, \mathcal{L}_{\texttt{triplet}}, \mathcal{L}_{\texttt{InfoNCE}}, \mathcal{L}_{\texttt{NCA}}\}.$$

We use the following terminology in the article: linear interpolation of `Pointwise` and SCL is referred to as **`RankingSCL`**. Although all of the aforementioned losses are contrastive, each of them has subtle differences in the way they optimize learning of the representation space.

— *Supervised Contrastive Loss* (Equation 2) encourages the features of positive examples from the same class to be similar while making sure that the features of negative examples from different classes are dissimilar.
— *Centroid Triplet Loss* (Equation 4) is a contrastive loss function that is designed to encourage a larger *margin* between the distances of the positive and negative examples compared to the distance between the anchor and the centroid of the positive examples. This means that the loss function places a greater emphasis on making sure that the positive examples are tightly clustered around their centroid, while the negative examples are kept farther away.

— *Neighborhood Component Analysis (NCA)* (Equation 8) is a metric learning algorithm that is designed to learn a linear transformation that maximizes the accuracy of the KNN classifier. The goal of NCA is to find a transformation that reduces the distance between examples that belong to the same class and increases the distance between examples from different classes.
— *InfoNCE (InfoMax Contrastive Estimation)* (Equation 6) is based on the concept of maximizing mutual information between positive examples and minimizing it between negative examples. It is often used in self-supervised learning tasks where there is no labeled data available.

In summary, the CTL and SCL are used for supervised learning tasks where labeled data is available, while NCA and InfoNCE can be used for unsupervised or self-supervised learning tasks where labeled data is not available.

6 EXPERIMENTAL SETUP

In this section, we describe the setup we used to answer the research questions. Note that we focus on the re-ranking task and not the retrieval task.

6.1 Datasets

We conduct experiments on in-domain and out-of-domain benchmarks to showcase the utility and performance benefits of data augmentation.

6.1.1 In-Domain Benchmark—TREC-DL. In this study, we utilize the dataset provided by the TREC Deep Learning track in 2019. We evaluate our proposed model on Doc'19, comprising of 200 distinct queries. The training and development set is obtained from MS MARCO, which contains a total of 367K queries. For the retrieval of the top 100 documents for each query, we use Indri [9].

6.1.2 Out of Domain Benchmark—BEIR. The BEIR dataset is a collection of datasets used for benchmarking and evaluating the performance of IR models [67]. The datasets are selected from various domains, such as news articles, scientific papers, and product reviews. It consists of 17 different datasets.

6.2 Ranking Models

We use different cross-attention models for our experiments:

(1) **BERT** [13], a transformer-based large, pre-trained contextual model. Our implementation employs the base version of BERT, which is composed of 12 encoder layers, 12 attention heads, and 768-dimensional output representations. Additionally, the maximum input length is limited to 512 tokens.
(2) **RoBERTa** [42] is another cross-attention model which is architecturally identical to BERT; the only difference between the two models is the pre-training procedure.
(3) **DistilBERT** [59] is a smaller, distilled version of BERT that is designed for faster and more efficient training and inference. It has only 6 transformer layers, 768-dimensional output representations, and a maximum input length of 512 tokens. DistilBERT was trained using a novel technique called knowledge distillation, where a larger pre-trained model is used to teach a smaller model, resulting in significant reductions in model size and computational cost while maintaining a high level of performance.
(4) **BERT-Large** [13] is a large-scale pre-trained language model that uses a bidirectional transformer architecture to learn representations of text. It has 336 million parameters, which is 4 times larger than the original BERT-base model. It has 24 encoder layers, 16 attention heads, and 1,024-dimensional output representations.

We experimented with (a) different types of contextual models - BERT, RoBERTa, DistilBERT, (b) varying dataset sizes—100, 1K, 2K, 10K, 100K instances for Doc'19 (c) ranking losses - `RankingSCL`, `RankingInfoNCE`, `RankingCTriplet`, `RankingNCA` d) different datasets Doc'19, BEIR. This leads to a large experimental space for exploration, which we have synthesized into key insights in our results. For instance, the number of models trained on Doc'19 is around 1,050 with 208 best combinations chosen for reporting. Given a large number of models, it is difficult to report all combination of results and their respective hyperparameters. So, we report the best models in the article and a subset of the results, hyperparameters and can be found in Appendix A.

6.3 Batch Creation and Hyperparameters

In Section 4.3, we described our approach to creating batches for SCL, where we start with positive query-document pairs from the top-k retrieved set and randomly sample negative pairs for the original dataset. We also use a selector to generate augmented versions of documents. For evaluating our approach, we experimented with different sizes of query-document pairs for MS MARCO, including 1k, 2k, 10k, and 100k. For instance, the 1k dataset contains 500 positive and 500 negative pairs in the original dataset, to which we added 1k more pairs through the augmentation process, resulting in a total of 2k query-document pairs. This pattern holds for the other three sizes as well. It is worth noting that we only augment the training data, and the validation and test sets remain unaltered.

Hyperparameters. We have two hyperparameters in our models with `RankingSCL`: the temperature (τ), and the degree of interpolation (λ) as in `RankingSCL` [Equation (3)]. For all other losses ([Equation (5)][Equation (7)][Equation (9)]), we only have one hyperparameter (λ) for interpolation. We use the MS MARCO development set to determine the best combination of hyperparameters. These parameters are different for different ranking models and augmentation strategies. For example, in TREC-DL, BERT ranking model using `Attention` data augmentation and `RankingCTriplet` loss objective returns the best score on the validation set at $\lambda = 0.3$. In all our experiments, we use a batch size of 16. A brief of the hyperparameters used is given in Appendix A.1 (Table 6).

7 EXPERIMENTAL RESULTS

The results section is divided into three major subsections. Section 7.1 presents the results of our in-domain experiments where we strictly observe how augmentation and contrastive learning can benefit sample efficiency and performance for the same dataset. Then, we discuss the out-of-domain performance of the models born out of our training regimen. More specifically, we study the zero-shot transfer of our models to other ranking datasets that vary in topicality and type of query in Section 7.2. Finally, we discuss the failure cases, limitations, and drawbacks of our approach in Section 7.3. The key research questions we seek answers for are as follows:

The key research questions we seek answers for are as follows:

RQ I. Does data augmentation require a different training paradigm?
RQ II. Which data augmentation technique provides the highest sample efficiency gains?
RQ III. Which contrastive loss objective leads to the highest gains when data augmentation type and budget are fixed?
RQ IV. Does our training regimen impact various model sizes differently?
RQ V. Do augmented models transfer better than their non-augmented counterparts?

