Contrastive Fine-tuning Improves Robustness for Neural Rankers

Xiaofei Ma **Cicero Nogueira dos Santos** **Andrew O. Arnold**
AWS AI Labs
{xiaofeim, cicnog, anarnld}@amazon.com

Abstract

The performance of state-of-the-art neural rankers can deteriorate substantially when exposed to noisy inputs or applied to a new domain. In this paper, we present a novel method for fine-tuning neural rankers that can significantly improve their robustness to out-of-domain data and query perturbations. Specifically, a contrastive loss that compares data points in the representation space is combined with the standard ranking loss during fine-tuning. We use relevance labels to denote similar/dissimilar pairs, which allows the model to learn the underlying matching semantics across different query-document pairs and leads to improved robustness. In experiments with four passage ranking datasets, the proposed contrastive fine-tuning method obtains improvements on robustness to query reformulations, noise perturbations, and zero-shot transfer for both BERT and BART-based rankers. Additionally, our experiments show that contrastive fine-tuning outperforms data augmentation for robustifying neural rankers.

1 Introduction

Recent advances in neural language modeling have shifted the paradigm of natural language processing (NLP) towards a two-stage process: pre-training on a large amount of data with self-supervised tasks followed by fine-tuning on the target datasets with task-specific loss functions. Current state-of-the-art neural rankers for information retrieval fine-tune pre-trained language models using ranking losses on datasets containing examples of positive and negative query-document pairs. While usually achieving good performance on in-domain test sets, neural rankers trained on large datasets can still exhibit poor transferability when tested in new domains, and suffer from robustness problems when exposed to various types of perturbations. For example, a neural ranker trained on a dataset with mostly natural language queries can perform badly when tested on keyword queries which are very common in information retrieval (Bhatia et al., 2020).

A considerable number of previous works have focused on domain adaptation to improve model 's overall transferability. While domain adaptation approaches can help to address the out-of-domain robustness problem (Pan and Yang, 2010; Zhang et al., 2019; Ma et al., 2019), they rely on the availability of either labeled data or at least a target corpus which is usually not available at training time for a neural ranking model deployed in the wild.

The vulnerability of deep NLP models to various forms of adversarial attacks such as word-importance-based replacement (Jin et al., 2020), human-curated minimal perturbations (Khashabi et al., 2020), misspelling (Sun et al., 2020), grammatical errors (Yin et al., 2020), rule-based perturbations (Si et al., 2020; Ribeiro et al., 2018) is well-documented in the literature (Emma Zhang et al., 2019). While various methods have been proposed to remediate model robustness issues in NLP, most of them are either task-specific (Shah et al., 2019; Zhou et al., 2020; Gan and Ng, 2020; Wang and Bansal, 2018), requiring auxiliary tasks (Zhou et al.), or relying on data augmentation (Min et al., 2020; Kaushik et al., 2019; Cheng et al., 2020; Wei and Zou, 2019) which highly depends on the quality and diversity of the perturbed data.

An alternative strategy for optimizing machine learning models that has the potential to improve both out-of-domain generalization and robustness is contrastive learning. Representations obtained under contrastive self-supervised settings have demonstrated improved robustness to out-of-domain distributions and image corruptions in computer vision tasks (Hendrycks et al., 2019; Radford et al., 2021). In contrastive learning, representa-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 570–582

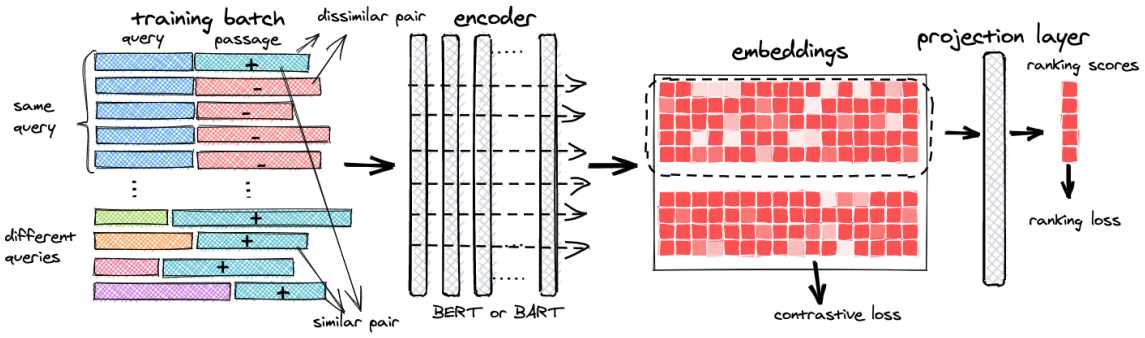

Figure 1: Contrastive fine-tuning for neural rankers. During fine-tuning, a batch of positive and negative samples from different queries is fed into a neural encoder. The embeddings of query-document pairs from the same query are used to generate ranking scores, which are employed to compute the ranking loss. In parallel, the embeddings of all pairs are used to compute the contrastive loss.

tions are learned by comparing among similar and dissimilar samples (Le-Khac et al., 2020; Khosla et al., 2020; Van Den Oord et al., 2018; Hjelm et al., 2018). This is different from discriminative learning, where models learn a mapping of input samples to labels, and generative learning, where models reconstruct input samples. While several works have investigated contrastive learning for sentence classification (Gunel et al., 2020), sentence representation learning (Wu et al., 2020), and multi-modal representation learning (Radford et al., 2021) under either self-supervised or supervised settings, their potential for improving the robustness of neural rankers has not been explored yet.

In this paper, we propose a novel contrastive learning approach to fine-tune neural rankers and investigate its benefits for improving model robustness. We focus on rankers that use single-tower architectures and are normally trained by optimizing a ranking loss that compares scores of positive and negative query-document pairs involving the same query. We propose to additionally use a contrastive loss that compares the distance between the representation of positive and negative pairs involving distinct queries (i.e, representations of positive pairs should be close in the latent space and distant to the representation of negative pairs, and vice-versa). The goal of using this contrastive loss in addition to the ranking loss is to stimulate the model to learn the underlying matching semantics across different query-document pairs, which can potentially lead to improved robustness.

