Bi-CAT: Improving Robustness of LLM-based Text Rankers to Conditional Distribution Shifts

Sriram Srinivasan*
Amazon LLC
USA
srirs@amazon.com

Stephen Sheng*
Amazon LLC
USA
shenstep@amazon.com

Rishabh Deshmukh
Amazon LLC
USA
derishab@amazon.com

Chen Luo
Amazon LLC
USA
cheluo@amazon.com

Yesh Dattatreya
Amazon LLC
USA
ydatta@amazon.com

Subhajit Sanyal
Amazon LLC
USA
subhajs@amazon.com

SVN Vishwanathan
Amazon LLC
USA
vishy@amazon.com

ABSTRACT

Retrieval and ranking lie at the heart of several applications like search, question-answering, and recommendations. The use of Large language models (LLMs) such as BERT in these applications have shown promising results in recent times. Recent works on text-based retrievers and rankers show promising results by using bi-encoders (BE) architecture with BERT like LLMs for retrieval and a cross-attention transformer (CAT) architecture BERT or other LLMs for ranking the results retrieved. Although the use of CAT architecture for re-ranking improves ranking metrics, their robustness to data shifts is not guaranteed. In this work we analyze the robustness of CAT-based rankers. Specifically, we show that CAT rankers are sensitive to item distribution shifts conditioned on a query, we refer to this as *conditional item distribution shift* (CIDS). CIDS naturally occurs in large online search systems as the retrievers keep evolving, making it challenging to consistently train and evaluate rankers with the same item distribution. In this paper, we formally define CIDS and show that while CAT rankers are sensitive to this, BE models are far more robust to CIDS. We propose a simple yet effective approach referred to as Bi-CAT which augments BE model outputs with CAT rankers, to significantly improve the robustness of CAT rankers without any drop in in-distribution performance. We conducted a series of experiments on two publicly available ranking datasets and one dataset from a large e-commerce store. Our results on dataset with CIDS demonstrate that the Bi-CAT model significantly improves the robustness of CAT rankers by roughly 100-1000bps in F1 without any reduction in in-distribution model performance.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; **Language models**; *Online shopping*.

KEYWORDS

information retrieval, language models, ranker, robustness

*Both authors contributed equally to this research.

ACM Reference Format:
Sriram Srinivasan, Stephen Sheng, Rishabh Deshmukh, Chen Luo, Yesh Dattatreya, Subhajit Sanyal, and SVN Vishwanathan. 2024. Bi-CAT: Improving Robustness of LLM-based Text Rankers to Conditional Distribution Shifts. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3589335.3651947

1 INTRODUCTION

With drastic leaps in deep learning (DL) and large language models (LLMs), search systems have been tapping into the power of these techniques to improve their search systems to give customers a better search experience [4, 10, 13, 15, 29]. Production search systems have several stages of data processing before serving the final list to the customers, of which retrieval and ranking are two key stages that have largely adopted the use of DL [15] and LLM [29]. The retrieval task is generally trained with focus on recall and efficiency where the model is capable of retrieving thousands of items given a query from a catalog with billions of items. The ranking task, on the other hand, focuses on precision and is tasked with re-ranking the retrieved results to improve their relevance in the top-k results. Production search systems typically use several retrievers such as BM25, bi-encoder semantic matchers, behavioral data cache, etc., followed by gradient boosted decision trees (GBDTs), multilayer perceptrons (MLPs), etc. for ranking. With drastic improvements in LLMs, several approaches in production off-late have started adopting LLMs and training cross-attention transformer (CAT) models to improve ranking performance[10, 16, 19]. While these LLMs are powerful and produce significant performance boosts on test datasets they are generally not tested for robustness to datasets not belonging to same distribution as the train dataset. Since rankers are the final gateway in several search systems, it is important for them to be robust to minimize negative user impact.