7.1 In-Domain Experiments

RQ I. Does data augmentation require a different training paradigm?

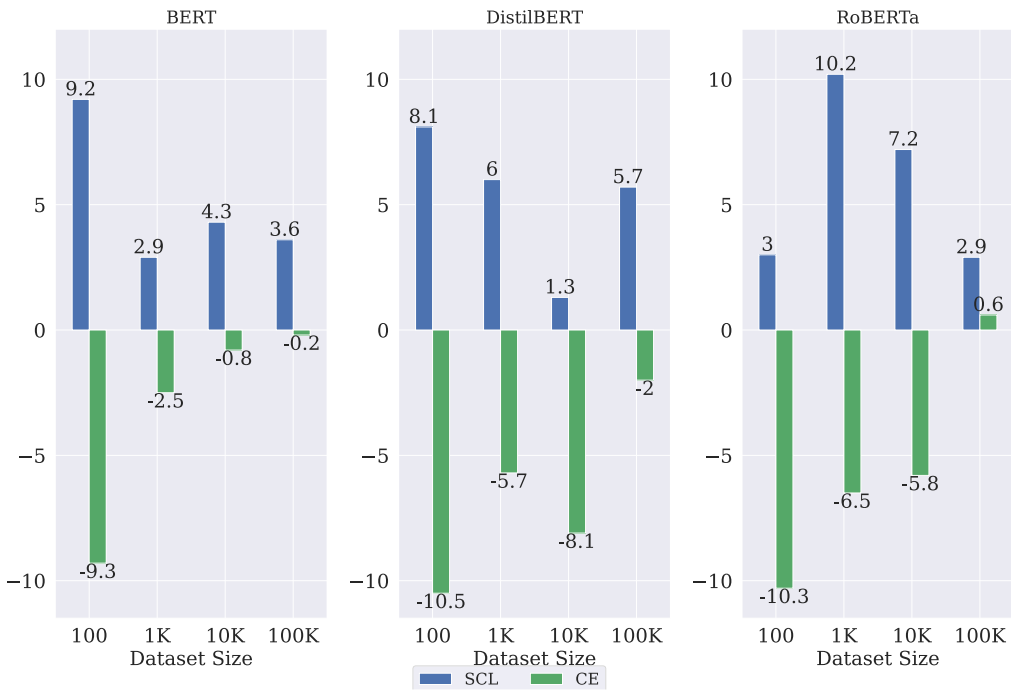

Fig. 3. Relative nDCG@10 improvement of **CE** (cross entropy) model and **SCL** model over **Baseline** model. CE model is trained on an augmented dataset using pointwise loss, SCL is trained on an augmented dataset with `PointwiseRankingSCL` loss and the Baseline model is trained on a non-augmented dataset using pointwise loss. The dataset used here is Doc'19 and BM25 augmentation strategy.

We first verify the impact of our proposed approach by comparing three types of models—a model with data augmentation and SCL versus a model trained with only `Pointwise` loss (baseline) versus a model trained with `Pointwise` loss and augmented data (CE). We also repeated the experiment with a pairwise objective and found the same trends. The SCL model is trained with augmented data using the `PointwiseRankingSCL` loss objective, i.e., pointwise ranking loss interpolated with the SCL objective. In Figure 3, we plot the relative improvement (in terms of nDCG@10) of the CE and `PointwiseRankingSCL` model over the baseline (zero line) (trained without data augmentation) with BM25 augmentation and pointwise objective. We see that only using `Pointwise` loss with augmented data performs worse than baseline. We find that `PointwiseRankingSCL` more effectively utilizes augmented data to learn better representations, which are reflected in consistent improvements over the baseline. As the size of the training set increases, the detrimental effect of data augmentation on the standard CE loss objective diminishes but does not increase sample efficiency. The `PointwiseRankingSCL` model on the other hand sees gains in efficiency for all models. Augmenting a dataset of 100 and 1K examples leads to a 8.1% and 6% improvement over the baseline, respectively, for DistilBERT only when using our `RankingSCL` loss. This establishes that using data augmentation with traditional ranking loss functions is detrimental to ranking performance.

Insight 1. A different training paradigm is required to take advantage of the data augmentation techniques we propose. In particular, adding a contrastive learning objective leads to gains between 1.3% and 10.2% in terms of sample efficiency whereas training without an additional objective and the augmented dataset results in lowered sample efficiency.

RQ II. Which data augmentation technique provides the highest sample efficiency gains?

Table 1. Document Re-Ranking Results on the Doc'19 Datasets for Pointwise with RankingSCL on Supervised(Linear, Attention) vs Un-Supervised (BM25,GloVe) Augmentation

	Supervised Augmentation		**Unsupervised Augmentation**	
Ranking Models	Linear	Attention	BM25	GloVe
BERT				
100	**0.599**(▲29.3%)*	0.569(▲22.7%)*	0.506(▲9.2%)*	0.515(▲11.1%)*
1k	0.554(▲3.7%)	**0.583**(▲9.1%)*	0.541(▲2.9%)#	0.538(▲2.4%)#
2k	0.592(▲5.1%)	**0.597**(▲5.9%)	0.561(▲3.8%)	0.563(▲4%)
10k	0.600(▲1.4%)	0.599(▲1.3%)#	0.617(▲4.3%)	**0.626**(▲5.8%)
100k	**0.628**(▲2.2%)	0.618(▲0.5%)	0.602(▲3.6%)	0.611(▲5.1%)
RoBERTa				
100	0.272(▲15.8%)*	**0.296**(▲25.9%)*	0.242(▲3%)	0.266(▲13.3%)
1k	0.288(▼ − 3%)	0.296(▼ − 0.5%)	**0.325**(▲10.2%)	0.308(▲4.3%)
2k	0.516(▲75.6%)*	**0.529**(▲80%)*	0.328(▲7.2%)	0.356(▲16.3%)
10k	**0.599**(▲7.6%)*	0.593(▲6.4%)*	0.597(▲7.2%)*	0.596(▲7.0%)*
100k	**0.637**(▲10.2%)	0.610(▲5.6%)	0.598(▲2.9%)	0.611(▲5.0%)
DistilBERT				
100	0.247(▲20.7%)*	**0.258**(▲25.8%)#	0.222(▲8.1%)#	0.219(▲7%)
1k	0.287(▲31%)*	**0.372**(▲70%)*	0.254(▲6%)	0.253(▲5.3%)
2k	**0.517**(▲85.7%)*	**0.517**(▲85.7%)*	0.309(▲6.2%)#	0.314(▲7.7%)
10k	0.545(▼ − 3.6%)	0.573(▲1.4%)	0.573(▲1.3%)	**0.583**(▲3.1%)
100k	0.582(▼ − 4%)	0.590(▼ − 2.6%)	**0.641**(▲5.7%)	0.608(▲0.2%)

We show the relative improvement of the augmentation approaches against a baseline without augmentation in parentheses. Statistically significant improvements at a level of 95% and 90% are indicated by * and #, respectively [16]. The best results for each dataset and each model is in **bold** and second is underlined .