Our main contributions are as follows:

- We propose to combine contrastive loss with ranking loss during fine-tuning of neural ranking models and investigate its impact in improving model robustness and generalization.
- Our experimental results using two language model-based neural rankers (BERT and BART) on four different datasets indicate that our proposed method improves upon standard ranking loss in zero-shot transfer across domains, leading to an increase of up to 9 absolute points in Mean Average Precision (MAP).
- We develop new datasets for evaluating the robustness of neural rankers. The datasets are based on WikiQA test set (Yang et al., 2015) and were created semi-automatically. We plan to release these datasets upon acceptance of the paper.
- We show that contrastive fine-tuned rankers are robust to 1) different types of query reformulations commonly seen in information retrieval (headline, paraphrase, and change of voice); and 2) query perturbations such as adding/removing punctuations, typos, and contractions/expansions.

2 Contrastive Representation Learning for Neural Ranking

In neural ranking models, given an input query q and a set of candidate documents $\{d_0, d_1, ..., d_n\}$, a neural network h is used to create vector representations $\{h(q, d_1), ..., h(q, d_n)\}$, which are given to a function $s : \overrightarrow{x} \rightarrow \mathbb{R}$, that computes a score for each query-document pair, $\{s(h(q, d_1)), ..., s(h(q, d_n))\}$. Normally s performs a simple linear projection of the input embedding, and the training of neural ranking models consists in optimizing a ranking loss that tries to

enforce $s(h(q, d^+)) > s(h(q, d^-))$ for each training query q, where d^+ is a positive document for q while d^- is a negative one (See top part of Fig. 1).

We propose to augment the training of neural rankers with the use of contrastive representation learning. While ranking-based methods compute the loss with respect to the predicted scores, contrastive losses measure the distance/similarity between similar and dissimilar samples in the representation space. In our case, the key idea consists in using a loss that compares the distance between the representation of query-document pairs, and enforces that positive pairs are close together in the latent space while being far apart from negative pairs, i.e., $D(h(q, d^+), h(q', d'^+)) < D(h(q, d^+), h(q, d^-))$, where q' is either a variation of q or a completely different query, and d'^+ is a positive document for q'. Figure 1 illustrates our proposed approach, which is detailed in the remainder of this section.

2.1 Ranking Loss

Popular ranking losses include 1) the *pairwise ranking loss*, in which the relevance information is given in the form of preferences between pairs of candidates, and 2) the *listwise ranking loss* which directly optimizes a rank-based metric. In this work, we experiment with two pairwise ranking losses. The first one is the *standard hinge loss* (SHL) defined on a triplet (q, d^+, d^-) as follows:

$$\mathcal{L}_{SHL}(q, d^+, d^-; \theta) = \max\{0, \lambda - s(h(q, d^+); \theta) + s(h(q, d^-); \theta)\} \quad (1)$$

The other is a *modified hinge loss* (MHL) function defined as:

$$\mathcal{L}_{MHL}(q, d^+, \{d_i^-\}; \theta) = \max\{0, \lambda - s(h(q, d^+); \theta) + \max_i\{s(h(q, d_i^-); \theta)\}\} \quad (2)$$

where q is a query, λ is the margin of the hinge loss, d^+ refer to the positive document. d^- and $\{d_i^-\}$ refer to a negative document and the list of negative documents of the query q within the same batch, respectively. θ includes the set of parameters of the network h and the projection layer in s. Based on preliminary experiments, our modified ranking hinge loss generally performs better than the standard pairwise ranking hinge loss. Note that MHL loss has been used in previous work on passage ranking (dos Santos et al., 2016).

2.2 Contrastive Loss

For contrastive learning of representations, we employ the conceptually simple but widely adopted *triplet margin loss* (**TML**) (Weinberger et al.; Chechik et al., 2010), which has the following form:

$$\mathcal{L}_{TML}(a, k^+, k^-; \theta) = \max\{0, m + D(a, k^+; \theta) - D(a, k^-; \theta)\} \quad (3)$$

where a is the anchor point, k^+ and k^- are the similar and dissimilar samples with respect to the anchor point a. m is the margin of the TML loss. In our neural ranking setting, an anchor point is the representation of a query-document pair. We use Euclidean or L2 distance D in our experiment. The contrastive loss can be applied to the representations from a variety of encoders $h(\cdot) \in R^d$. In this work, we explore contrastive fine-tuning for both BERT (Devlin et al., 2018) and BART (Lewis et al.) models.

The key to effective contrastive learning is to design the *notion of similarity* such that positive pairs may be very different in the input space yet semantically related. In this work, we leverage the relevance label in the training data and consider as similar positive pairs (q_i, d_i^+) and (q_j, d_j^+) from different queries i and j in the same batch (as illustrated in Fig. 1). Our intuition is that, by enforcing that positive pairs are close together in the embedding space and distant from negative pairs, we make the scoring task easier. Additionally, it allows the model to learn the underlying matching semantics across different query-document pairs, which leads to improved robustness. We additionally conduct a brief experiment in Sec. 5.4 where we use paraphrases of the original query to generate similar pairs.

2.3 Combined Loss

Our final loss is a weighted average of the ranking loss $\mathcal{L}_{ranking}$ and the contrastive loss $\mathcal{L}_{contrastive}$:

$$\mathcal{L} = w_1 \cdot \mathcal{L}_{ranking} + w_2 \cdot \mathcal{L}_{contrastive} \quad (4)$$

The weights w_1 and w_2 are hyper-parameters that need to be determined. Our main experiments use a simple but effective combination method which consists in given equal weights to the ranking loss and contrastive loss.

3 Related Work

Our work is related to the recent body of works that demonstrate contrastive and self-supervised approaches can improve model robustness and generalization. Hendrycks et al. (2019) have shown that self-supervision increases image classifier's robustness to adversarial examples, label corruption, and common input corruptions. Radford et al. (2021) have demonstrated that multi-modal contrastive learning can significantly improve the robustness of image classifiers to distribution shift.

In the NLP space, some recent works on sentence-level contrastive representation learning have shown its potential to improve robustness for classification (Gunel et al., 2020) and semantic text similarity tasks (Wu et al., 2020). There are two main distinctions between our work and these two papers: 1) they focus on classification and text similarity tasks, while we focus on ranking; and 2) while they rely on data augmentation approaches to define the notion of similarity, our approach mainly relies on document relevance information which is already present in the training data.

