Supervised Contrastive Learning Approach for Contextual Ranking

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

DOI
10.1145/3539813.3545139

Publication date
2022

Document Version
Final published version

Published in
ICTIR 2022 - Proceedings of the 2022 ACM SIGIR International Conference on the Theory of Information Retrieval

Supervised Contrastive Learning Approach for Contextual Ranking

Abhijit Anand
L3S Research Center
Hannover, Germany
aanand@L3S.de

Jurek Leonhardt
L3S Research Center
Hannover, Germany
leonhardt@L3S.de

Koustav Rudra*
Indian Institute of Technology
(Indian School of Mines)
Dhanbad, India
koustav@iitism.ac.in

Avishek Anand*
Delft University of Technology
Delft, Netherlands
avishek.anand@tudelft.nl

ABSTRACT

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.

This paper proposes a simple yet effective method to improve ranking performance on *smaller datasets* using *supervised contrastive learning* for the document ranking problem. We perform data augmentation by creating training data using parts of the relevant documents in the query-document pairs. We then use a supervised contrastive learning objective to learn an effective ranking model from the augmented dataset. Our experiments on subsets of the TREC-DL dataset show that, although data augmentation leads to an increasing the training data sizes, it does not necessarily improve the performance using existing pointwise or pairwise training objectives. However, our proposed supervised contrastive loss objective leads to performance improvements over the standard non-augmented setting showcasing the utility of data augmentation using contrastive losses. Finally, we show the real benefit of using supervised contrastive learning objectives by showing marked improvements in smaller ranking datasets relating to news (Robust04), finance (FiQA), and scientific fact checking (SciFact).

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

document ranking, supervised contrastive loss, data augmentation, interpolation, ranking performance

*Research was primarily conducted while affiliated to L3S Research Center.

ICTIR '22, July 11–12, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545139

ACM Reference Format:
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 the Theory of Information Retrieval (ICTIR '22), July 11–12, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539813.3545139

1 INTRODUCTION

Recent approaches for ranking documents have focused heavily on contextual transformer-based models for both retrieval [22, 33] and re-ranking [7, 19, 28, 40, 72]. To further improve the effectiveness of contextual ranking models, earlier works have explored negative sampling techniques [65], pre-training approaches [2], and different architectural variants [19, 22]. In this paper we investigate the use of simple yet effective data augmentation techniques for ad-hoc document retrieval.

Data augmentation (DA) encompasses methods of increasing training data without directly collecting more data but by either adding slightly modified copies of existing data or creating synthetic data. Data augmentation has been successfully used to help train more robust models, particularly when using smaller datasets in computer vision [55], speech recognition [45], spoken language understanding [50], and dialog system [78]. Most of the augmentation approaches are based on a heuristic set of rules based on well-understood domain-specific phenomena. However, the use of data augmentation for document ranking has not been investigated in detail to the best of our knowledge.

Both in NLP and IR tasks, the use of large amounts of language data to *pre-train* an initial version of an contextual model, followed by refinement or *fine-tuning* using a small amount of domain-specific data this has delivered impressive gains both in sample efficiency and better generalization. However, popular contextualized models are over-parameterized with more than *100 million* parameters and might overfit the training data when the task-specific fine-tuning data is small. Training data for rankings can be either small due to smaller query workloads [61] or incomplete labels as in [44] and this is where data augmentation techniques can be valuable. However, simply augmenting training data with existing point- or pairwise ranking losses does not lead to performance improvements. In fact, we show that our data augmentation techniques using existing pointwise ranking losses, i.e. cross-entropy losses,

results in *degradation of performance* in many cases. This can be attributed to known lack of robustness to noisy labels [76] and the possibility of poor margins [35], leading to reduced generalization performance.

Towards improving training in the *limited data setting* using data augmentation, we propose *supervised contrastive learning objectives* (`RankingSCL`) for document ranking. A key idea in contrastive learning is to learn the input representation of an instance or *anchor* such that its positive instances are embedded closer to each other, and the negative samples are farther apart. In this work, we construct **augmented query-document** from existing positive instances by multiple augmentation strategies. We extend the idea of supervised contrastive learning (SCL) 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. An important technical challenge while extending SCL loss to ranking data is that of *data sparsity of positive pairs*. One could, in principle, decrease data sparsity by including multiple positive instances of the query in the same batch. However, decreasing sparsity also decreases randomness, which is crucial in training generalizable ranking models. Towards this, we propose a *practical batching strategy* that maximizes randomness while allowing for augmented query-document pairs.

We conduct extensive experiments on multiple contextual models – BERT, RoBERTa, DistilBERT – and multiple low-data ranking settings to establish the effectiveness of our approaches. Note that our primary aim in this paper is to improve ranking performance for smaller ranking datasets by using simple data-augmentation techniques. We do not intend to engineer a state-of-art ranking model for document ranking but instead focus on optimization strategies that work well for simple data augmentation techniques in low data settings.

1.1 Research Questions

RQ I. Does data augmentation or Supervised Contrastive Learning help to improve document re-ranking performance for smaller datasets?
RQ II. Does the augmentation style impact the performance?
RQ III. How does training data size impact performance?

Towards answering these research questions we conduct extensive experiments on the following ranking datasets – TREC-DL, Robust04, FiQA, and SciFact. While TREC-DL and Robust04 contain longer documents and under-specified queries, FiQA is a question-answering dataset over financial text, and SciFact deals with fact checking queries. Note that FiQA and SciFact are much smaller datasets compared to TREC-DL.

1.2 Contributions

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

- We propose and study data augmentation approaches for document ranking.
- We propose a ranking-based supervised contrastive loss for exploiting positive augmented pairs for improving ranking performance.
- We show that our ranking-based SCL delivers substantial performance improvements for a wide variety of ranking models under both low and high data settings.

2 RELATED WORK

The related work can be broadly divided into three areas. We start by outlining prominent strategies for document ranking using contextual models. Next, we review various data augmentation strategies and their application in text tasks. Finally we reflect upon various loss functions used in text ranking and their relationship with supervised contrastive loss.

2.1 Contextual Models for Ad-hoc Document Retrieval

A standard strategy for the text ranking task involves a fast retrieval step followed by a more involved *re-ranking* step. In this paper, we are concerned with improving the performance of the re-ranking stage that typically involves the use of contextual models. Contextual models like [8, 36] have shown promising improvements in document ranking task [7, 28, 53].