To answer RQ II, we extend the previous experiment with more data augmentation techniques. In Table 1, we compare two different classes of augmentation techniques: data augmentation using supervised methods (Linear, Attention) and unsupervised methods (BM25, GloVe). We use nDCG@10 as the key performance indicator. We trained various models using the RankingSCL loss objective on different dataset sizes and different contextual models (BERT, RoBERTa, DistilBERT).

Our first major observation is that augmentation helps across the board. As dataset size increases, all methods show steady improvement which is not surprising but the magnitude of relative improvement over the baseline is large in nearly all cases. With dataset of 100 instances attention based approach performs 20% better than baseline on all models. With 1k samples to train on, attention based augmentation with RankingSCL leads to a near 70% relative improvement for DistilBERT. BERT achieves the highest absolute performance on the 1k and 2k samples using attention based augmentation. The exact choice of augmentation technique is not always clear although supervised techniques tend to generally outperform unsupervised techniques. Supervised techniques benefit from being exposed to a ranking oriented sentence selection task beforehand.

For RoBERTa in particular, we see a different technique winning for each dataset. However, when observing the absolute ndcg@10 measurements, we find that in all cases except for 1k, supervised methods are either comparable or better by a large margin. Unsupervised methods, while simpler, are particularly poor for lean models like DistilBERT. We believe that supervised methods are usually more capable of constructing meaningful augmentations. Unsupervised methods create noisy samples (see Table 2) by either ignoring semantics (BM25—syntactic matching) or not being aware of the ranking task (GloVe).

Table 2. Anecdotal Augmented Training Instances from Doc'19

Method	Augmented Document
Query: 293327 — *how many players are in a cricket team*	**Document**: D863798
BM25	The quota system was introduced as part of South Africa's re-admission into international cricket [...]
GloVe	South Africa have announced a commitment to a minimum of five black players in their squad for the World Cup [...]
`Linear`	Salute Michael Jordan at 40Is this the end for Warne?
`Attention`	A team at any level should be based on the best 11 players available, not on what colour you are.
Query: 428365 — *is there a natural muscle relaxer*	**Document**: D101612
BM25	Which muscle relaxer is best for long term use?. What muscle relaxer [...] and muscle spasms chronically. Any non addictive muscle relaxant? [...]
GloVe	Non habit-forming muscle relaxers Public Forum Discussions Which muscle relaxer is best for long term use? [...]
`Linear`	is a muscle relaxer an opiate?
`Attention`	Is Ativan a good muscle relaxant?
Query: 197720 — *gussie's house of flowers madison ga*	**Document**: D3201531
BM25	The idea was simple - he figured he could sell a few hundred floral bouquets to parents for a local high school graduation ceremony [...]
GloVe	Gussie's House Of Flowers Gussie's House Of Flowers Business name: Gussie's House Of Flowers Address: 136 W Jefferson St City: Madison State: Georgia [...]
`Linear`	Gussie's House Of Flowers "Gussie's House Of Flowers Business name: Gussie's House Of Flowers Address: 136 W Jefferson St City: Madison State: Georgia
`Attention`	The block between St. Clair and Madison, which Cason's house is on, is lined with potted fake flowers.

BM25 and GloVe are unsupervised methods that select paragraphs from the original document denoted here by it's ID. `Linear` and `Attention` are supervised techniques that select relevant sentences based on the query from the same document.

We also observe that the relative improvements over baseline are greater in the case of smaller datasets compared to the larger datasets (which ratifies the findings from Reference [2]) for both classes of approaches. In the case of RoBERTa and DistilBERT we see an 80% improvement (statistically significant) over the baseline for the 2K dataset when using supervised augmentation. For the 100k dataset, surprisingly unsupervised methods are key to achieving sample efficiency for DistilBERT which requires further investigations into the interplay between model size and the ability to learn effectively from augmented data.

The difference between linear and attention is more evident when observing smaller datasets (100, 1k, and 2k). For BERT and DistilBERT we see large sample efficiency gains when using supervised approaches for the 1k dataset but attention results in the largest gains: 3.7% vs 9.1% for BERT and 31% vs 70% for DistilBERT when compared with linear. The same can be observed for 100 instances where `Attention` gains are 22.7% vs 29.3% in case of BERT and 25.8% vs 20.7% in case of DistilBERT in comparison with `Linear`

Table 3. nDCG@10 Performance of Different Language Models (BERT, RoBERTa, and DistilBERT) on Different Loss Functions (RankingSCL, RankingInfoNCE, RankingCTriplet, and RankingNCA) at Different Training Set Sizes (100, 1k, 2k, 10k, and 100k) with **Attention** Data Augmentation

Losses	RankingSCL	RankingInfoNCE	RankingCTriplet	RankingNCA
BERT				
100	0.569(▲22.7%)*	**0.573**(▲23.7%)*	0.521(▲12.3%)*	0.563(▲21.5%)*
1k	**0.583**(▲9.1%)*	0.529(▼ − 1%)	0.576(▲7.9%)*	0.564(▲5.5%)
2k	0.597(▲5.9%)	0.590(▲4.7%)	**0.601**(▲6.6%)	0.580(▲2.9%)
10k	0.599(▲1.3%)#	0.603(▲1.9%)	**0.620**(▲4.8%)	0.590(▼ − 0.4%)
100k	0.618(▲0.5%)	0.612(▼ − 0.5%)	**0.634**(▲3.1%)	0.615(▲0%)
RoBERTa				
100	**0.296**(▲25.9%)	0.269(▲14.7%)	0.275(▲17.1%)*	0.285(▲21.4%)*
1k	0.296(▼ − 0.5%)	0.303(▲1.9%)	**0.317**(▲6.6%)*	0.289(▼ − 2.9%)*
2k	0.529(▲80%)*	0.502(▲70.6%)*	**0.548**(▲86.6%)*	0.534(▲81.7%)*
10k	0.593(▲6.4%)*	0.572(▲2.6%)	**0.612**(▲9.9%)*	0.603(▲8.3%)
100k	0.610(▲5.6%)	0.602(▲4.1%)	**0.620**(▲7.3%)	0.610(▲5.5%)
DistilBERT				
100	**0.258**(▲25.8%)#	0.235(▲14.5%)	0.241(▲17.6%)	0.212(▲3.37%)
1k	0.372(▲70%)*	0.368(▲68.1%)*	**0.401**(▲83%)*	0.236(▲7.9%)
2k	0.517(▲85.7%)*	0.505(▲81.5%)*	**0.532**(▲91.1%)*	0.485(▲74.3%)*
10k	0.573(▲1.4%)	0.563(▼ − 0.4%)	**0.582**(▲2.9%)	0.551(▼ − 2.6%)
100k	0.590(▼ − 2.6%)	0.610(▲0.6%)	**0.630**(▲3.3%)	0.561(▼ − 7.5%)