Previous research [3, 26] has focused on making ranking models more robust to shift in distribution. These studies primarily address two types of distribution shifts: re-ranking items for queries that the model has not seen before, and re-ranking items that have

been recently added to the catalog. The first part fall under zero-shot learning problems and the latter under cold-start problem. Approaches such as [22, 24] focus on improving model robustness on typos and domain adaptation tasks by performing training with data augmentation. Other approaches [17, 21] explored to improve model performance by improving the representation by combining encoders, or denoising strategies which also leads to robust models. More works such as [9, 24] improve performance and robustness of models by combining BE and CAT models, such as iteratively train BE and CAT or data augmentations from CAT to BE. Several approaches [8, 11, 22] leverage contrastive loss in CAT rankers in novel ways to improve both robustness and model performance. Yet another approach [8] combines external information using knowledge graphs to debias the learnings and improve model performance. While it is generally accepted that CAT rankers perform better than BE models, [12] propose distillation approach and using BE to bridge this gap. While these issues are important, little research has been conducted on making search rankers robust to *conditional item distribution shift* (CIDS), where the set of items to re-rank changes based on a query at inference time. Given that the retrievers and rankers may be trained independently in production systems, it is important for the rankers to be robust to any CIDS introduced by a change in retriever's output.

In this paper, we focus on making CAT models used as rankers more robust to CIDS. We first explain why CAT rankers are not robust to CIDS and then propose a simple-yet-effective approach called Bi-CAT, which ensembles the output of a bi-encoder (BE) model with a CAT ranker. Through our experiments, we show that Bi-CAT models improve the robustness of the CAT ranker to CIDS without affecting the model's performance on in-distribution data. Overall the key contributions of our work are: 1) Introduce the concept and importance of CIDS in search systems and demonstrate that widely used powerful CAT rankers may not be robust to CIDS. 2) Introduce a novel approach, Bi-CAT, which improves the robustness of CAT rankers to CIDS without degrading overall performance. 3) Demonstrate through empirical evaluations on three large datasets that the Bi-CAT approach performs comparably to CAT ranker when evaluated with in-distribution data, and outperforms it by 100-1000bps in F1 on datasets with CIDS.

2 RELATED WORK

Several works off-late focus on robustness in search systems and combining BE and CAT to improve model performance. Approaches such as [22, 24] focus on improving model performance on typos and domain adaptation tasks by performing training with data augmentation. Other approaches [17, 21] try to improve model performance by improving the representation by combining encoders, or denoising strategies which also leads to robust models. More works such as [9, 24] improve performance and robustness of models by combining BE and CAT models, such as iteratively train BE and CAT or data augmentations from CAT to BE. Several approaches [8, 11, 22] leverage contrastive loss in CAT rankers in novel ways to improve both robustness and model performance. Yet another approach [8] combines data from knowledge graphs to debias the learnings and improve model performance. While it is generally accepted that CAT rankers perform better than BE models, [12] propose distillation approach and using BE to bridge this gap. While all these approaches focus on robustness in search, none explicitly address the issue of robustness to CIDS. Our paper primarily focuses on making CAT rankers robust to CIDS without losing its high effectiveness on any given task.

3 BACKGROUND

In this section, we will first cover the Bi-Encoder (BE) approach, which is commonly used for retrieval, followed by the CAT model, a commonly used re-ranker.

3.1 Bi-Encoder as Retriever

BE models are one of the most effective and popular approaches used in retrieval tasks [17] as they can efficiently produce a list of items for any query from a large catalog. Although there are several variants of this approach, they all follow a similar architecture. Figure 1a shows the general architecture of a BE model. In the standard architecture, the query text (q) and item text (i) are tokenized, then passed through an encoder to generate embeddings $\phi(q)$ and $\phi(i)$, respectively. A function θ, such as dot or $cosine$, is applied to the embeddings, followed by a loss function. The approach is summarized in the equation below.