Our work is also related to recent work on neural retrieval that focus on hard negative mining to improve model performance (Gillick et al., 2019; Xiong et al., 2020; Karpukhin et al., 2020; Lu et al., 2020). The main differences between our work and this line of research are: 1) while we leverage relevance information across different queries to create a notion of similarity, the focus on those papers are on finding hard negatives for each individual query in order to improve training efficiency. Hard negative mining can actually be used together with our method, as we show in Sec. 6.2. 2) we focus in reranking models, which use single-tower model that create a single representation for a query-document pair. In contrast, neural retrieval models create separate representations for query and document.

4 Experimental Setup

In this section, we describe the details of our experimental setup.

4.1 Passage Ranking Datasets

We test our method on four publicly available passage ranking/answer selection datasets that vary in size and domain. Passage ranking is an important task in information retrieval. It is often used to retrieve relevant content for open-domain question-answering systems (Wang et al., 2020).

WikiQA (Yang et al., 2015) is a dataset of question and sentence pairs, collected and annotated for research on open-domain question answering. The questions are factoid and selected from Bing query logs. The answers are in the summary section of a linked Wikipedia page. The candidates are retrieved using Bing.

WikiPassageQA (Cohen et al., 2018) is a benchmark collection for the research on non-factoid answer passage retrieval. The queries are created from Amazon Mechanical Turk over the top 863 Wikipedia documents from the Open Wikipedia Ranking.

InsuranceQA (Feng et al., 2016) The question and answer pairs from this dataset are collected from the internet in the insurance domain. Each question has an answer pool of 500 candidates retrieved using SOLR.

YahooQA (Tay et al., 2017) contains questions and answers from Yahoo! Answers website. The dataset is a subset of the Yahoo! Answers corpus from a 10/25/2007 dump. The questions are selected for their linguistic properties. For example, they all start with how $\{to \mid do \mid did \mid does \mid can \mid would \mid could \mid should\}$.

The statistics of the datasets are presented in Table 1. All four datasets provide validation sets, which have size similar to the respective test sets.

Dataset	Domain	Train: #Q (#P/Q)	Test: #Q (#P/Q)
WikiQA	Wikipedia	873 (9)	243 (9)
WikipassageQA	Wikipedia	3,332 (58.3)	416 (57.6)
InsuranceQA	insurance	12,889 (500)	2,000 (500)
YahooQA	community	50,112 (5)	6,283 (5)

Table 1: Dataset statistics. #Q stands for *number of questions* and #P/Q is the *average number of passages per question*

4.2 Datasets for Robustness Assessment

In order to assess the robustness of our models to different types of query reformulations and query perturbations, we built robustness test datasets based on the original WikiQA test set.

We assessed query perturbations by leveraging CheckList (Ribeiro et al., 2020) to construct three types of popular perturbations: *adding/removing punctuation*, *introducing typos* and *changing of contraction form*. For each query in WikiQA test set, we produce three new versions of the query, one for each perturbation type.

We assessed robustness to three types of query reformulations: *paraphrase*, *headline* and *change of voice*. We semi-automatically created the datasets

for query reformulations in two steps: (1) a pre-trained T5-base model (Raffel et al., 2019) is fine-tuned on a combination of large public paraphrase datasets (Quora [1] and PAWS (Zhang et al., 2019)) and human-curated query reformulations. The human-curated reformulations are based on the queries from the SQuAD 1.1 official dev set [2]. For each query in SQuAD 1.1 dev set, the annotators are asked to generate three new versions of the query (one for each reformulation type). During fine-tuning and inference, we use control codes to instruct T5 on the type of reformulation to be generated. Note that this T5 model can be also used for the purpose of data augmentation, as shown in Sec. 5.4. (2) each query in WikiQA test set is processed by the fine-tuned T5 and three reformulations of the query are generated. All generated queries are post-processed in order to ensure it is grammatically correct and semantically equivalent to the original query. To ensure reliable evaluation, we did a round of human annotations to filter out low-quality generations. Examples of query reformulations are presented in Table 2.

We evaluate the lexical diversity of generated query reformulations by computing the BLEU scores between the original query and the reformulated query. The results of comparing four different generation methods are presented in Figure 2. Our T5 generated queries overall exhibit higher diversity than human generated and back translation generated paraphrases. Note that the lower the BLEU score the higher the diversity.

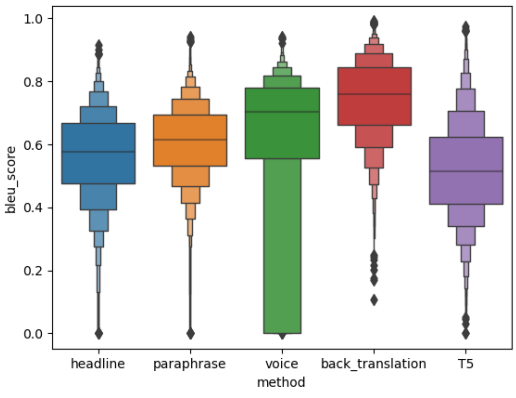

Figure 2: Comparison of BLEU scores between original query and reformulations generated by human annotation, back translation and fine-tuned T5 model.

[1]https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[2]https://rajpurkar.github.io/SQuAD-explorer/

Query	*What fueled the economy of early Vancouver?*
Headline	Factors that fuel Vancouver's economy
Paraphrase	What were some factors that stimulated the early Vancouver's economy?
Chg of Voice	The economy of early Vancouver was fueled by what?
Query	*How was the enlightenment shaped by science of the time period?*
Headline	type of science shaping enlightenment
Paraphrase	How was science of the time influenced the Enlightenment?
Chg of Voice	How did science of the time frame shape the Enlightenment?
Query	*How did South America gain independence from Spain and Portugal?*
Headline	Process of independence of South America from Spain and Portugal
Paraphrase	How did South America come to independence from Spain and Portugal?
Chg of Voice	How was independence from Spain and Portugal gained by South America?

Table 2: Examples of T5 generated styled paraphrases

4.3 Neural Ranker Training

We train neural rankers by fine-tuning two pre-trained language models: BERT and BART. For fine-tuning BERT, we use BERT-base model (12 layers, 110M parameters) from Huggingface's transformer codebase (Wolf et al., 2019). Similar to the setup of sentence pair classification task in (Devlin et al., 2018), we concatenate the query sentence and the candidate passage together as a single input to the BERT encoder. We compute both the contrastive loss and ranking scores based on the [CLS] token embedding of the final hidden layer. For BART model fine-tuning, we use a BART-base model (6 layers encoder, 6 layer decoder, 139M parameters). We adopt the setting of BART for classification task in (**?**). The concatenation of query text and passage text is fed into both the encoder and decoder and the last layer's hidden state of the end decoder token is fed into a linear scorer. Similar to the [CLS] token in BERT, the embedding of the end token from the decoder is used as the representation of the complete input. For training with SHL, we sample triplets (q, d^+, d^-) from different queries to form a single batch. For MHL training, a single batch consists of a positive passage d^+ and a list of negative passages $\{d_i^-\}$ from the same query q. We leverage the toolkit developed by Musgrave et al. (2020) for contrastive loss calculation, and fine-tune the models for a maximum of 10 epochs and adopt early stopping using the validation sets of each dataset. The hyper-parameters for fine-tuning neural rankers are listed in Appendix A.