Statistically significant improvements at a level of 95% and 90% are indicated by * and #, respectively [16]. The best results for each dataset and each model is in bold.

Insight 2. Supervised augmentation techniques, specifically Attention leads to higher sample efficiency rates especially for smaller datasets (100, 1K, 2K) when training with RankingSCL.

RQ III. Which contrastive loss objective leads to the highest gains when data augmentation type and budget are fixed?

Different contrastive loss functions have different properties that may make them more or less effective for our particular problem. When the data augmentation type and training budget are fixed, we find that the contrastive loss objective which leads to the highest gains depends on specific characteristics of the dataset and the model being used. To this extent, we experiment with four different contrastive losses (RankingCTriplet, RankingNCA, RankingInfoNCE, RankingSCL) with fixed supervised augmentation techniques (Attention, Linear) for different models (BERT, RoBERTa, DistilBERT). The results are shown in Table 3 and Appendix Table 8.

In Table 3, we compare different loss objectives while using attention augmentation. It is clear that RankingCTriplet outperforms all other losses across varying dataset sizes. It shows more than 85% improvement over baseline (95% statistically significant) in the case of RoBERTa and DistilBERT when trained on the 2K dataset. Additionally, the performance of DistilBERT matches the performance of BERT and RoBERTa when training with the 100K dataset. Notice that for RoBERTa, the choice of the loss function is crucial to sample efficiency on the 1k dataset—with RankingCTriplet we see a 6.6% improvement.

The results also show that increasing the training set size generally leads to better performance across all models and loss functions. However, the rate of improvement varies depending on the model and loss function. For example, for RoBERTa, RankingCTriplet shows the largest

percentage change when increasing the training set size from 1k to 2k, while for DistilBERT, RankingNCA shows the largest percentage change.

RankingCTriplet loss has 2 unique properties—the margin hyperparameter and the usage of centroids rather than the actual data points. Max margin based losses are commonly used for ranking tasks [1], which makes this style of contrastive loss more suited to our problem. Additionally, since the augmentation techniques can be noisy, the usage of the centroid dampens it by averaging over all augmentations of the anchor. RankingCTriplet loss is generally used when there is more intra-class variation which is the case in document ranking where positive documents can vary drastically based on the query at hand. Other loss functions are more suited to classification problems and do not translate as well to our problem setting.

Our empirical findings suggest that pairing data augmentation with the right loss function is imperative to maximizing sample efficiency. Comparing RankingSCL loss to RankingCTriplet we see consistent gains across models and data sizes in terms of nDCG@10 and relative improvement to the baseline. Even for 100k data points, we see gains of 3.1% to 7.3% when using RankingCTriplet, which is considerably higher than all other losses.

Insight 3. Empirically CTL (RankingCTriplet) is the superior choice for all models for all dataset sizes when using attention based augmentation.

RQ IV. Does our training regimen impact various model sizes differently?

To look at the impact of model sizes on ranking performance, we conduct baseline and augmented data experiments on DistilBERT, BERT, and BERT-Large by varying the dataset sizes (100, 1K, 2K, 10K, 100K). For the baseline, we use non-augmented data same as all experiments above. For augmented experiments, we use Attention augmentation with RankingCTriplet loss. The performance results (nDCG@10) results are shown in Figure 4.

We observe that larger BERT-based language models tend to perform better on smaller datasets, indicating that increasing the model size can compensate for the limited amount of data. Furthermore, we already showed, augmented models outperform their non-augmented counterparts (Insight 1). However, it is worth noting that as the dataset size increases, the performance gap between smaller and larger models narrows, implying that increasing model size may have diminishing returns in the presence of large datasets.

For DistilBERT, we see that data augmentation leads to large improvements. For practitioners with limited computing budgets, data augmentation provides a significant method to improve the performance of leaner models. Fine tuning lean models on smaller ranking datasets is not always straightforward. We experimented with several hyperparameters to improve the performance of the DistilBERT baseline for 1k and 2k. However, with augmentation minimal hyperparameter tuning led to gains of over 50% in sample efficiency. Hard-to-tune models can benefit directly from our approach due to more informative samples and a better loss objective.

Insight 4. Larger models (BERT-Large) out perform smaller models (BERT, DistilBERT) in the case of smaller datasets (100, 1k, 2k, 10k) with augmentation primarily benefiting the smallest model (DistilBERT).

7.2 Out-of-Domain (BEIR) Experiments

RQ V. Do augmented models transfer better than their non-augmented counterparts?

Zero-shot out-of-domain transfer for ranking datasets is challenging due to the diversity of topical domains and the information needs of users. Table 4 shows the performance of three different BERT-based models (BERT, RoBERTa, and DistilBERT) we trained using our proposed paradigm

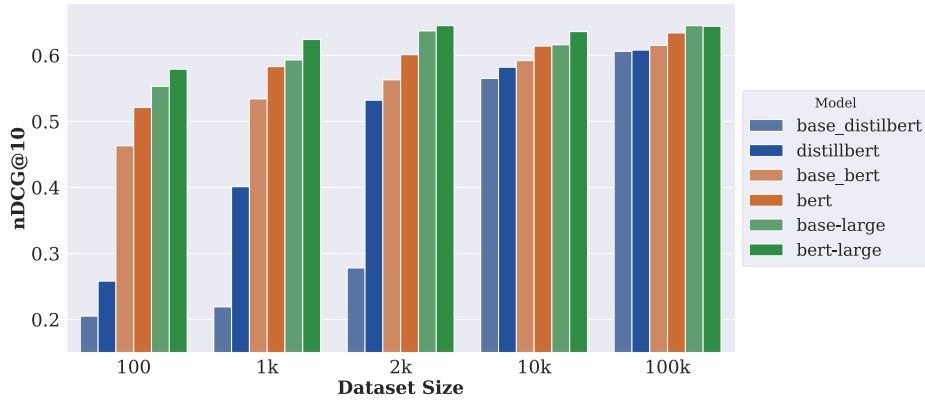

Fig. 4. Performance of models of different model sizes compared to their respective baselines. The values with "**base_**" represents the base model.