$$q_{emb} = \phi(q);\ i_{emb} = \phi(i);\ score_{q,i} = \theta(q_{emb}, i_{emb});$$
$$loss = l(score_{q,i}, y_{qi})$$

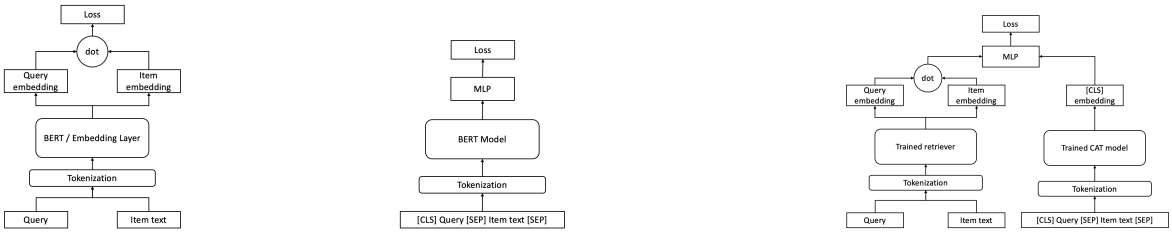
where l is a loss function and y_{qi} is the true label for the pair (q, i). Several variants of this approach has been proposed by modifying the θ function [2, 7, 21] and loss function [1, 2, 6, 15, 17, 21]. Further explorations use different negative mining approaches [18]. In this paper, we use $\theta = cosine$ and $l = contrastive\ loss$ as it has shown highest improvements with in-batch negatives and hard negatives.

3.2 CAT Ranker

CAT models are generally used to re-rank the retrieved top-k set of items as they are lot more compute intensive but precise [10, 16, 19]. Figure 1b shows a high-level architecture of the a CAT ranker. At a high-level, the query text and item text are concatenated with an [SEP] token and tokenized before passing to the encoder. A transformer model is then used to get an embedding for the query-item pair. This embedding is then passed through a sequence of linear layers with activation to get a final score. Finally, a loss function is used with the score to train the network. The overall training can be summarized using equations below:

$$score_{q,i} = \phi(q, i);\ loss = l(score_{q,i}, y_{qi}),$$

where the ϕ is the combination of the transformer and linear layers. Several loss functions have been proposed for CAT rankers [1, 5, 23, 25, 25]. In this work, we simply use the binary cross-entropy (BCE) loss for its simplicity, effectiveness and efficiency, and is generally a production standard choice. While "a positive example" in ranker problem is given, negatives are sampled from some distribution. In [28], the author shows that we can get better performance when the negative is sampled from the retrieved results. So, we follow this approach that the positives are given from data and negatives are sampled from retrieved top-K if not explicitly given from the dataset (more details in Section 5).



(a) **Retriever model: BE model.** (b) **Ranker model: CAT ranker model.** (c) **Robust ranker: Bi-CAT ranker model.**

Figure 1: Retriever and ranker models.

4 ROBUST RANKER

In this section, we discuss the issue of robustness in a CAT models using an example and finally propose Bi-CAT approach to make the CAT models more robust.

4.1 Robustness problem

Large-scale search systems use a variety of retrievers along with BE models for both business and coverage reasons. Other retrievers such as BM25 [20], paid ads, etc. are mixed before passing the results to a ranker. Therefore, there is no guarantee that the retrieved item distribution at inference is same as during training. This implies that there could be a shift in distribution of item features/text conditioned on a query (referred to as CIDS) and the final rankers need to be robust to this shift to avoid poor customer experience. We formally define CIDS as follows:

Definition 4.1. *Let q be a query, I_q^{tr} a set of items in the training data with feature distribution $F(I_q^{tr})$, and I_q^{te} a set of items in evaluation dataset with feature distribution $F(I_q^{te})$. We define the data to have undergone a CIDS if the feature distribution of $F(I_q^{tr}) \neq F(I_q^{te})$.*