There are two major paradigms to encode the input, i.e., query document pairs, for training a contextual re-ranker – (a) joint encoding, and (b) independent encoding. The most common way for applying contextual models for the problem of document re-ranking is to **jointly encode** the query and document using a over-parameterized language model [40, 46]. Independent encoding, on the other hand, encoding the document and the query independently of each other. Such models that implement independent query and document encoding are called **dual encoders**, **bi-encoders**, or **two-tower models**. Typically, dual encoders are used in the retrieval phase [1, 2, 19, 21, 22, 25] however there have been recent proposals that use dual encoders in the re-ranking phase [27]. Note that a common problem in both approaches is due to an upper bound on the acceptable input length of contextual models that restricts its applicability to shorter documents. When documents do not fit into the model the documents are chunked into passages/sentences to fit within token limit either by using transformer-kernels [18, 19], truncation [7], or careful pre-selection of relevant text [26, 53].

In this work, we focus on the joint encoding models for document ranking and employ simple document truncation whenever longer documents overflow the overall input upper bound.

2.2 Data Augmentation

Data augmentation has a significant impact in different segments such as text, speech, image, vision, etc. Researchers proposed new data augmentation strategies [3, 6, 77] and their influence on deep learning models [13, 37, 51, 74]. Data augmentation helps in speech recognition [45], spoken language understanding [50], and dialog system [78]. Data augmentation [24, 56, 59] using pre-trained transformer models show significant boost in the performance of several downstream natural language processing (nlp) and text related tasks. Morris et al [42] proposed a framework *Text Attack* for data augmentation, adversarial attacks, and training in nlp. Different natural language tasks such as named entity recognition [34], language inference [11], text categorization, classification [38, 73],

query based multi-document summarization [49]. Image and vision related tasks also significantly benefit through data augmentation. Shorten et al [55] provides a survey on the role of image data augmentation strategy on deep learning. Data augmentation helps to boost performance in multiple image/vision related tasks such as user identification [41], image retrieval [67], image segmentation [66], text recognition and object detection [29, 39, 67, 79].

Recently, data augmentation strategies have been deployed for retrieval tasks. It shows promising results for question retrieval [47], query translation [71], question-answering [69, 70], cross-language sentence selection [4], machine reading [60], query expansion [32]. Yang et al [68] proposed cross-momentum contrastive learning [16] based open-domain question answering scheme. Recent dense retriever models [21, 65] sample negative documents to train dense retrievers in contrastive way. However, such methods do not take care of uniformity nature of contrastive learning [63]. Li et al [30] proposed a contrastive dual learning based method for dense retrieval that takes care of uniformity. Most of these strategies focus on negative samples and try to train an efficient dense retriever framework. Data augmentation strategy with contrastive loss setup is also not yet explored for document ranking task. In this paper, we take a step towards that and explore the effect of different *data augmentation strategies* with *supervised contrastive learning* setup on the re-ranking performance.

2.3 Supervised Contrastive Learning (SCL)

Contrastive losses using data augmentation have been popularized in the machine learning literature in the unsupervised learning setting. Specifically, augmentation of an instance are treated as positive samples and other random instances from the batch are treated as negative samples. We are inspired from the recent idea of contrastive loss that also exploit the label information for more fine-grained supervision signal from data augmentation [23]. Recent methods utilize this approach to learn representations from unsupervised data [17, 48, 57, 64] and they outperform other approaches [10, 14]. Training instances are generated from original ones using different data augmentation strategies and contrastive loss helps to bring the representation of similar/related entities close to each other in the embedding space. For a more detailed overview we point the interested reader to a recent survey on supervised and self-supervised contrastive learning [20]. Recently, SCL has been applied to fine-tuning regimes using pre-trained language models but with limited success [15], and also for the retrieval stage (not re-ranking) [31]. To the best of our knowledge, SCL has not been used in document ranking using joint encoder models.

The learning objective of neural ranking models is broadly studied under three types – pointwise, pairwise, and list-wise losses. A pointwise learning objective tries to optimize a ranking model by directly predicting the relevance class using for example the widely popular cross-entropy loss. Pairwise ranking objectives, on other hand, focus on optimizing the preference-pairs induced by the document labels in the training dataset. Note that pairwise losses aim to always distinguish between different labels, i.e., relevant vs. irrelevant or highly relevant vs. relevant. Finally, list-wise losses directly try to optimize the ranking as a whole. In principle, contrastive losses can be used in conjugation with any of the aforementioned losses and in this paper we experiment with pointwise and pairwise losses.

The idea of supervised contrastive loss has its roots in self-supervised contrastive learning.

3 METHOD

In this section we begin by defining the document re-ranking problem (cf. Section 3.1). We then describe our contributions, which comprise multiple training data augmentation techniques for re-ranking data (Section 3.3) and a supervised contrastive learning objective which is used to train our ranking models (Section 3.2).

3.1 Contextual Document Rankers

In this paper we aim to learn a *document re-ranking* model. Given a query-document pair (q, d) as input, the model outputs a *relevance score*. This relevance score may then be used to rank a number of documents with respect to their relevance to a given query.

Formally, we have a training set of 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 to it, depending on its label y_i. Our goal is to train a ranker R that predicts a relevance score $\hat{y} \in [0; 1]$ given a query q and a document d:

$$R : (q, d) \mapsto \hat{y} \tag{1}$$

Finally, the trained ranking model R can be used to re-rank a set of documents obtained in the first-stage retrieval process by a light-weight, typically term-frequency-based, retriever w.r.t. a query. This is the usual practice for ranking tasks, where the documents are retrieved first and then re-ranked by a more involved and computationally expensive model. In recent times, pre-trained contextual language models have shown promising performance for document ranking task [7, 46, 53, 72]. Such cross-attention models jointly model queries and documents. In this paper, we consider three different joint modeling approaches based on BERT [8], RoBERTa [36] and DistilBERT [54] and check their performance under supervised contrastive learning setup with different amount 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

$$\texttt{[CLS]}\ q\ \texttt{[SEP]}\ d\ \texttt{[SEP]}. \tag{2}$$

Due to the input length limitation of the models, long documents may be truncated to fit.

3.2 Supervised Contrastive Learning for Rankings

For training, we operate in the mini-batch training setup with a batch of training examples $\{x_i, y_i\}_{i=1,\ldots,N}$. Traditionally, ranking models are often trained in one of the following ways:

In **pointwise** training, the document ranking task is considered as a binary classification problem with a relevant and a non-relevant class. Each training instance $x_i = (q_i, d_i)$ is a query-document pair and $y_i \in \{0, 1\}$ is a relevance label. Let $\hat{y}_i$ be the predicted score of x_i. The cross-entropy loss function can be written as follows:

$$\mathcal{L}_{\texttt{Point}} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \tag{3}$$

In **pairwise** training, each training example consists of a query and two documents, i.e. $x_i = (q_i, d_i^+, d_i^-)$, where the former is more relevant to the query than the latter. The pairwise loss function is

$$\mathcal{L}_{\texttt{Pair}} = \frac{1}{N}\sum_{i=1}^{N} \max\left\{0, m - \hat{y}_i^+ + \hat{y}_i^-\right\} \tag{4}$$

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*.