Table 4. nDCG@10 Values for Different Models (BERT, RoBERTa, DistilBERT) on the BEIR Dataset for Evaluating Out-of-Distribution Performance

Datasets	**BERT**	**RoBERTa**	**DistilBERT**	**SPLADE**
SciFact	0.678(▲3.1%)	0.676(▲134%)*	0.526(▲14.2%)*	**0.699**
Baseline	0.658	0.288	0.461	
FiQA	0.315(▲4.4%)*	0.339(▲50.6%)*	0.211(▲20.5%)*	**0.351**
Baseline	0.302	0.225	0.108	
DBPedia	0.481(▼ − 0.4%)*	**0.514**(▲36.3%)*	0.435(▲16.2%)*	0.442
Baseline	0.483	0.377	0.236	
TREC-COVID	0.694(▲2.7%)*	**0.721**(▲21.8%)*	0.603(▲2.3%)*	0.711
Baseline	0.676	0.592	0.589	
NFCorpus	0.260(▲0.9%)	0.282(▲18.8%)*	0.260(▲2.2%)*	**0.345**
Baseline	0.257	0.237	0.255	
Robust04	0.442(▲8.8%)#	0.429(▲30.3%)*	0.386(▲17%)*	**0.458**
Baseline	0.406	0.329	0.330	
Average	0.478	0.494	0.404	**0.501**

The model used for zero-shot performance are trained on 100K `Attention` dataset with `RankingCTriplet` loss. Statistically significant improvements at a level of 95% and 90% are indicated by * and # respectively [16].The best results for each dataset and each model is in bold.

on 6 different datasets against the current SOTA model (SPLADE [15]) on the BEIR benchmark without any further fine-tuning on each dataset.

We find that our augmented models are significantly better than their corresponding baselines making them not only sample efficient but also more robust. RoBERTa in particular sees large gains across all datasets and at times outperforms SOTA (DBPedia and TREC-COVID). Augmentation helps improve the model's ranking performance more broadly by exposing it to a wider range of examples and scenarios during training. More concretely, we believe that without augmentation, models suffer from the problem of over-fitting which can occur especially when a model is trained on a limited set of examples. Augmentation can help prevent over-fitting by providing the model with a larger and more diverse set of examples to learn from. Furthermore, the added loss objective also helps to regularize the model.

Table 5. Example Queries from BEIR

Dataset	Domain	Queries
Robust04	News	price fixing, Russian food crisis, ADD diagnosis treatment, Modern Slavery
FiQA	Finance	How are various types of income taxed differently in the USA?, Looking for good investment vehicle for seasonal work and savings, Understanding the T + 3 settlement days rule, How should I prepare for the next financial crisis?
Dbpedia	Entity	south korean girl groups, electronic music genres, digital music notation formats, FIFA world cup national team winners since 1974
Trec-Covid	Medical	what is known about those infected with Covid-19 but are asymptomatic?,what evidence is there for the value of hydroxychloroquine in treating Covid-19?
NFCorpus	Science	Is apple cider vinegar good for you?, How can you believe in any scientific study?, organotins, oxen meat

Each dataset varies in topicality and query type.

The BEIR datasets also varied in the type of queries—our training dataset Doc'19 has simple factual questions whereas DBPedia for instance has entity specific attribute queries and Robust has topical keywords as queries (Table 5 shows anecdotal evidence).

Our augmentation with contrastive learning exposes the model to more general matching patterns since the model has to not only learn how to estimate relevance between a question and a document but also a sentence and a passage selected by our augmentation. This helps improve the overall semantic understanding of the model which greatly aids transfer. These results hold for all model types which makes a strong case for our augmentation models to be used as universal ranking models when computing and training data is severely limited.

This result shows that we can save costs and increase efficiency by training a single model that performs well across IR datasets with limited amounts of out-of-domain training data. Practitioners need not invest in further fine tuning or deploying individual models for each dataset.

Insight 5. Augmented models are better suited for zero-shot transfer because our approach improves the model's ability to estimate relevance between two pieces of text by training on more diverse examples.

7.3 Limitations

In our experiments, we considered 3 contextual models, 4 losses, 4 augmentation techniques, and 4 dataset sizes. We ran experiments with all combinations of dataset size, loss and augmentation. We had 2 key questions to verify our insights from the previous sections:

— Is attention based augmentation always the right choice irrespective of the loss function?
— Is `RankingCTriplet` loss the best choice irrespective of augmentation technique?

From Figure 5, it is apparent that attention augmentation and `RankingCTriplet` are not always the best choice. Even though the best performance at 1k, 2k, and 10k is from the proposed combination, we see certain combinations being close to or surpassing it in the case of 100k (BM25 + `RankingSCL`).

In the case of 1k and 2k, `RankingCTriplet` performs considerably better than all other losses for unsupervised augmentation. When the dataset size increases however, the differences are much