Given CIDS is common in production, we need to ensure that the high precision CAT rankers are robust to such shifts. However, given that CAT models use query-item pair as training data point, it is not feasible to train the models with all possible negative pairs. This could lead to unforseen behaviors on CIDS. Below is an toy example on how exposure to specific negatives only can be detremental:

Example 4.1. *Consider a training set with a set of labled query-item pairs where every positive pair is an exact matches and negative is not. Let the negative pairs be hard negatives with query-item pairs obtained from production matchset with mostly near exacts. To make the example more concrete, let q = "15 kg laundry detergent" and a positive item i_p = "Tide 15 kg washing powder for washing machine" and negative item i_n = "Ariel 1 kg laundry detergent liquid heavy duty". Given such data during training, the model might end up learning that for the query above, '15 kg' is the most important keyword. However, at inference if there is a distribution shifts and the model is made to predict a label for a new item not seen during training "Basmati rice 15 kg bag.", the model is likely to mark this as an exact as the item also contains '15 kg'.*

We show that CIDS indeed have a significant effect in our experiments section. To overcome this, we propose Bi-CAT which improves robustness of CAT to CIDS with minimal changes to the original results.

4.2 Bi-CAT: a Robust CAT ranker

To make CAT ranker more robust to CIDS, we propose augmenting BE with CAT. Figure 1c shows a high-level architecture of the Bi-CAT approach. Essentially, we train an MLP that combines the output of a BE and CAT model. While we could train the model end-to-end, which is a lot more expensive, we propose to train the CAT ranker and BE model independently and then combine them. During the training of Bi-CAT we freeze the parameters of both the BE and CAT. This makes the training of Bi-CAT efficient (abalation study in Section 5.5). We summarize our model by the equations below:

$$score_{q,i}^{be} = \theta(q_{emb}, i_{emb}); \quad cls_{emb}^{cat} = \hat{\phi}_{cat}(q, i)$$
$$score_{q,i}^{bicat} = mlp(cls_{emb}^{cat}, score_{q,i}^{be})$$

where q_{emb} and i_{emb} are query and item embedding from trained and frozen BE $\hat{\phi}_{cat}$ is the trained CAT with frozen parameters, and mlp is a set of trainable layers with activation.

4.3 Why is Bi-CAT more robust?

To understand why Bi-CAT models are more robust than CAT models, we first show why BE are likely to be more robust than CAT and then answer our original question.

4.3.1 Why is BE more robust? Intuitively, BE models are less susceptible to performance drop compared to CAT models on CIDS dataset because of two reasons. First, BE models are faster to train and because of this, incorporation of in-batch negative sampling and other negative augmentations are feasible. This implies that the model is generally trained with a lot more diverse set of negatives, making them more robust to different set of negatives. For instance, if the batchsize used is B, then the number of query-item pairs used can be $B \times B$ with almost no additional cost to training, which is not possible in a CAT ranker. Second, BE models are generally trained for easy retrieval, which implies the model must be capable of differentiating a positive item given the entire catalog. To do this, they make use of geometry and create embeddings such that a dot product or a variant of a dot product is correlated with relevance. This leads to the model being constrained geometrically to place relevant items close to each other i.e., if i_1^{emb} is the embedding of item 1 and i_2^{emb} is the embedding of item 2, then $cos(i_1^{emb}, i_2^{emb}) \approx 1$ if item 1 and item 2 are similar to each other and ≈ -1 otherwise.