We propose a novel ranking objective that includes a **supervised contrastive learning** (SCL) term for fine-tuning contextual ranking models in addition to the standard ranking loss. The SCL loss is meant to capture the similarities between relevant parts of documents for the same query and contrast them with the examples from non-relevant queries. Let $\Phi(\cdot) \in \mathbb{R}^t$ denote the query-document representation that is output by the ranking model (for example, this corresponds to the `[CLS]` output for BERT-based models). Let N_+ be the total number of positive examples in the batch (relevant query-document pairs). $\tau > 0$ is an adjustable scalar temperature parameter that controls the separation between the relevant and non-relevant examples and λ is a scalar weighting hyper-parameter that we tune for each downstream task and setting. The SCL loss can be written as

$$\mathcal{L}_{\texttt{SCL}} = \sum_{i=1}^{N} -\frac{1}{N_+}\sum_{j=1}^{N_+} \mathbf{1}_{\substack{q_i=q_j,\\ i\neq j,\\ y_i=y_j=1}} \log \frac{\exp\left(\Phi\left(x_i\right)\cdot\Phi\left(x_j\right)/\tau\right)}{\sum_{k=1}^{N}\mathbf{1}_{i\neq k}\exp\left(\Phi\left(x_i\right)\cdot\Phi\left(x_k\right)/\tau\right)} \tag{5}$$

Note that $\mathcal{L}_{\texttt{SCL}}$ constrains the positive pair that has the same query to be embedded close to each other instead of a pair of documents that are relevant for different queries. This is crucial since we want to enforce that the representations for the "relevant parts" of the same query be close to each other.

The overall ranking SCL loss is then given by

$$\mathcal{L}_{\texttt{RankingSCL}} = (1-\lambda)\mathcal{L}_{\texttt{Ranking}} + \lambda\mathcal{L}_{\texttt{SCL}} \tag{6}$$

where $\mathcal{L}_{\texttt{Ranking}}$ may be either $\mathcal{L}_{\texttt{Point}}$ or $\mathcal{L}_{\texttt{Pair}}$. We illustrate the RankingSCL loss in Figure 1 using a pairwise 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. We use the following terminology in the paper: linear interpolation of Pointwise and RankingSCL is referred to as **Pointwise RankingSCL** and linear interpolation of Pairwise and RankingSCL is referred to as **Pairwise RankingSCL**.

3.3 Creating and Augmenting Training Batches

During the creation of mini-batches, our objective is to preserve the randomness in the data while augmenting the training set. Previous studies showed that the performance of self-supervised contrastive learning depends on the quality of the augmented data [15].

We start with the top-k documents per query retrieved using a first stage retrieval method. We create the training dataset by collecting all positive query-document training instances from this top-k set. We then randomly sample one irrelevant document for each positive pair. The resulting training set of (q, d^+, d^-) triples is shuffled to ensure randomness. Note that, for pointwise training, we create two query-document pairs from each triple.

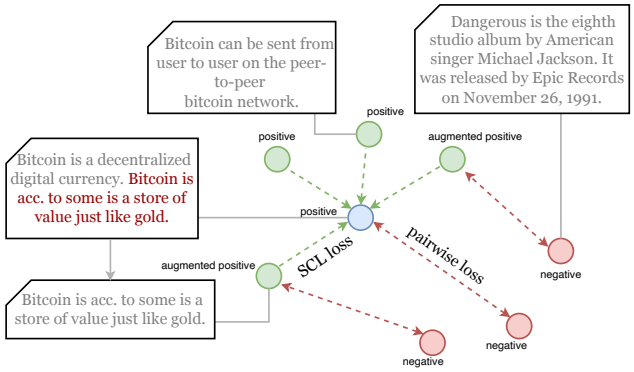


Figure 1: Example batch using data augmentation for rankings for the example query `bitcoin currency`. Augmented positives are derived from a relevant document, negatives are randomly sampled from the batch. SCL loss tries to bring the representations of positives closer, while pairwise loss repels the representations of positives and negatives apart.

Algorithm 1: Training data augmentation

Input: training batch B
Output: augmented training batch B'
$B' \leftarrow$ empty list
foreach (q, d^+, d^-) **in** B **do**
  `// keep the original example`
  append (q, d^+, d^-) to B'
  `// create augmented example`
  $d_a^+ \leftarrow \texttt{augment}(d^+, q)$
  $d_a^- \leftarrow$ random irrelevant document
  append (q, d_a^+, d_a^-) to B'
end
return B'

3.3.1 ***Data Augmentation.*** Next, we augment the training instances. For each triple (q, d^+, d^-) in the training set, we create an augmented version d_a^+ of d^+ by selecting relevant sentences with respect to the corresponding query and randomly sample an irrelevant document d_a^- from the corpus. The augmented training instances are appended to the respective batch. Consequently, after augmentation, each batch contains twice as many training instances. The augmentation approach is illustrated in Algorithm 1.

In order to augment a document, we consider it as a sequence of sentences s_i, i.e. $d = (s_1, s_2, ..., s_{|d|})$. A query-specific selector selects a fixed number of sentences from the document. The selector defines a distribution $p(s|q, d)$ over sentences in d given the input query q, encoding the relevance of the sentence given the query. This distribution is used to select an extractive, query-dependent summary $d' \subseteq d$. Here, we extract a summary as the 20 highest scoring sentences.

We consider the following three sentence selection strategies:

- **Embedding-based (GloVe)**: We use semantic (cosine) similarity scores between the query q and sentences s_i to determine the best sentences. Both the query and sentence are represented as average over the constituent word embeddings.
- **Term-matching-based (BM25)**: We use tf-idf scores between the query q and sentences s_i to determine the best sentences. Inverse document frequencies are computed over the complete corpus.
- **Sampling-based (Sampling)**: We randomly sample k sentences from the document.

4 EXPERIMENTAL SETUP

In our experiments we answer the following research questions:

RQ I. Does data augmentation or Supervised Contrastive Learning help to improve document re-ranking performance for smaller datasets?
RQ II. Does the augmentation style impact the performance?
RQ III. How does training data size impact performance?

Towards answering these research questions we employ the following datasets, rankers and training settings. Note that we focus on the re-ranking task and not the retrieval task.