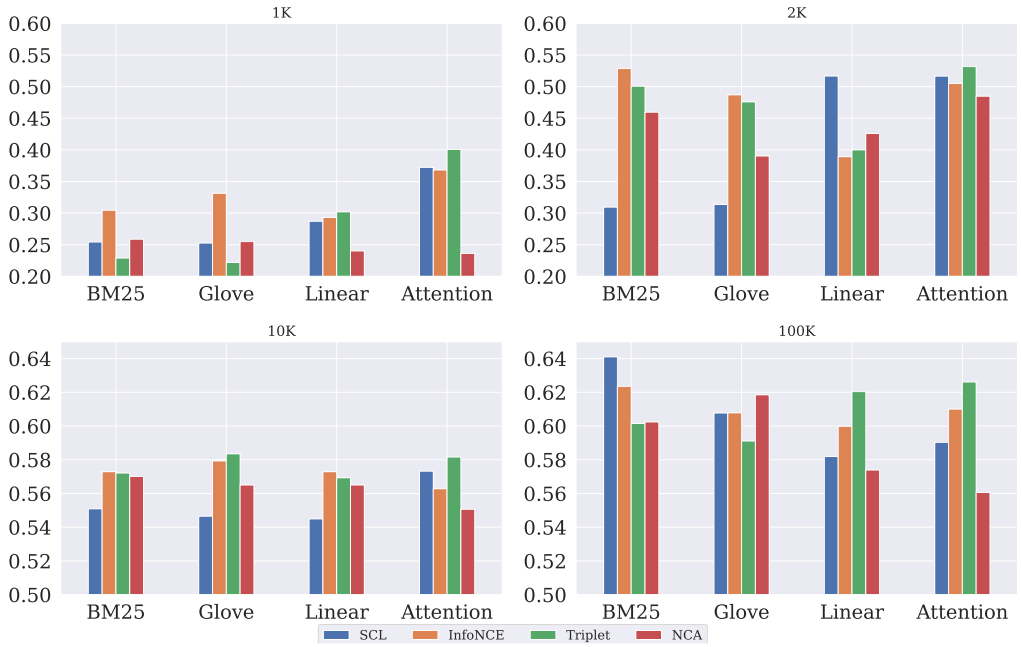

Fig. 5. DistilBERT nDCG@10 performance on difference size datasets with different augmentation and losses.

smaller. When using a 100K dataset, the choice of the loss function and augmentation is not clear. `RankingSCL` performs relatively poorly, especially for unsupervised augmentation in 1k and 2k but slightly outperforms our proposed combination for 100k. `RankingCTriplet` does not result in the best performance in low data regimes when using unsupervised augmentation.

In the previous section, we studied the impact of various losses on attention augmentation. Attention and linear augmentation techniques are both supervised selectors from Reference [33]. In both cases, the objective of the selector is to pick the most relevant sentence. The linear selector pools word embeddings using a max operation, so positional context is lost. This design choice leads to attention outperforming linear irrespective of the loss function and dataset size. For 1k and 2k, linear is outperformed by GloVe and BM25. This could mean that selecting noisy sentences for contrastive learning is worse than simple term matching when paired with the correct loss. Supervised augmentation alone is not sufficient to gain the best performance for 1k and 2k. It must be paired with the right loss. `RankingCTriplet` in general is seemingly a good choice for a loss function

If we consider the simplicity of implementation, the combination of BM25 and `RankingInfoNCE` is a good choice. It rivals `Attention` and `RankingCTriplet` in all datasets above 1k. For 1k however, attention based augmentation makes the largest difference since all losses except `RankingNCA` exhibit large improvements. The value of attention based augmentation compared to other augmentations diminishes as the data size grows. While augmentation still leads to overall sample efficiency gains, the choice of augmentation is not as crucial only if paired with the right loss function. Empirically trying to detect which combination works the best for a practitioner's dataset however goes against the spirit of efficiency. Our proposed combination of `RankingCTriplet` and `Attention` displays the clearest trend even if there are a few cases where it is not the best.

8 DISCUSSION AND CONCLUSION

In this article, we empirically explored the impact of data augmentation on sample efficiency for ranking problems. Our central premise is that data augmentation when paired with a contrastive learning objective leads to significant improvements in performance with the same number of training instances. Our experimental space includes both supervised and unsupervised augmentation techniques and 4 different contrastive learning objectives. We found that a combination of attention-based supervised augmentation and `RankingCTriplet` provides the highest sample efficiency gains for a range of models and dataset sizes. We see large benefits for smaller models on smaller dataset sizes which is an important step toward their wider adoption. DistilBERT sees gains of up to 85% when fine-tuned using our proposed setup. We also observe that not all models exhibit the same level of gains with BERT gaining between 1% and 10% depending on the size of the dataset. Another benefit of augmented training is the drastic improvement in zero-shot transfer. We showed that our best-augmented models improve performance by large margins compared with their non-augmented counterparts. RoBERTa on average sees a near 60% improvement when augmented and can rival fine-tuned SOTA models.

In conclusion, we believe the approach we propose is only the first step toward sample efficient training of ranking models with contrastive losses. Augmentation and adjusting loss objectives are cheaper alternatives for most practitioners instead of gathering expensive training data. There remain several areas of future work—for instance, augmentation techniques that can operate on queries is still under-explored. We still lack a clear understanding of the impact of various contrastive losses on the type of augmentation. Further research is needed to identify why a specific loss function benefits from a particular type of augmentation. We are also yet to explore the usage of synthetic training data in this context. Generative models have been used in the past to augment datasets with new queries, which we can also leverage in future work.

APPENDIX

A APPENDIX

Here, we add additional details relating to experimental setup and also show additional results.

We have a large experimental space. Here are the number models best trained : 5 Datasets * (3 Models * 4 Loss Function * 4 Augmentation Techniques) + (5 Datasets * 3 Models) Baseline = 255 models

— Datasets: 100, 1K, 2k, 10K, 100K
— Models: BM25, RoBERTa, DistilBERT
— Losses: `RankingSCL`, `RankingCTriplet`, `RankingNCA`, `RankingInfoNCE`
— Augmentation techniques: BM25, GloVe, `Attention`, `Linear`

This does not include intermediate models, models trained with different hyperparameters. That would increase the number of models trained by a factor of 5 on average.

A.1 Hyperparameters

Table 6. Learning Rates Used for Training Models for Different Datasets

Learning Rate	Model	Dataset size (# instances)
3e-3	BERT, RoBERTa, DistilBERT	100
1e-4	BERT, RoBERTa	1000; 2000; 10, 000
1e-5	BERT, RoBERTa	100, 000
3e-4	DistilBERT	1000; 2000; 10, 000
3e-5	DistilBERT	100, 000

A.2 Results

Here in Table 7, we compare the performance of full MS MARCO document dataset with 100K dataset. Both the datasets are augmented using BM25 augmentation and the loss used here is `RankingSCL`. We can see that the performance difference between the two datasets is not large. The absolute values start converging as the dataset grows.

Table 7. Comparing nDCG@10 Values for Different Models (BERT, RoBERTa, DistilBERT) on the Full MS MARCO and 100K Dataset with BM25 and `RankingSCL`

Model	**full**	**100K**
BERT	0.648(▲2.4%)	0.602(▲3.6%)
RoBERTa	0.652(▲11.2%)	0.598(▲2.9%)
DistilBERT	0.653(▲6.6%)	0.641(▲5.7%)

The percentage improvements in brackets are improvements from their respective baselines.