Further, if we assume that CIDS is mainly caused due to a change in negative item distribution, then with high probability such items will have a low score in a BE model. This is because the item unseen in training will likely be assigned a random embedding (from a normal distribution) and cosine similarity of any query embedding with any random embedding will be centered around 0. This makes the BE model robust to CIDS caused by negative examples. However, in a CAT model with binary label such unseen items are likely to produce a random score between 0 and 1 which implies a 50% probability of assigning an incorrect label which makes CAT models more susceptible to CIDS caused by a shift in negatives than BE models. Formally, we can state the following:

Proposition 4.1. *A trained BE model used as a ranker is more robust to CIDS caused by a change in negative samples compared to a CAT model trained with the same data.*

4.3.2 How does BE help in Bi-CAT?. The Bi-CAT model uses both the output of BE and CAT inheriting the robustness form BE and effectiveness from CAT. BE enhances the robustness of the CAT ranker for CIDS by identifying several types of negatives as mentioned in Section 4.3.1, while the inclusion of the CAT component also makes the embeddings effective at being precise at identifying positives from harder negatives. As a result, the Bi-CAT model is both robust and as effective as CAT models. We provide a detailed robustness analysis in section 5.4.

4.4 General Applicability of Bi-CAT on Other Tasks

The effect of CIDS on CAT rankers has been mentioned so far. But the effect of CIDS can occur on any task with pair inputs that use CAT model, such as sentence pair classification. While Bi-CAT models are easy to apply to any ranker, the method can be applied in general to any CAT model to improve robustness to CIDS. Any task that uses a CAT model can be improved by switching to a Bi-CAT model. This would involve training a BE model with the same training data and then create a Bi-CAT model by combining the BE and CAT models. Such classifiers will be more robust to CIDS compared to the original CAT model with no drop in original performance.

5 EMPIRICAL EVALUATION

In this section, we evaluate Bi-CAT on multiple ranking/classification tasks using two public datasets and one proprietary dataset. We primarily aim to answer the following questions: (1) Can Bi-CAT have similar performance as CAT on *'in-distribution'* data? (2) Are BE and Bi-CAT models more robust than CAT model on data with CIDS? (3) Is Bi-CAT more robust than CAT and yet retain the same performance as CAT? (4) Why is Bi-CAT more robust than CAT and yet retain the same performance as CAT?

5.1 Data and Models Details

In this subsection, we go over the details of the task and dataset used, and the baseline models and model details used to answer the questions above.

5.1.1 Datasets. We use two publicly available MS MARCO [14] and ESCI [19] datasets an one proprietary dataset from a large e-commerce store.

MS-MARCO Dataset [14]: is a large scale MAchine Reading COmprehension dataset frequently used to benchmark question-answering models. The MS MARCO dataset consists of 1.01M anonymous questions from Bing's search query logs, each with a human-generated answer and 182K human-rewritten answers. The dataset also includes 8.8M passages from 3.5M web documents retrieved by Bing, which provide the information for the natural language answers.

ESCI Dataset [19]: is a large-scale e-commerce multi-lingual data with each row query and product information and a label of exact (E), or substitute (S), or complement (C), or irrelevant (I). We use this dataset for two tasks, 1) exact classification: *E v.s. SCI classification task (E-SCI)*, and 2) irrelevant classification: *ESC v.s. I classification task (ESC-I)* as they are the commonly used as search rerankers. The dataset contains 2M training rows, and 600K test rows. We use 100K rows from train as validation to perform early stopping. Full dataset contains around 100K train queries and 30K test queries with an average depth of 20 items per query. The ratio between E and SCI is 2:1, and the ratio between ESC and I is 4:1, in both tasks we consider the first as positive class.

Proprietary E-commerce Search (PES) Dataset: We sample one year's weekly rolled-up search data from a large e-commerce store. Any item purchased for a query is marked as positive labels. We use 16M positive rows with 3M queries and 28M catalog size. For evaluation we compile a test data by sampling 30K queries from subsequent four weeks data. For validation dataset, we sample a 7k positive and equal number of negative rows from the training data.

5.1.2 Metrics. To understand the performance of our models we use standard set of metrics for the tasks. For the MS-MARCO datasets we perform retrieval of 1000 items per query and re-rank and compute Recall@100 (R@100) and MRR@10. Additionally we also mention R@1000 to indicate the maximum attainable recall when using the specific retriever. For both the tasks using the ESCI dataset we use ROCAUC and F1-Score of the positive class.