5 Results and Discussion

5.1 In-Domain Fine-tuning

The results of in-domain fine-tuning of BERT and BART-based neural rankers on four passage ranking datasets are presented in table 3. To ensure a fair comparison, all the hyper-parameters between the ranking and the contrastive settings are kept the same and equal weights between ranking loss and

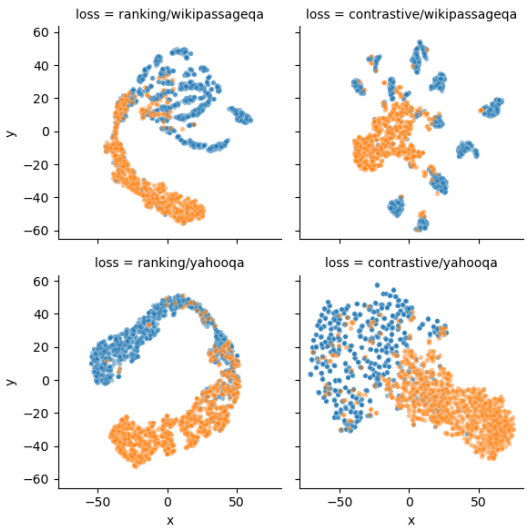

Figure 3: t-SNE plot of representations from BERT-based rankers trained with either ranking or combined ranking and contrastive loss for samples from WikiPassageQA (top) and YahooQA (bottom) test sets. Positive samples are represented by orange color.

contrastive loss are used. Our rankers produce state-of-the-art results when training with either of the two ranking-based losses **MHL** and **SHL**. Adding contrastive loss (**TML**) slightly improves the in-domain performance for BERT-based rankers, and performs similarly as ranking loss for BART-based rankers. Since our modified hinge loss (MHL) generally performs better than standard hinge loss (SHL), most of the results presented in the following sections are based on MHL, which corresponds to the setting illustrated in Figure 1.

To illustrate the effect of the contrastive loss on the representation space, we present the t-SNE plot of sample representations from the test set of two datasets WikiPassageQA and YahooQA in Figure 3. The color in the figures represents the positive and negative labels of query-passage pairs. As we can see from the plots, adding contrastive loss enables further separation of the positive samples from the negative samples.

5.2 Zero-Shot Transfer

The zero-shot transfer performance of neural rankers reflects their robustness to out-of-domain distributions, which is a key property of neural rankers since they are usually deployed in the wild. In Table 4, we show the results of applying the models trained in each one of the four datasets (source) and applied to the other three datasets (targets). Overall, we see significant improvements in the zero-shot transferability of the model across all datasets when the neural ranker is trained using the combination of ranking and contrastive losses. The biggest improvement is from YahooQA $\rightarrow$ WikiPassageQA where we observe absolute 9 points, 10.6 points, and 11.8 points improvement in MAP, MRR, and P@1, respectively. As expected, the transfer between datasets from similar domains tends to be better (e.g. WikiQA $\leftrightarrow$ WikiPassageQA) than that between dissimilar domains. Our intuition regarding the benefit of contrastive learning of representations to improve zero-shot transfer consists on the fact that, by using information from different queries and enforcing that positive pairs are close together in the embedding space and distant from negative pairs, the model ends up learning representations that are more general and therefore easier to transfer to new domains.

5.3 Robustness to Query Perturbations

In this section, we evaluate the model robustness to various types of reformulations and noisy transformations of the input queries. The test sets used in the experiments are the 6 variations of the WikiQA test set described in Sec. 4.2. We compare the results of using ranking loss (MHL) and the combination of ranking and contrastive loss (MHL+TML). The robustness evaluation is are presented in Table 5. As shown in Table 5, adding contrastive loss improves model robustness against all types of perturbations we tested. We also conduct experiments by fine-tuning neural rankers on combined SHL loss and TML loss. The robustness evaluation of BERT-based rankers trained on SHL loss or combined SHL loss and TML loss are presented in Table 6. Similar to the MHL case, the combined loss achieves a significant improvement in robustness than SHL loss only. More results on model robustness can be found in Appendix B.

5.4 Comparison with Data Augmentation

One of the traditional approaches for improving the robustness of machine learning models is to augment the training data with noisy data. In this section, we compare our contrastive fine-tuning method with a data augmentation approach in which automatically generated query reformulations are added to the training data. For each query in the training set, we use our fine-tuned T5 model to generate 5 new queries of each reformulation type (headline, paraphrase, change of voice). Effec-

Dataset →	WikiQA			WikiPassageQA			InsuranceQA			YahooQA		
Model	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
BERT												
BERTSel-base (2019)	75.3	77.0	-	-	-	-	-	-	-	94.2	94.2	-
BERT-PR-base (2019)	-	-	-	73.5	80.9	70.2	41.3	49.6	40.1	-	-	-
BERT-base-MHL (ours)	82.1	84.0	74.5	76.3	83.0	73.6	39.4	47.4	37.9	96.2	96.1	93.4
BERT-base-MHL+TML (ours)	83.8	85.8	77.4	76.9	83.1	73.6	41.1	49.6	39.5	96.1	96.1	93.4
BERT-base-SHL (ours)	82.3	84.1	75.7	74.2	81.2	71.6	40.0	47.6	37.3	95.9	95.9	92.9
BERT-base-SHL+TML (ours)	82.6	84.5	76.1	74.7	81.1	70.0	40.1	47.5	36.9	95.8	95.8	92.8
BART												
BART-base$_{LUL}$ (2020)	77.8	78.8	65.8	73.8	81.3	71.9	44.0	52.6	43.4	92.8	92.8	87.6
BART-base$_{RLL}$ (2020)	77.5	79.2	65.4	76.1	83.4	74.3	42.2	50.3	40.8	96.1	96.1	93.4
BART-base-MHL (ours)	85.8	87.4	78.2	77.8	85.3	77.6	43.5	51.8	42.0	96.5	96.5	94.0
BART-base-MHL+TML (ours)	84.6	86.1	75.7	77.4	84.5	76.2	43.4	51.9	42.4	96.5	96.5	93.9
BART-base-SHL (ours)	82.4	83.9	73.3	75.9	83.1	73.6	42.9	51.2	41.6	96.0	96.0	93.1
BART-base-SHL+TML (ours)	81.6	82.6	70.0	75.1	82.1	71.9	42.9	51.3	41.4	96.5	96.5	93.8

Table 3: In-domain results of neural rankers trained on ranking loss vs contrastive loss.