Datasets. We conduct experiments on the following ranking datasets:

(1) **TREC-DL**: We consider the dataset from the TREC Deep Learning track (2019). We evaluate our model on Doc'19 and Doc'20 containing 200 queries each. For training and dev set we use MS MARCO which contains 367K queries. Top 100 documents are retrieved for each query using Indri [5].
(2) **Robust04**: We have 249 queries with their description and narratives. Along with queries, we also have a 528K document collection. Top 100 documents are retrieved for each query using Indri [58] framework. We consider the folds and top documents retrieved directly from Dai and Callan [7].
(3) **FiQA** was released in 2018 as an open challenge in the Web Conference.[1] It comprises questions and answers from the financial domain, with one of two tasks being *opinion-based QA over financial data*. The QA test collection was crawled from Stackexchange, Reddit and StockTwits. We have in total 6650 queries of which 650 are present in test set and 5500 in training set. The corpus size is around 57K. The top-100 documents are retrieved for each query using BM25.
(4) **SciFact** [62] is a *scientific fact verification* dataset. We have 1110 queries of which 810 are in the training set and 300 in test set with a document corpus size of 5K. We retrieve the top-100 documents per query using BM25. The dataset contains scientific claims written by experts as well as annotated abstracts that may support or refute a given claim. We treat the fact verification task as a ranking problem by retrieving relevant documents for a given query (fact) from the whole corpus.

[1] https://sites.google.com/view/fiqa/home

Ranking Models. We use three different cross-attention models for our experiments:

(1) **BERT** [8] is a large, pre-trained contextual model based on the transformer architecture. We use the *base* version with 12 encoder layers, 12 attention heads and 768-dimensional output representations. The input length is restricted to a maximum of 512 tokens.
(2) **RoBERTa** [36] is another cross-attention model which is architecturally identical to BERT; the only difference of the two models is the pre-training procedure.
(3) **DistilBERT** [54] employs *knowledge distillation* techniques to compress the original BERT model to roughly 40% of its original size, while largely maintaining performance. As a result, DistilBERT is a much smaller parametric ranker with only 6 encoder layers. We choose DistilBERT to study the effect of `RankingSCL` and augmentation on models with low parameterization.

4.1 Experiments Conducted

To answer the above RQ's, we experiment with (a) different types of contextual models – BERT, RoBERTa, DistilBERT, (b) varying dataset sizes – 1K, 2K, 10K, 100K instances for Doc'19 and Doc'20, (c) two ranking losses – `Pointwise RankingSCL` and `Pairwise RankingSCL` d) different data augmentation strategies BM25, GloVe, Sampling and e) different datasets Doc'19, Doc'20, Robust04, SciFact and FiQA. To give an example, the number of models trained on Doc'19 is 1440 (72 best model combinations are chosen for reporting). Given the large number of models it is difficult to report all combination of results and their respective hyperparameters. So we chose to report a selective part of it due to space constraints.

4.2 Batch Creation and Hyperparameters

As mentioned in Section 3.3, we consider the positive query document pairs from the top-k retrieved set and sample an equal number of negative pairs for the original dataset. After that, we use the selector to generate augmented versions of documents. For TREC-DL, we tried with varying amounts of query-document pairs - 1k, 2k, 10k, and 100k. For example, for 1k, we have 500 positive and 500 negative pairs that constitute our original dataset. Further, we add 1k more through the augmentation process. Hence, 1k contains a total of 2k query-document pairs. The same pattern holds for the other three sizes. In Robust04, we consider all the pairs from the training set because it contains fewer queries. *Note that we only augment the training data, the validation and test sets are not augmented.*

Hyperparameters. We have two hyperparameters in our models: the temperature (τ), and the degree of interpolation (λ) as in `RankingSCL` [eq. (6)]. We use the MS MARCO development set to determine the best combination of τ and λ. For Robust04, we use the validation set as shared by Dai and Callan [7], i.e. a small subset of training queries. These parameters are different for different ranking models and augmentation strategies (BM25, GloVe, Sampling). For example, in TREC-DL, BERT ranking model using BM25 data augmentation and `PointwiseRankingSCL` loss objective returns best score on validation set at $\tau = 0.4$ and $\lambda = 0.8$

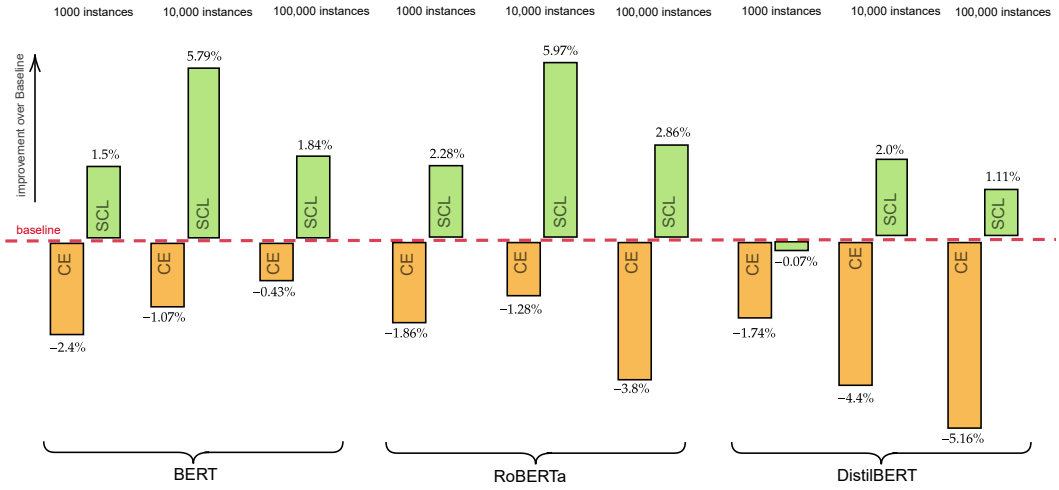


Figure 2: Relative ranking improvement over pointwise baseline ranker trained over non-augmented data. CE in the figure refers to cross entropy (pointwise loss), and SCL refers to pointwise variant of RankingSCL. Dataset: Doc'19, augmentation strategy: BM25 selection.