Table 8. nDCG@10 Performance of Different Language Models (BERT, RoBERTa, and DistilBERT) on Different Loss Functions (`RankingSCL`, `RankingInfoNCE`, `RankingCTriplet`, and `RankingNCA`) at Different Training Set Sizes (1k, 2k, 10k, and 100k) with **`Linear`** Data Augmentation

Losses	`RankingSCL`	`RankingInfoNCE`	`RankingCTriplet`	`RankingNCA`
BERT				
1k	0.554(▲3.8%)	**0.572**(▲7.0%)	0.541(▲1.2%)	0.558(▲4.4%)
2k	0.592(▲5%)	0.584(▲3.8%)	0.590(▲4.8%)	**0.603**(▲7.1%)
10k	0.600(▲1.4%)	0.607(▲2.7%)	**0.620**(▲4.7%)	0.605(▲2.2%)
100k	0.628(▲2.2%)	0.610(▼ − 0.9%)	**0.637**(▲3.6%)	0.626(▲1.8%)
RoBERTa				
1k	**0.288**(▼ − 3%)	0.287(▼ − 3.5%)*	0.271(▼ − 9%)#	0.273(▼ − 8%)
2k	**0.516**(▲75.6%)*	0.427(▲45.4%)*	0.421(▲43.3%)*	0.470(▲59.9%)*
10k	0.594(▲6.6%)*	0.570(▲2.3%)	**0.609**(▲9.3%)*	0.588(▲5.6%)*
100k	**0.637**(▲10.2%)	0.619(▲7.1%)	0.615(▲6.5%)	0.632(▲9.4%)
DistilBERT				
1k	0.287(▲31.1%)*	0.293(▲33.8%)*	**0.302**(▲37.9%)*	0.240(▲9.7%)
2k	**0.517**(▲85.7%)*	0.389(▲39.9%)*	0.400(▲43.8%)*	0.426(▲53.1%)*
10k	0.545(▼ − 3.6%)	**0.573**(▲1.4%)	0.569(▲0.7%)	0.565(▲0.0%)
100k	0.582(▼ − 4%)	0.600(▼ − 1.1%)	**0.620**(▲2.3%)	0.574(▲5.3%)

Statistically significant improvements at a level of 95% and 90% are indicated by * and # respectively [16].The best results for each dataset and each model is in bold.

REFERENCES

[1] Shivani Agarwal and Michael Collins. 2010. Maximum margin ranking algorithms for information retrieval. In *Proceedings of the Advances in Information Retrieval: 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings 32*. Springer, 332–343.

[2] Abhijit Anand, Jurek Leonhardt, Koustav Rudra, and Avishek Anand. 2022. Supervised contrastive learning approach for contextual ranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. 61–71.

[3] Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. FIRE 2019 AILA track: Artificial intelligence for legal assistance. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*. 4–6.

[4] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2387–2392.

[5] Kiran Butt and Abid Hussain. 2021. Evaluation of scholarly information retrieval using precision and recall. *Library Philosophy and Practice* (2021), 1–11.

[6] Wei-Cheng Chang, X. Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *8th International Conference on Learning Representations (ICLR'20)* Addis Ababa, Ethiopia, April 26-30, 2020. https://openreview.net/forum?id=rkg-mA4FDr

[7] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2020. Gridmask data augmentation. arXiv:2001.04086. Retrieved from https://arxiv.org/abs/cs/001.04086

[8] Yanda Chen, Chris Kedzie, Suraj Nair, Petra Galuščáková, Rui Zhang, Douglas W. Oard, and Kathleen Mckeown. 2021. Cross-language sentence selection via data augmentation and rationale training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 3881–3895.

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2019. TREC-2019-Deep-Learning. Retrieved from https://microsoft.github.io/TREC-2019-Deep-Learning/. (2019).

[10] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 113–123.

[11] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the ACM SIGIR'19*. 985–988.

[12] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/pdf?id=gmL46YMpu2J

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. [n. d.]. 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)*. 4171.

[14] Jeff Donahue and Karen Simonyan. 2019. Large scale adversarial representation learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 10542–10552.

[15] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2288–2292.

[16] Luke Gallagher. 2019. Pairwise t-test on TREC Run Files. Retrieved from https://github.com/lgrz/pairwise-ttest/. (2019).

[17] Luyu Gao and Jamie Callan. 2021. Condenser: A pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 981–993. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.75

[18] Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2843–2853. DOI: https://doi.org/10.18653/v1/2022.acl-long.203

[19] Xiang Gao, Ripon K. Saha, Mukul R. Prasad, and Abhik Roychoudhury. 2020. Fuzz testing based data augmentation to improve robustness of deep neural networks. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 1147–1158.

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised Representation Learning by Predicting Image Rotations. In *6th International Conference on Learning Representations (ICLR'18)*, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. https://openreview.net/forum?id=S1v4N2l0-

[21] Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. 2004. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, (NIPS'04, December 13-18, 2004, Vancouver, British Columbia, Canada)*. 513–520.

[22] Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for Pre-trained language model fine-tuning. In *9th International Conference on Learning Representations (ICLR'21)*. Virtual Event, Austria. https://openreview.net/forum?id=cu7IUiOhujH

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.

[24] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations (ICLR'19)*, New Orleans, LA.

[25] Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. 2020. Local self-attention over long text for efficient document retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021–2024.

[26] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable & time-budget-constrained contextual- ization for re-ranking. In *ECAI 2020 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain-Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS'20)*. IOS Press, 1–8.

[27] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2021. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2021), 2.

[28] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6769–6781.

[29] 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*. 39–48.

[30] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 18661–18673.

[31] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using Pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. 18–26.

[32] Carlos Lassance, Hervé Dejean, and Stéphane Clinchant. 2023. An experimental study on pretraining transformers from scratch for IR. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I*. 504–520.

[33] Jurek Leonhardt, Koustav Rudra, and Avishek Anand. 2023. Extractive explanations for interpretable text ranking. *ACM Transactions on Information Systems* 41, 4 (2023), 1–31.

[34] Jurek Leonhardt, Koustav Rudra, Megha Khosla, Abhijit Anand, and Avishek Anand. 2022. Efficient neural ranking using forward indexes. In *WWW'22: The ACM Web Conference 2022*, Virtual Event, Lyon, France, 266–276.

[35] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2023. PARADE: Passage representation aggregation for document reranking. *ACM Transactions on Information Systems* 42, 2 (2023), 1–26.

[36] Minghan Li, Diana Nicoleta Popa, Johan Chagnon, Yagmur Gizem Cinar, and Eric Gaussier. 2023. The power of selecting key blocks with local pre-ranking for long document information retrieval. *ACM Transactions on Information Systems* 41, 3 (2023), 1–35.

[37] Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. More robust dense retrieval with contrastive dual learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 287–296.