	Target →	WikiQA			WikiPassageQA			InsuranceQA			YahooQA		
Source Domain	**Loss**	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
BERT													
WikiQA	MHL	-	-	-	58.3	66.4	51.7	7.3	8.8	3.2	49.4	49.3	26.7
WikiQA	MHL+TML	-	-	-	58.4	66.3	51.0	11.6	14.9	7.5	50.1	50.1	27.9
WikiPassageQA	MHL	72.4	73.2	60.1	-	-	-	30.1	36.8	26.9	57.5	57.5	37.8
WikiPassageQA	MHL+TML	73.8	74.9	61.7	-	-	-	30.8	37.8	28.0	59.5	59.5	40.1
InsuranceQA	MHL	59.6	60.4	41.6	56.6	65.4	52.9	-	-	-	69.2	69.2	52.0
InsuranceQA	MHL+TML	61.1	62.3	45.3	57.2	65.1	52.2	-	-	-	78.0	78.0	64.3
YahooQA	MHL	35.3	36.3	16.5	30.2	32.5	15.6	3.8	4.4	0.9	-	-	-
YahooQA	MHL+TML	38.7	39.5	18.1	39.2	43.1	27.4	5.5	6.2	1.7	-	-	-
BART													
WikiQA	MHL	-	-	-	52.4	58.9	41.4	10.7	13.6	6.7	51.8	51.8	30.0
WikiQA	MHL+TML	-	-	-	58.6	67.1	51.9	14.0	17.7	9.7	52.4	52.4	30.9
WikiPassageQA	MHL	74.6	75.9	63.0	-	-	-	29.0	35.8	23.9	62.6	62.6	44.7
WikiPassageQA	MHL+TML	76.5	78.1	66.3	-	-	-	30.8	37.8	28.0	63.8	63.8	45.7
InsuranceQA	MHL	64.9	66.1	50.2	62.5	70.8	59.1	-	-	-	69.1	69.1	51.6
InsuranceQA	MHL+TML	65.5	66.3	50.6	63.3	72.2	61.8	-	-	-	71.7	71.7	55.4
YahooQA	MHL	34.1	35.4	15.6	16.2	18.6	7.0	2.2	2.7	0.6	-	-	-
YahooQA	MHL+TML	35.3	36.2	17.3	18.3	21.3	8.5	2.5	3.2	0.6	-	-	-

Table 4: Results of zero-shot transfer for models trained on ranking loss vs. contrastive loss.

	Reformulation →	Headline		Paraphrase		Chg of Voice		Punctuation		Typo		Contraction	
Training Data	**Loss**	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1
BERT													
WikiQA	MHL	75.3	64.9	79.4	69.3	72.6	60.0	82.3	74.5	76.8	66.3	82.6	77.6
WikiQA	MHL+TML	76.7	66.5	81.8	73.4	77.5	68.0	83.0	75.3	79.4	71.3	84.3	77.6
WikiPassageQA	MHL	61.8	45.5	67.3	51.9	64.2	48.0	71.7	58.9	57.1	39.2	74.7	64.3
WikiPassageQA	MHL+TML	63.3	47.5	68.3	53.5	68.9	56.0	72.3	60.1	58.8	40.8	74.9	64.3
BART													
WikiQA	MHL	80.6	69.4	84.3	75.9	76.8	64.0	83.8	74.5	82.5	73.8	86.3	79.6
WikiQA	MHL+TML	81.1	70.7	85.9	78.0	79.2	72.0	86.0	79.0	83.7	75.4	86.5	77.6
WikiPassageQA	MHL	72.0	58.7	75.3	62.7	73.4	60.0	74.6	62.6	70.0	55.8	79.1	71.4
WikiPassageQA	MHL+TML	74.3	62.0	76.1	63.5	76.0	64.0	77.3	67.1	69.3	56.7	79.6	70.4

Table 5: Robustness to various types of query reformulations and perturbations for rankers with MHL loss.

	Reformulation →	Headline		Paraphrase		Chg of Voice		Punctuation		Typo		Contraction	
Training Data	**Loss**	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1
WikiQA	SHL	73.1	62.0	79.6	71.0	74.9	64.0	81.0	72.4	74.7	63.3	80.8	73.5
WikiQA	SHL+TML	75.2	64.5	80.6	71.4	80.9	72.0	81.7	74.5	79.9	71.3	82.0	75.5
WikiPassageQA	SHL	60.4	45.5	66.9	51.9	54.6	40.0	68.9	54.7	55.1	36.7	69.6	55.1
WikiPassageQA	SHL+TML	62.5	46.3	66.9	51.9	62.7	48.0	68.9	54.7	56.9	39.2	72.0	58.2

Table 6: Robustness to various types of paraphrase for BERT-based rankers trained with SHL.