	Doc'19			Doc'20			Robust04		
	AP	RR	nDCG$_{10}$	AP	RR	nDCG$_{10}$	AP	RR	nDCG$_{10}$
BERT									
Baseline	0.244	0.834	0.592	0.373	0.891	0.547	0.264	0.763	0.506
Sampling	0.253(▲3.8%)	0.886(▲6.3%)	0.617(▲4.3%)	0.391(▲4.8%)	0.941(▲5.6%)	0.594(▲8.6%)*	0.276(▲4.7%)	0.797(▲4.5%)	0.537(▲6%)
BM25	0.258(▲5.8%)	0.924(▲10.8%)	0.617(▲4.3%)	0.378(▲1.3%)	0.944(▲6.0%)	0.562(▲2.8%)	0.273(▲3.1%)	0.793(▲3.9%)	0.533(▲5.3%)
GloVe	0.253(▲4.0%)	0.898(▲7.7%)	0.626(▲5.8%)	0.387(▲3.8%)	0.940(▲5.6%)	0.566(▲3.5%)	0.278(▲5.2%)	0.799(▲4.7%)	0.541(▲6.8%)
RoBERTa									
Baseline	0.243	0.812	0.557	0.307	0.725	0.470	0.205	0.594	0.378
Sampling	0.245(▲1.0%)	0.878(▲4.1%)	0.583(▲4.6%)	0.365(▲18.8%)	0.922(▲27.2%)	0.557(▲18.5%)*	0.257(▲25.8%)	0.746(▲25.5%)	0.496(▲37.4%)
BM25	0.257(▲6.0%)	0.873(▲7.5%)	0.597(▲7.1%)*#	0.362(▲18.1%)	0.922(▲27.2%)	0.548(▲16.7%)*	0.265(▲29.7%)	0.766(▲28.8%)	0.509(▲34.9%)
GloVe	0.259(▲6.8%)	0.863(▲6.3%)	0.596(▲7.0%)*#	0.354(▲15.3%)	0.870(▲20.0%)	0.550(▲17%)*	0.267(▲30.4%)	0.787(▲32.4%)	0.519(▲37.3%)
DistilBERT									
Baseline	0.244	0.843	0.565	0.322	0.849	0.515	0.201	0.614	0.395
Sampling	0.253 (▲4%)	0.883(▲4.7%)	0.591(▲4.6%)#	0.350(▲8.8%)	0.919(▲8.2%)	0.557(▲8.1%)	0.213 (▲6.3%)	0.713(▲16.2%)	0.480(▲21.6%)
BM25	0.248(▲2.0%)	0.909(▲7.8%)	0.573(▲1.3%)#	0.346(▲7.6%)	0.915(▲7.7%)	0.538(▲4.4%)	0.211(▲5.3%)	0.704(▲14.7%)	0.505(▲27.8%)
GloVe	0.250(▲2.5%)	0.872(▲3.8%)	0.583(▲3.1%)	0.338(▲5.1%)	0.907(▲6.8%)	0.505(▼−1.9%)	0.210(▲3.9%)	0.681(▲11.0%)	0.509(▲28.9%)

Table 1: Long document re-ranking results on the Doc'19, Doc'20, and Robust04 datasets. We train each ranker using three different data augmentation techniques on 10K instances. The models are trained using a pointwise variant of the RankingSCL loss function. Statistically significant improvements at a level of 95% and 90% are indicated by $*$ and # respectively [12].

values respectively. In all our experiments we use a batch size of 16. Reporting all hyper-parameter values is not possible owing to the large number of experiments.

5 EXPERIMENTAL RESULTS

We start by first answering the question if existing loss functions are sufficient in delivering performance improvements when considering augmented datasets. Next, we take a detailed look into the effect of the training data size, contextual model type and the augmentation strategy on the ranking performance when using RankingSCL. Finally, we look at the impact of RankingSCL on training contextual ranking models on smaller datasets.

5.1 Augmentation with and without SCL

To answer **RQ I**, we first compare the performance of rankers trained using the standard loss functions in comparison to RankingSCL losses. We conducted several experiments to check the relative improvement of rankers trained with PointwiseRankingSCL and PairwiseRankingSCL loss on the *same augmented datasets* over different training set sizes and augmentation strategies. Figure 2 plots

the relative performance improvement (in terms of MAP) of models trained using data augmentation over a baseline model trained without data augmentation. Note that performance in terms of ranking metrics and augmentation strategies for PointwiseRankingSCL and PairwiseRankingSCL follows a similar pattern and we use Figure 2 as a representative result.

We observe that the Pointwise loss on the augmented datasets in fact performs worse than the baseline non-augmented variant. This is not surprising, since it has been shown in the ML literature [9, 75] that cross-entropy is sensitive to label and data noise.

On the other hand, Figure 2 shows that PointwiseRankingSCL effectively utilizes augmented data to learn better representations. This is reflected in consistent improvements over the baselines. By considering increasing amount of training data, i.e. 1000 to 100,000 instances, we observe that for RoBERTa and DistilBERT more data augmentation can negatively impact ranking performance when RankingSCL loss is not used. This establishes that increasing data augmentation with traditional ranking loss functions is detrimental to ranking performance.

Insight 1. Our first insight is that data augmentation is useful only when a proper loss function is used in conjugation, i.e. PointwiseRankingSCL or PairwiseRankingSCL loss.

5.2 The Impact of augmentation type

We now answer **RQ II** by comparing different data augmentation approaches – *matching-based*, *embeddings-based*, and *sampling-based* augmentation methods. We show the performance of our data augmentation techniques applied to re-ranking models in Table 1 on the three datasets that involve longer documents. This is due to the fact that the likelihood of getting an unrelated piece of text as an augmentation candidate is higher for longer texts.

We see that there are no clear winners. Firstly, matching-based augmented datasets result in consistent performance over all datasets and rankers. Secondly, Sampling augmentation already helps in improving ranking performance with the PointwiseRankingSCL loss. Note that an artifact of the the Doc'19 and Doc'20 datasets is that most of the queries have exactly one relevant document even if there are multiple relevant documents due to the data collection strategy, i.e. both the datasets have incomplete labels. Arguably, just having one positive document-per-query results in augmented instances being parts of the original relevant document and even a random selection having a high likelihood of being relevant.

The Robust04 dataset, unlike Doc'19 and Doc'20, has multiple documents-per-query with positive relevance labels. Having multiple relevant documents per query results in multiple positive-document pairs without resorting to augmentation.

Interestingly, we see that Sampling is as competitive as GloVe and BM25, even for the case where the labelling is complete, i.e. in case of the Robust04 dataset. The conclusion that we draw from this experiment is that for the common low-data scenarios of smaller instances and incomplete labels simple augmentation approaches like Sampling already provide reasonable improvements when using PointwiseRankingSCL as the optimization objective. Experiments with Pairwise RankingSCL have similar trend to PointwiseRankingSCL as described above.

Insight 2. We find that choice of simple data augmentation strategies do not have a big impact on the ranking performance when using RankingSCL (Pairwise or Pointwise).

5.3 The Impact of Data Augmentation

To answer **RQ III**, we experiment with (a) different types of contextual models – BERT, RoBERTa, DistilBERT, (b) varying dataset sizes – {1000, 2000, 10000, 100000} instances, and (c) two ranking losses – PointwiseRankingSCL and PairwiseRankingSCL with (d) different data augmentation strategies BM25, GloVe, Sampling – on a ranking dataset Doc'19. In Table 2 we present the results of this experiment where we choose the BM25 augmentation strategy for our three contextual rankers. We compare the relative ranking improvements of using data augmentation against a baseline that is trained over a non-augmented dataset. *Note that we refer to the fine-tuned model on the non-augmented dataset as the baseline model.* Specifically, for a non-augmented dataset (say 1000,10000, or 100000 instances) an augmented dataset is constructed as described in the previous section (see Section 4.2).