[38] Yijiang Lian, Zhenjun You, Fan Wu, Wenqiang Liu, and Jing Jia. 2020. Retrieve synonymous keywords for frequent queries in sponsored search in a data augmentation way. arXiv:2008.01969. Retrieved from https://arxiv.org/abs/cs/2008.01969

[39] Jimmy Lin. 2019. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*, Vol. 52. ACM, 40–51.

[40] Erik Lindgren, Sashank Reddi, Ruiqi Guo, and Sanjiv Kumar. 2021. Efficient training of retrieval models using negative cache. In *Advances in Neural Information Processing Systems*. M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 4134–4146. Retrieved from https://proceedings.neurips.cc/paper/2021/file/2175f8c5cd9604f6b1e576b252d4c86e-Paper.pdf

[41] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. 2016. Large-margin softmax loss for convolutional neural networks. In *Proceedings of the ICML*, Vol. 2. 7.

[42] 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. arXiv:1907.11692. Retrieved from https://arxiv.org/abs/cs/1907.11692

[43] Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: (EMNLP'20)*. 4401–4411.

[44] Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. arXiv:2305.02156. Retrieved from https://arxiv.org/abs/cs/2305.02156

[45] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104.

[46] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 119–126.

[47] Markus Mühling, Nikolaus Korfhage, Kader Pustu-Iren, Joanna Bars, Mario Knapp, Hicham Bellafkir, Markus Vogelbacher, Daniel Schneider, Angelika Hörth, Ralph Ewerth, and Bernd Freisleben. 2022. VIVA: Visual information retrieval in video archives. *International Journal on Digital Libraries* 23, 4 (2022), 319–333.

[48] Thai-Son Nguyen, Sebastian Stueker, Jan Niehues, and Alex Waibel. 2020. Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation. In *Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7689–7693.

[49] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. arXiv:1901.04085. Retrieved from https://arxiv.org/abs/cs/1901.04085

[50] Helmi Satria Nugraha and Suyanto Suyanto. 2019. Typographic-based data augmentation to improve a question retrieval in short dialogue system. In *Proceedings of the 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 44–49.

[51] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748. Retrieved from https://arxiv.org/abs/cs/1807.03748

[52] Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2021. Data augmentation for spoken language understanding via pretrained language models. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association*. 1219–1223. https://doi.org/10.21437/Interspeech.2021-117

[53] Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2021. CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*. 3853–3860.

[54] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Freisleben. 2023. Large language models are effective text rankers with pairwise ranking prompting. arXiv:2306.17563. Retrieved from https://arxiv.org/abs/cs/2306.17563

[55] 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*. Association for Computational Linguistics, Online, 5835–5847. DOI: https://doi.org/10.18653/v1/2021.naacl-main.466

[56] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. 2021. Automatic Data augmentation for generalization in reinforcement learning. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 5402–5415.

[57] Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2021. Synthetic data augmentation for zero-shot cross-lingual question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 7016–7030.

[58] Koustav Rudra and Avishek Anand. 2020. Distant supervision in BERT-based adhoc document retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2197–2200.

[59] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108. Retrieved from https://arxiv.org/abs/cs/1910.01108

[60] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.

[61] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of Big Data* 8, 1 (2021), 1–34.

[62] Jaspreet Singh, Wolfgang Nejdl, and Avishek Anand. 2016. History by diversity: Helping historians search news archives. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*. 183–192.

[63] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems 29, Annual Conference on Neural Information Processing Systems*. 1857–1865.
[64] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, S. Yu Philip, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*. 3436–3440.
[65] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, (EMNLP'23)*. 14918–14937.
[66] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 464–473. DOI: https://doi.org/10.18653/v1/P16-1044
[67] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. Retrieved from https://openreview.net/forum?id=wCu6T5xFjeJ
[68] Hoang Van, Vikas Yadav, and Mihai Surdeanu. 2021. Cheap and good? simple and effective data augmentation for low resource machine reading. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2116–2120.
[69] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the International Conference on Machine Learning*. PMLR, 9929–9939.
[70] Mikołaj Wieczorek, Barbara Rychalska, and Jacek Dąbrowski. 2021. On the unreasonable effectiveness of centroids in image retrieval. In *Proceedings of the International Conference on Neural Information Processing*. Springer, 212–223.
[71] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3733–3742.
[72] 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'21)*, Virtual Event, Austria. https://openreview.net/forum?id=zeFrfgyZln
[73] Nan Yang, Furu Wei, Binxing Jiao, Daxing Jiang, and Linjun Yang. 2021. xmoco: Cross momentum contrastive learning for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 6120–6129.
[74] Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. Data augmentation for bert fine-tuning in open-domain question answering. arXiv:1904.06652. Retrieved from https://arxiv.org/abs/cs/1904.06652
[75] Yinfei Yang, Ning Jin, Kuo Lin, Mandy Guo, and Daniel Cer. 2021. Neural retrieval for question answering with cross-attention supervised data augmentation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 263–268.
[76] Liang Yao, Baosong Yang, Haibo Zhang, Boxing Chen, and Weihua Luo. 2020. Domain transfer based data augmentation for neural query translation. In *Proceedings of the 28th International Conference on Computational Linguistics*. 4521–4533.
[77] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. 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)*. 3490–3496.
[78] Yi Zeng, Han Qiu, Gerard Memmi, and Meikang Qiu. 2020. A data augmentation-based defense method against adversarial attacks in neural networks. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 274–289.
[79] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. *Optimizing Dense Retrieval Model Training with Hard Negatives*. Association for Computing Machinery, New York, NY, 1503–1512. DOI: https://doi.org/10.1145/3404835.3462880
[80] Xingyu Zhang, Tong Xiao, Yidong Chen, and Qun Liu. 2021. Text augmentation for neural machine translation: A review. arXiv:2103.09065. Retrieved from https://arxiv.org/abs/cs/2103.09065
[81] Zijian Zhang, Koustav Rudra, and Avishek Anand. 2021. Explain and predict, and then predict again. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 418–426.
[82] Zhilu Zhang and Mert R Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS)*.

[83] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13001–13008.

[84] Qingqing Zhu, Xiwei Wang, Chen Chen, and Junfei Liu. 2020. Data augmentation for retrieval-and generation-based dialog systems. In *Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, 1716–1720.

[85] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive learning of user behavior sequence for context-aware document ranking. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. Association for Computing Machinery, New York, NY, 2780–2791.DOI: https://doi.org/10.1145/3459637.3482243

Received 1 March 2023; revised 15 August 2023; accepted 20 November 2023