		Reformulation →	Headline			Paraphrase			Change of Voice		
Loss	**Notion of Similarity**	**Training Data**	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
MHL	-	WikiQA	75.3	76.9	64.9	79.4	80.9	69.3	72.6	73.6	60.0
MHL+TML	relevance label	WikiQA	76.7	78.0	66.5	81.8	83.2	73.4	77.5	78.6	68.0
MHL	-	WikiQA + headline	76.1	77.6	63.6	78.2	79.7	66.4	73.8	75.2	60.0
MHL+TML	query reformulations	WikiQA + headline	77.1	78.6	65.7	80.2	81.4	71.0	81.2	83.7	76.0
MHL	-	WikiQA + paraphrase	70.0	71.0	54.6	81.3	82.6	71.8	78.9	80.3	68.0
MHL+TML	query reformulations	WikiQA + paraphrase	75.1	76.4	63.2	81.2	82.4	70.5	80.3	83.1	76.0
MHL	-	WikiQA + chg voice	73.6	75.0	60.7	82.8	83.9	73.4	81.2	84.5	76.0
MHL+TML	query reformulations	WikiQA + chg voice	75.3	76.6	63.2	83.0	84.3	73.4	83.3	88.5	80.0

Table 7: Comparison with data argumentation.

tively, we increase the training set and the number of training steps by a factor of 5 for each reformulation type. Since our proposed training approach is general and can be used with any dataset, we also experiment with data augmentation combined with contrastive fine-tuning. When performing contrastive fine-tuning, we use the 5 reformulations of each query to create similar pairs (the notion of similarity == query formulation) in each batch, which essentially keeps the number of training steps the same as when training with the original dataset. The results on data augmentation for BERT-based neural rankers are presented in Table 7. For rows with MHL loss, we argument the training data with paraphrased queries and train the model on a combined dataset using MHL loss only. Note this is the standard way to do data argumentation training. For rows using MHL+TML loss, we pair each query in the batch with its paraphrased query for contrastive loss calculation. The model is trained on combined MHL+TML loss. For both methods, we expose the model with the same amount of paraphrased training samples. As we can see from the table, augmenting the training data with a similar type of query reformulation can improve the robustness of the model against that particular type of reformulation. However, it is not as effective in improving the robustness against other reformulation types. On the other hand, contrastive fine-tuning, even trained with a single type of query reformulation can generally improve the model robustness against the other two types. Furthermore, contrastive fine-tuning achieves this with significantly less ($4\times$ less) training time even after considering the additional computation of the contrastive loss calculation. Essentially, the experimental results indicate that using paraphrased training samples to perform contrastive learning is both effective (produces more robust rankers) as well as efficient (faster to train since augmented data is used in parallel, i.e. same batch as original data) than using regular data augmentation.

5.5 Comparison with Ranking Loss using same Batch Size

When training with MHL plus contrastive loss we effectively increase the batch size because we need to augment the training batch with additional positive samples from different queries. To check if the improvements achieved by our approach are due to the increase of training batch size only, we perform an ablation study where we compare the performance of models trained with a ranking loss but with the same batch size as the contrastive fine-tuning setting. The comparison results are presented in Table 8. As we can see in the table, for WikiQA datase, increasing the batch size helps the performance of in-domain and some of the robustness test sets. The contrastive setting still outperforms the ranking setting in all the test categories. Increasing the batch size of MHL is not always beneficial. We see big degradation on the WikiPassageQA dataset. On the other hand, we observed consistent improvement when the model is trained with contrastive loss.

6 Ablation Study

We present ablation experiments that check the impact of the number of positive samples per batch and the use of hard negative mining. Additionally, in Appendix C, we also present a preliminary experiment on formulating our combined loss (Equation 4) as a multi-objective optimization (MOO).

6.1 Effect of Number of Positive Samples Per Batch

The number of positive samples within a single batch determines the total number of potential triples constructed. In this section, we vary the number of positive samples within a batch and evaluate its effect on the model performance. The results are presented in Table 9. As expected, the model performance benefits by increasing the number of positives in a batch. As shown in Table 9, although not strictly monotonically, both the in-domain performance and zero-shot transfer performance improve with the number of positive pairs.

6.2 Effect of Hard Negative Mining

In a batch of N samples, there are $O(N^3)$ possible triplets, many of which are not very helpful to model convergence (e.g triplets where $D(a, k^+) >> D(a, k^-)$). It's important to construct only the most important triplets. Many works have discussed the benefit of hard negative mining techniques that produce useful gradients and help the models converge quickly. In this section, we explore the effect of three hard negative mining methods that are compatible with TML: Angular miner (Wang et al., 2017) (output triplets that form an angle greater than a threshold), BatchHard (Hermans et al., 2017) (produce a single triplet for each

Loss	Batch Size	Test Set → Training Set	In-domain MAP	In-domain P@1	Headline MAP	Headline P@1	Paraphrase MAP	Paraphrase P@1	Chg of Voice MAP	Chg of Voice P@1
MHL	16	WikiQA	82.1	74.5	75.3	64.9	79.4	69.3	72.6	60.0
MHL	31	WikiQA	83.0	76.1	74.9	64.1	81.0	71.8	77.2	68.0
MHL+TML	31	WikiQA	83.8	77.4	76.7	66.5	81.8	73.4	77.5	68.0
MHL	16	WikiPassageQA	76.3	73.6	61.8	45.5	67.3	51.9	64.2	48.0
MHL	31	WikiPassageQA	76.2	73.8	58.7	40.5	64.4	47.7	57.1	40.0
MHL+TML	31	WikiPassageQA	76.9	73.6	63.3	47.5	68.3	53.5	68.9	56.0

Table 8: Comparison of ranking loss and contrastive loss with same batch size for BERT-based rankers.

Target →	WikiQA			WikiPassageQA			InsuranceQA			YahooQA		
# Pos / Batch	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
2	82.4	84.1	74.9	58.1	66.4	51.7	12.8	16.3	8.0	46.4	46.3	24.2
4	83.4	85.2	76.1	58.1	66.1	51.0	12.1	15.3	7.0	47.9	47.8	26.0
8	83.5	85.3	75.7	59.9	67.4	52.9	12.1	15.2	7.1	48.7	48.7	26.8
16	83.8	85.8	77.4	59.4	66.3	51.0	11.6	14.5	6.4	50.1	50.1	27.9
32	83.3	85.2	76.1	59.0	65.9	49.3	10.4	12.9	4.9	49.9	49.9	27.6
64	82.5	84.2	74.5	61.4	69.1	55.1	13.0	16.7	8.2	49.2	49.2	26.9

Table 9: Effect of number of positive pairs per batch. Underlined cells indicate in-domain results.