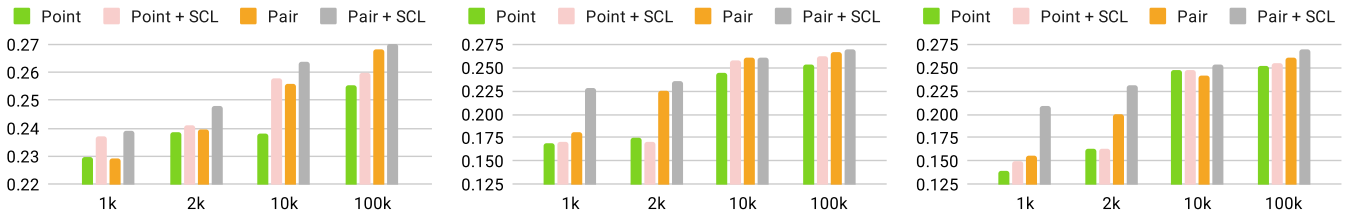
The reported results measure the ranking performance when the contextual models are fine-tuned on the augmented dataset using the RankingSCL loss. The corresponding values in the parentheses measure the increase or decrease in performance compared to the corresponding baseline model (as described earlier).

In general, we clearly observe that the ranking performance increases in a majority of cases when using data augmentation using the RankingSCL loss function. Firstly, data augmentation is particularly useful for smaller instances, i.e. dataset of sizes 10, 000 instances or less. Specifically, we see improvements of up to 12.9% in reciprocal rank when using BERT ranker (with augmented data) over the baseline BERT ranker (without augmentation) in the Doc'19 dataset using PointwiseRankingSCL. To put the query workload into context, note that the Doc'19 dataset contains 300, 000 training instances. The superior performance using augmentation for smaller datasets can be clearly attributed to the small number of training queries, which is insufficient for training over-parameterized contextual rankers without any augmentation. However, when the training set increases to 100, 000 instances, i.e. closer to the full size of the dataset, we see diminishing marginal utility of using data augmentation. Similar results have also been reported in other studies in NLP [15] while fine tuning contextual models for other language tasks.

Secondly, we observe that the improvements are much larger when using the DistilBERT ranker instead of BERT or RoBERTa especially in the PairwiseRankingSCL setting. We present a grouped bar plot to clearly show the trends in Figure 3. To start off, the DistilBERT ranker performs poorly in the low-data regime when using both the baseline non-augmented setting as well as in the case of augmentation. However, when using data augmentation for slightly larger datasets, the performance improvement over the baseline is considerable. Especially, for the 10, 000 instance dataset, we see an improvement of around 4% in reciprocal rank and 7.7% in NDCG (also statistically significant). More striking is the performance improvement in the PairwiseRankingSCL setting, where the NDCG improvements are about 60% for the 1000 instance dataset. This shows that for models that require large amounts of

	Pointwise			Pairwise		
Ranking Models	AP	RR	$nDCG_{10}$	AP	RR	$nDCG_{10}$
BERT						
1k	0.237(▲1.5%)	0.868(▲3.7%)	0.551(▲3.1%)	0.239(▲4.3%)	0.851(▲6.2%)	0.576(▲5.7%)
2k	0.241(▲1.9%)	0.916(▲12.9%)	0.592(▲5.2%)	0.248(▲3.6%)	0.892(▼ − 0.4%)	0.603(▲1.5%) *
10k	0.258(▲5.8%)	0.924(▲10.8%)	0.617(▲4.3%)	0.264(▲3.1%)	0.926(▲3.9%)	0.627(▲7.5%)*
100k	0.260(▲1.8%)	0.942(▲4.3%)	0.653(▲6.3%)	0.270(▲0.6%)	0.959(▲2.7%)	0.666(▲3.4%)
RoBERTa						
1k	0.170(▲2.3%)	0.697(▲25.9%)	0.319(▲7.4%)	0.228(▲25.9%)	0.803(▲15.7%)	0.533(▲59.8%)
2k	0.171(▲1%)	0.670(▲12.4%)	0.322(▲9.5%)	0.236(▲4.4%)	0.871(▲4.7%)	0.587(▲7.4%)
10k	0.257(▲6%)	0.873(▲7.5%)	0.597(▲7.1%)*#	0.261(▲3.5%)	0.914(▲3.8%)	0.633(▲3.5%)*
100k	0.263(▲2.9%)	0.946(▲4.7%)	0.646(▲11.7%)	0.270(▲1.2%)	0.955(▲1.4%)	0.6667(▲0.3%)
DistilBERT						
1k	0.150(▲0%)	0.553(▲14.3%)	0.239(▲9.2%)	0.208(▲33.9%)	0.802(▲35.8%)	0.471(▲61.4%)
2k	0.164(▲2.3%)	0.589(▲0.6%)	0.304(▲9.2%)	0.231(▲15%)	0.862(▲13.1%)	0.526(▲19.4%)
10k	0.248(▲2.0%)	0.909(▲7.8%)	0.573(▲1.3%) #	0.253(▲5.1%)	0.893(▲3.9%)	0.613(▲7.7%)*
100k	0.255(▲1.1%)	0.942(▲3.1%)	0.641(▲5.7%)	0.270(▲3.3%)	0.927(▲2.9%)	0.645(▲1.5%)*

Table 2: Document re-ranking results on the Doc'19 datasets for `Pointwise` and `Pairwise` with `RankingSCL` with data augmentation using BM25 selection strategy. 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 [12].



(a) BERT **(b) RoBERTa** **(c) DistilBERT**

Figure 3: Comparing MAP score of `Pointwise` and `Pairwise` with corresponding `RankingSCL` variant for different training instance sizes and models on TREC-DL.

training data to perform well, such as DistilBERT, our training data augmentation techniques turn out to be particularly useful, achieving large improvements over the baseline models and matching or even improving on the other models, despite the rather poor baseline performance. A similar trend is also seen in RoBERTa for the smallest dataset with performance improvements are considerably large, e.g. more than 25% MAP and more than 50% in NDCG for pairwise learning.

Insight 3. `RankingSCL` has the highest marginal utility when the dataset sizes are small. The utility diminishes with increasing dataset size.