Target →	WikiQA			WikiPassageQA			InsuranceQA			YahooQA		
Mining Method	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
BERT												
No Mining	83.8	85.8	77.4	59.4	66.3	51.0	11.6	14.9	7.5	50.1	50.1	27.9
Angular	83.8	85.6	76.1	59.1	66.2	51.0	6.7	8.1	3.1	49.7	49.6	27.4
BatchHard	82.3	84.0	73.7	63.0	70.7	57.0	10.7	13.9	6.4	52.0	52.0	30.2
TripletMargin	83.6	85.6	77.0	59.4	66.4	51.0	7.2	8.7	3.2	50.1	50.1	27.8
BART												
No Mining	84.6	86.1	75.8	58.6	67.1	51.9	14.0	17.7	9.7	52.4	52.3	30.9
Angular	84.5	86.1	76.1	59.9	67.9	53.1	16.0	20.0	11.3	56.4	56.4	35.6
BatchHard	85.4	86.7	77.0	57.8	66.2	50.5	12.5	15.8	7.9	55.3	55.3	33.9
TripletMargin	85.9	87.5	78.6	61.2	69.6	55.3	16.0	19.9	11.0	56.6	56.5	35.8

Table 10: Effect of hard negative mining. Underlined cells indicate in-domain results.

anchor point consisting of the hardest positive and hardest negative samples), and TripletMargin (only output a triplet when the difference between the anchor-positive distance and the anchor-negative distance is smaller than a margin). The results of hard negative mining on models trained on WikiQA dataset are presented in Table 10, in which we evaluate both the in-domain and zero-shot performance of the rankers. As we can see from the results, hard negative mining can further improve the transferability of of both BERT-based ranker and BART-based ranker. In particular, BatchHard outperforms other mining methods and improve the overall performance significantly for BERT-based rankers while TripletMargin is more effective for BART-based rankers. We believe there is still a margin for improvement if the hyper-parameters of the miners are properly tuned.

7 Conclusion

In this paper, we propose a novel method for fine-tuning neural rankers by combining contrastive loss with ranking loss. Using a semi-automatic approach, we created 6 new versions of WikiQA test set to assess the robustness of our models to query reformulations and perturbations. Our experimental results show that the proposed method improves ranker's robustness to out-of-domain distributions, query reformulations, and perturbations. Comprehensive experiments and ablation studies were conducted to investigate the impact of some design choices as well as to confirm that the gains do not originate only from larger batch sizes. Contrastive fine-tuning with generated data is more effective than data augmentation. As future work, we plan to evaluate the performance of other state-of-the-art contrastive loss functions and novel methods of aggregating multiple losses.

References

Parminder Bhatia, Lan Liu, Kristjan Arumae, Nima Pourdamghani, Suyog Deshpande, Ben Snively, Mona Mona, Colby Wise, George Price, Shyam Ramaswamy, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, Bing Xiang, and Taha Kass-Hout. 2020. AWS CORD-19 Search: A Neural Search Engine for COVID-19 Literature.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image sim-

ilarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2020. Robust neural machine translation with doubly adversarial inputs. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2018, pages 4324–4333.

Daniel Cohen, Liu Yang, and W Bruce Croft. 2018. WikiPassageQA: A Benchmark Collection for Research on Non-factoid Answer Passage Retrieval. In *SIGIR*, pages 1–4.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Cicero Nogueira dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Beyond [CLS] through Ranking by Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1722–1727, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2019. Adversarial Attacks on Deep Learning Models in Natural Language Processing: A Survey. 1(1):40.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2016. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015 - Proceedings*, pages 813–820. Institute of Electrical and Electronics Engineers Inc.

Wee Chung Gan and Hwee Tou Ng. 2020. Improving the robustness of question answering systems to question paraphrasing. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 6065–6075.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, volume 32.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In defense of the triplet loss for person re-identification.

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. pages 1–24.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Yaochu Jin, Tatsuya Okabe, and Bernhard Sendhoff. 2004. Neural network regularization and ensembling using multi-objective evolutionary algorithms. In *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, volume 1, pages 1–8.

Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. 2001. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: {W}hy Does It Work and How? In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*, pages 1042–1049.

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)*, pages 6769–6781.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the Difference that Makes a Difference with Counterfactually-Augmented Data.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. Natural Perturbation for Robust Question Answering.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Dilip Krishnan, and Ce Liu. 2020. Supervised contrastive learning.

Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive Representation Learning: A Framework and Review. *IEEE Access*, 8:193907–193934.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Technical report.

Dongfang Li, Yifei Yu, Qingcai Chen, and Xinyu Li. 2019. BERTSel: Answer selection with pre-trained models.

Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2020. Neural passage retrieval with improved negative contrast.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain Adaptation with BERT-based Domain Classification and Data Selection. pages 76–83. Association for Computational Linguistics (ACL).

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic Data Augmentation Increases Robustness to Inference Heuristics. pages 2339–2352.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. Pytorch metric learning.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

A. Radford, J. W. Kim, Chris Hallacy, Aditya Ramesh, G. Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, J. Clark, G. Krüger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 856–865.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP models with CheckList. pages 4902–4912.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. 2019. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 6642–6651.

Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. 2020. Benchmarking robustness of machine reading comprehension models.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-BERT: BERT is not robust on misspellings! Generating nature adversarial samples on BERT.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Learning to rank question answer pairs with holographic dual LSTM architecture. In *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704. Association for Computing Machinery, Inc.

Aaron Van Den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding.

Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep Metric Learning with Angular Loss. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 2612–2620. Institute of Electrical and Electronics Engineers Inc.

Yicheng Wang and Mohit Bansal. 2018. Robust Machine Comprehension Models via Adversarial Training. pages 575–581.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2020. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 5878–5882. Association for Computational Linguistics.

Jason W. Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks.

Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. Technical report.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: Contrastive Learning for Sentence Representation.

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.

Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Passage ranking with weak supervision.

Yi Yang, Wen-Tau Yih, and Christopher Meek. 2015. WIKIQA: A Challenge Dataset for Open-Domain Question Answering. *Proceedings of EMNLP 2015*, (September 2015):2013–2018.

Fan Yin, Quanyu Long, Tao Meng, and Kai-Wei Chang. 2020. On the Robustness of Language Encoders against Grammatical Errors. pages 3386–3403.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. 2019. Bridging theory and algorithm for domain adaptation. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 12805–12823.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2020. Robust Reading Comprehension with Linguistic Constraints via Posterior Regularization. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28:2500–2510.

Shuyan Zhou, Xiangkai Zeng, Yingqi Zhou, Antonios Anastasopoulos, and Graham Neubig. Improving Robustness of Neural Machine Translation with Multi-task Learning. Technical Report 1.

A Details of Neural Ranker Fine-tuning

The fine-tuning of neural rankers is conducted on an AWS EC2 P3 machine. Important hyper-parameters of fine-tuning for each model-dataset combination are listed in Table 11.

B More Results of Robustness Against Query Perturbations

In this section, we present more results of model robustness evaluation for neural rankers trained on InsuranceQA and YahooQA datasets. The results are shown in Table 12.

C Fine-tuning as Multi-objective Optimization

We performed a preliminary experiment on formulating equation 4 as a multi-objective optimization (MOO) problem in which optimizing both $\mathcal{L}_{ranking}$ and $\mathcal{L}_{contrastive}$ are two objectives of the task. We adopt a *dynamic weighted aggregation* (DWA) method (Jin et al., 2001, 2004) which is both effective and computationally efficient. In DWA, the weights of the two loss terms are changed gradually according to the following equations:

$$w_1(t) = |\sin 2\pi t/\ F| \quad (5)$$

$$w_2(t) = 1 - w_1(t) \quad (6)$$

where t is the iteration number. It is noticed that $w_1(t)$ changes from 0 to 1 periodically. The change frequency can be adjusted by F.

Figure 4 shows the evolution of the contrastive loss during fine-tuning of the BART-based ranker on the WikiQA dataset. As can be seen from

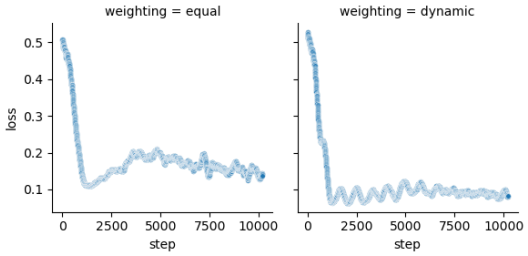

Figure 4: Training contrastive loss for neural ranker trained on WikiQA.

the plot, adopting the MOO method improves the model convergence. A lower contrastive loss is achieved using dynamic weighting which translates to an average improvement of 0.7 points over the equal weighting setting in zero-shot transfer performance (see Table 13).

Training Data	Loss	Learning Rate	Positives / Batch	Negatives / Batch	Gradient Accumulation	Block Size	Hinge Loss Margin
			BERT				
WikiQA	MHL	5e-6	1	15	8	256	2
WikiQA	MHL+TML	5e-6	16	15	8	256	2
WikiPassageQA	MHL	1e-5	1	15	8	256	2
WikiPassageQA	MHL+TML	1e-5	16	15	8	256	2
InsuranceQA	MHL	5e-5	1	15	16	256	2
InsuranceQA	MHL+TML	5e-5	16	15	16	256	2
YahooQA	MHL	1e-5	1	4	8	256	2
YahooQA	MHL+TML	1e-5	16	4	8	256	2
WikiQA	SHL	5e-6	15	15	8	256	2
WikiQA	SHL+TML	5e-6	15	15	8	256	2
WikiPassageQA	SHL	1e-5	15	15	8	256	2
WikiPassageQA	SHL+TML	1e-5	15	15	8	256	2
InsuranceQA	SHL	5e-5	15	15	16	256	2
InsuranceQA	SHL+TML	5e-5	15	15	16	256	2
YahooQA	SHL	1e-5	4	4	8	256	2
YahooQA	SHL+TML	1e-5	4	4	8	256	2
			BART				
WikiQA	MHL	5e-6	1	15	8	256	2
WikiQA	MHL+TML	5e-6	16	15	8	256	2
WikiPassageQA	MHL	1e-5	1	15	8	256	2
WikiPassageQA	MHL+TML	1e-5	16	15	8	256	2
InsuranceQA	MHL	5e-5	1	15	16	256	2
InsuranceQA	MHL+TML	5e-5	16	15	16	256	2
YahooQA	MHL	1e-5	1	4	8	256	2
YahooQA	MHL+TML	1e-5	16	4	8	256	2
WikiQA	SHL	5e-6	15	15	8	256	2
WikiQA	SHL+TML	5e-6	15	15	8	256	2
WikiPassageQA	SHL	1e-5	15	15	8	256	2
WikiPassageQA	SHL+TML	1e-5	15	15	8	256	2
InsuranceQA	SHL	5e-5	15	15	16	256	2
InsuranceQA	SHL+TML	5e-5	15	15	16	256	2
YahooQA	SHL	1e-5	4	4	8	256	2
YahooQA	SHL+TML	1e-5	4	4	8	256	2

Table 11: Hyper-parameters for neural ranker fine-tuning.

	Reformulation →	**Headline**		**Paraphrase**		**Chg of Voice**		**Punctuation**		**Typo**		**Contraction**		
Training Data	**Loss**	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	MAP	P@1	**Avg MAP**
							BERT							
InsuranceQA	MHL	53.2	34.7	57.4	39.8	44.4	24.0	59.2	41.2	42.8	23.4	60.4	42.9	52.9
InsuranceQA	MHL+TML	54.9	35.1	58.9	41.5	47.9	28.0	60.6	44.4	44.4	25.0	60.0	42.9	**54.5**
YahooQA	MHL	40.5	22.3	34.0	14.1	28.8	4.0	36.7	16.9	34.0	14.6	38.3	16.3	35.4
YahooQA	MHL+TML	41.0	21.9	35.5	14.5	29.4	8.0	40.1	20.2	35.5	15.0	41.6	20.4	**37.2**
							BART							
InsuranceQA	MHL	61.2	46.3	64.2	50.2	52.6	32.0	63.6	48.2	54.7	37.9	68.6	54.1	60.8
InsuranceQA	MHL+TML	63.4	50.8	65.0	50.6	58.5	44.0	64.7	50.2	54.6	37.5	68.8	56.1	**62.5**
YahooQA	MHL	35.7	17.4	32.9	13.7	29.7	8.0	34.2	15.2	33.6	15.8	36.5	15.3	33.8
YahooQA	MHL+TML	37.2	18.2	34.4	15.8	30.7	8.0	35.2	16.9	34.6	17.5	37.3	17.4	**34.9**

Table 12: Additional restuls of robustness to various types of paraphrase.

Target →	**WikiQA**			**WikiPassageQA**			**InsuranceQA**			**YahooQA**			
Weighting Method	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	**Avg MAP**
						BERT							
Equal Weighting	84.6	86.1	75.7	58.6	67.1	51.9	14.0	17.7	9.7	52.4	52.3	30.9	52.4
Dynamic Weighting	84.6	86.0	75.8	58.0	66.5	50.7	13.6	17.2	8.7	56.2	56.2	35.4	**53.1**

Table 13: Dynamic weighting vs equal weighting.