5.4 The Effect of SCL on Small Datasets

A natural question to ask from the last experiment is whether the performance improvements on smaller subsets of TREC-DL can be replicated on other diverse ranking datasets that have small training data sizes. In the next experiment, we considered datasets corresponding to three diverse tasks – a question answering task, a fact verification task, and a document ranking task – to finally evaluate our claim about the high utility of `RankingSCL` over smaller text ranking datasets. Both the question-answering task (FiQA) and the fact-verification task (SciFact) rank passages given a query that intends to maximize the likelihood of finding the right evidence document at the top part of the ranking.

Since the datasets used in this experiment are much smaller in comparison to the previously used web datasets, we trained multiple contextual models with different initialization and present the average ranking performance results in Table 3. Note that, variance of the ranking metrics in fine-tuning over smaller datasets using over-parameterized contextual models is a known phenomena [9, 43, 52, 63]. It is clear that, although there is reasonable variance due to model initialization, `RankingSCL` losses result in improved ranking performance, sometimes by a large margin. To

	Robust04			SciFact			FiQA		
	AP	RR	$nDCG_{10}$	AP	RR	$nDCG_{10}$	AP	RR	$nDCG_{10}$
BERT									
Base-pointwise	0.264	0.763	0.506	0.312	0.32	0.383	0.140	0.221	0.187
Pointwise	0.276(▲4.7%)	0.797(▲4.5%)	0.537(▲6%)	0.434(▲39%)	0.448(▲40%)	0.466(▲22%)	0.141(▲0.8%)	0.221(▲3.4%)	0.187(▼−1.5%)
Base-pairwise	0.195	0.599	0.382	0.454	0.466	0.504	0.136	0.205	0.174
Pairwise	0.200(▲2.7%)	0.601(▲0.4%)	0.388(▲1.6%)	0.562(▲33.6%)	0.575(▲23.5%)	0.616(▲29%)*	0.221(▲63%)	0.343(▲67%)	0.277(▲59%)
RoBERTa									
Base-pointwise	0.205	0.594	0.3776	0.615	0.626	0.668	0.113	0.173	0.146
Pointwise	0.258(▲26%)	0.746(▲25.5%)	0.496(▲37.4%)	0.638(▲3.7%)	0.649(▲3.7%)	0.687(▲2.8%)*	0.240(▲112%)	0.365(▲111%)	0.300(▲108%)
Base-pairwise	0.250	0.762	0.460	0.641	0.652	0.685	0.255	0.382	0.316
Pairwise	0.277(▲13.9%)	0.529(▲11.65%)	0.766(▲6.1%)*	0.668(▲4.2%)	0.681(▲4.5%)	0.712(▲3.8%)*	0.274(▲7.6%)	0.412(▲7.9%)	0.339(▲7.4%)*
DistilBERT									
Base-pointwise	0.201	0.614	0.395	0.551	0.567	0.595	0.111	0.188	0.132
Pointwise	0.258(▲28.5%)	0.688(▲12.1%)	0.480(▲21.6%)	0.532(▼−3.5%)	0.558(▼−3.3%)	0.574(▼−3.6%)	0.170(▲54%)	0.269(▲43%)	0.216(▲64%)*
Base-pairwise	0.186	0.372	0.576	0.538	0.554	0.577	0.235	0.362	0.288
Pairwise	0.182(▼−1.9%)	0.617(▲7%)	0.375(▲0.7%)*	0.558(▲3.8%)	0.573(▲3.4%)	0.599(▲3.8%)	0.238(▲1.2%)	0.366(▲1.2%)	0.319(▲10.8%)*

Table 3: Document re-ranking results on the Robust04, SciFact and FiQA datasets. We train each ranker using Sampling data augmentation techniques on different datasets. The models are trained using a linear interpolation of `Pointwise` (`Pointwise` and `RankingSCL`) and `Pairwise` (`Pairwise` and `RankingSCL`) loss functions. Values in brackets are percentage improvement from baseline. Statistically significant improvements at a level of 95% and 90% are indicated by $*$ and # respectively [12].

avoid further variance due to the training process and small test set sizes we report average ranking scores over *five runs* as mentioned in [43]. We observe that both `PointwiseRankingSCL` and `PairwiseRankingSCL` result in consistent performance gains for SciFact and FiQA. Impressive improvements are observed when training BERT with `Pairwise RankingSCL` loss for theFiQA. This is primarily because the baseline is ineffective to train a reasonable passage ranking model. In general, `Pairwise RankingSCL` outperform `Pointwise` variants except in Robust04 datasets. One might argue that this is due to the large variance of the baseline when training on smaller datasets.

Insight 4. `RankingSCL` results in large performance gains on a variety of small ranking datasets.

5.5 Threats to Validity

There are some threats to validity of our work that we detail in the following. Firstly, we put into perspective the actual gains or improvements from our experiments by analyzing if the improvements are statistically significant. We observe an important pattern that is worth discussing. The average improvement in the FiQA dataset using RoBERTa when considering `PointwiseRankingSCL` losses is above 100% but interestingly the improvements do not turn out to be statistically significant. On the other hand, even if the average improvements in the `Pairwise RankingSCL` are lesser than `Pointwise RankingSCL` the improvements turn out to be statistically significant. On closer examination, it turns out that there is a large variance in the ranking metrics for the `RankingSCL` model when trained in the pointwise regime, i.e., MAP value of 0.24 ± 0.11. In contrast, the MAP values (with variance) for the baseline model over the test set queries is 0.11 ± 0.005 showing the small variance in scores. This means a small set of queries starkly outperforming the baseline `Pointwise` model while there is little difference between a large fraction of queries. We see a similar pattern in the BERT model trained on pairwise `RankingSCL` loss for the FiQA and Robust04.

6 DISCUSSION AND CONCLUSION

We make several important observations from our results. Firstly, we find that using augmented training data with existing `Pointwise` or `Pairwise` objectives does not result in performance improvements. In many scenarios, the ranking performance decreases when using data augmentation with existing loss functions justifying existing work in the vision and language community that show the fragility of cross-entropy losses when using noisy labels [15, 23]. Instead we clearly show that `RankingSCL` improves the ranking performance when using data augmentation in a variety of datasets. Secondly, we find that this type of data augmentation surprisingly has little to no effect on the ranking performance. This suggests that using cheaper data augmentation schemes is already useful in simplifying the design decisions to be considered when using the `RankingSCL` loss. Finally, we observe that data augmentation is useful in improving the ranking performance specifically when training data sets are small and the marginal utility of data augmentation reduces with increasing data sizes with the maximum improvements being observed for low data settings. We also observe that different inductive biases (contextual models) react differently to `RankingSCL`, with RoBERTa-based ranker showing improvements in ranking metrics up to $> 50\%$ for smaller datasets over its non-augmented counterparts.

ACKNOWLEDGMENTS

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).

REFERENCES

[1] Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. ReQA: An evaluation for end-to-end answer retrieval models. *arXiv preprint arXiv:1907.04780* (2019).

[2] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932* (2020).

[3] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2020. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086* (2020).

[4] 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. *arXiv preprint arXiv:2106.02293* (2021).

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

[6] 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.

[7] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *ACM SIGIR'19*. 985–988.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805

[9] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305* (2020).

[10] Jeff Donahue and Karen Simonyan. 2019. Large scale adversarial representation learning. *arXiv preprint arXiv:1907.02544* (2019).

[11] Xin Luna Dong, Yaxin Zhu, Zuohui Fu, Dongkuan Xu, and Gerard de Melo. 2021. Data Augmentation with Adversarial Training for Cross-Lingual NLI. 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)*. 5158–5167.

[12] Luke Gallagher. 2019. Pairwise t-test on TREC Run Files. https://github.com/lgrz/pairwise-ttest/.

[13] 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 *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE, 1147–1158.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).

[15] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403* (2020).

[16] 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.

[17] 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. *arXiv preprint arXiv:1808.06670* (2018).

[18] Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. 2020. Local Self-Attention over Long Text for Efficient Document Retrieval. *arXiv preprint arXiv:2005.04908* (2020).

[19] Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. 2020. Interpretable & time-budget-constrained contextualization for re-ranking. *arXiv preprint arXiv:2002.01854* (2020).

[20] 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.

[21] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[22] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. arXiv:2004.12832

[23] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 18661–18673.

[24] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245* (2020).

[25] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).

[26] Jurek Leonhardt, Koustav Rudra, and Avishek Anand. 2021. Learnt Sparsity for Effective and Interpretable Document Ranking. *arXiv preprint arXiv:2106.12460* (2021).

[27] Jurek Leonhardt, Koustav Rudra, Megha Khosla, Abhijit Anand, and Avishek Anand. 2021. Fast Forward Indexes for Efficient Document Ranking. *arXiv preprint arXiv:2110.06051* (2021).

[28] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage Representation Aggregation for Document Reranking. *arXiv preprint arXiv:2008.09093* (2020).

[29] Hao Li, Xiaopeng Zhang, Qi Tian, and Hongkai Xiong. 2020. Attribute mix: Semantic data augmentation for fine grained recognition. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 243–246.

[30] 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.

[31] 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.

[32] 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 preprint arXiv:2008.01969* (2020).

[33] Jimmy Lin. 2019. The Neural Hype and Comparisons Against Weak Baselines. In *ACM SIGIR Forum*, Vol. 52. ACM, 40–51.

[34] Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq R. Joty, Luo Si, and Chunyan Miao. 2021. MulDA: A Multilingual Data Augmentation Framework for Low-Resource Cross-Lingual NER. In *ACL/IJCNLP*. 5834–5846. https://doi.org/10.18653/v1/2021.acl-long.453

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

[36] 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 preprint arXiv:1907.11692* (2019).

[37] Shayne Longpre, Yu Wang, and Christopher DuBois. 2020. How Effective is Task-Agnostic Data Augmentation for Pretrained Transformers? *arXiv preprint arXiv:2010.01764* (2020).

[38] Xinghua Lu, Bin Zheng, Atulya Velivelli, and ChengXiang Zhai. 2006. Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association* 13, 5 (2006), 526–535.

[39] Canjie Luo, Yuanzhi Zhu, Lianwen Jin, and Yongpan Wang. 2020. Learn to augment: Joint data augmentation and network optimization for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13746–13755.

[40] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Contextualized Word Representations for Document Re-Ranking. *arXiv preprint arXiv:1904.07094* (2019).

[41] Sakorn Mekruksavanich and Anuchit Jitpattanakul. 2021. Convolutional neural network and data augmentation for behavioral-based biometric user identification. In *ICT Systems and Sustainability*. Springer, 753–761.

[42] John X 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. *arXiv preprint arXiv:2005.05909* (2020).

[43] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884* (2020).

[44] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[45] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7689–7693.

[46] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). http://arxiv.org/abs/1901.04085

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

[48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[49] Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data Augmentation for Abstractive Query-Focused Multi-Document Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13666–13674.

[50] Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020. Data augmentation for spoken language understanding via pretrained models. *arXiv e-prints* (2020), arXiv–2004.

[51] Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. 2020. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862* (2020).

[52] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics* 8 (2020), 842–866.

[53] 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.

[54] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[55] 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.

[56] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text Data Augmentation for Deep Learning. *Journal of big Data* 8, 1 (2021), 1–34.

[57] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*. 1857–1865.

[58] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, Vol. 2. 2–6.

[59] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. 2020. Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks. *arXiv preprint arXiv:2010.02394* (2020).

[60] Hoang Van, Vikas Yadav, and Mihai Surdeanu. 2021. Cheap and Good? Simple and Effective Data Augmentation for Low Resource Machine Reading. *arXiv preprint arXiv:2106.04134* (2021).

[61] Ellen M. Voorhees. 2005. Overview of the TREC 2004 robust track. In *TREC, volume Special Publication 500-261*. National Institute of Standards and Technology (NIST).

[62] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *EMNLP*.

[63] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[64] 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.

[65] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[66] Ju Xu, Mengzhang Li, and Zhanxing Zhu. 2020. Automatic data augmentation for 3D medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 378–387.

[67] Rintaro Yanagi, Ren Togo, Takahiro Ogawa, and Miki Haseyama. 2020. Image Retrieval with Data Augmentation of Sentence Labels Based on Paraphrasing. In *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*. IEEE, 1–2.

[68] 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.

[69] 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 preprint arXiv:1904.06652* (2019).

[70] Yinfei Yang, Ning Jin, Kuo Lin, Mandy Guo, and Daniel Cer. 2020. Neural Retrieval for Question Answering with Cross-Attention Supervised Data Augmentation. *arXiv preprint arXiv:2009.13815* (2020).

[71] 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.

[72] 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.

[73] Shujuan Yu, Jie Yang, Danlei Liu, Runqi Li, Yun Zhang, and Shengmei Zhao. 2019. Hierarchical data augmentation and the application in text classification. *IEEE Access* 7 (2019), 185476–185485.

[74] Yi Zeng, Han Qiu, Gerard Memmi, and Meikang Qiu. 2020. A data augmentation-based defense method against adversarial attacks in neural networks. In *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 274–289.

[75] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015), 649–657.

[76] Zhilu Zhang and Mert R Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*.

[77] 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.

[78] Qingqing Zhu, Xiwei Wang, Chen Chen, and Junfei Liu. 2020. Data Augmentation for Retrieval-and Generation-Based Dialog Systems. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, 1716–1720.

[79] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. 2020. Learning data augmentation strategies for object detection. In *European Conference on Computer Vision*. Springer, 566–583.