5.1.3 Models Details. To compare and understand Bi-CAT performance, we use the following set of models:

BM25: is a standard BM25 model.

BE: is a bi-encoder model trained with a contrastive loss function and cosine similarity scoring function. Model uses in-batch negatives and any explicit negatives provided. The pretrained BERT encoder used is a publicly available $bert-base-uncased$ model for MS-MARCO dataset, $bert-base-multilingual-uncased$ model for ESCI dataset, and a 2-layer in-house BERT model for PES dataset.

CAT: is a crossattention ranker trained with binary crossentropy (BCE) loss and negatives from the retriever or the negatives provided with the dataset. The pretrained BERT encoder used is the publicly available $bert-base-uncased$ model for MS-MARCO dataset, $bert-base-multilingual-uncased$ model for ESCI and PES datasets.

CAT-CL: is an adaptation of the approach proposed in [11]. While the original paper combines contrastive loss with a ranking loss, for consistency, we combine the contrastive loss with the BCE loss.

Table 1: Model performance of different models on PES dataset for ranking task.

Model	Train with	RERANK BE matchset		RERANK production matchset	
		MRR@10	R@100	MRR@10	R@25
BE	NA	5.97%	73.15%	9.93%	59.93%
CAT	BE	6.31%	80.82%	8.98%	55.55%
CAT-CL	BE	6.44%	81.09%	9.28%	55.45%
Bi-CAT	BE	7.23%	81.36%	9.92%	56.91%

Table 2: Performance of models on ESCI dataset for E-SCI and ESC-I tasks

Models	ESCI Dataset			
	E-SCI ROC_AUC	E-SCI F1	ESC-I ROC_AUC	ESC-I F1
BE	70.86%	75.07%	78.63%	85.16%
CAT	83.70%	81.35%	84.63%	94.41%
CAT-CL	83.56%	81.27%	83.85%	94.46%
Bi-CAT	83.95%	81.16%	84.06%	93.97%

Bi-CAT: This is the approach we introduce in this paper.

Note that we don't report numbers from trivial options such as linear combination of the BE and CAT scores as they generally are not feasible as the scores are of different range and distribution. This causes high degradation of model performance.

5.1.4 Experimental Setup. To train all our models, we use code leveraging PyTorch and Huggingface. We use AWS p4d instances with multi-gpu data parallel training. We use a sequence length of 128 for item, and 32 for query. Early stopping and hyperparameter tuning was performed using the validation dataset and relevant metrics for the task. AdamW optimizer with a learning rate between $1e-4$ and $1e-6$ was used.

5.2 Performance on In-distribution Data

To measure the performance of different approaches on in-distribution data we train and evaluate BE, CAT, CAT-CL, and Bi-CAT on all four tasks with the standard train/test dataset. For the Ranking tasks (MS MARCO and PES), we first train the BE model with in-batch negatives and contrastive loss for MS MARCO dataset, and ANCE [27] approach to augment hard negatives with in-batch negatives using a constrastive loss for the PES dataset. BE and BM25 models are used as retriever for CAT-based rankers. For both ESCI tasks we use in-batch negatives and given negatives to train the BE model. For ranking tasks, to train the CAT, CAT-CL, and Bi-CAT models we use negatives generated from the retriever model and to evaluate the in-distribution test data the positives are augmented with negatives from retriever output. For the ESCI tasks, the in-distribution test data contains both postives and negatives. Table 1, 2, and 3 show the 'in-distribution' performance of the models. As expected, we observe that the CAT-based rankers have the best performance. We also observe that the Bi-CAT model has similar performance as CAT ranker on all tasks. This shows that Bi-CAT models are atleast as good as CAT models answering our first question.

Table 3: Model performance of different models on MS MARCO dataset for ranking task.

Model	Trained with	RERANK BM25 output		RERANK BE output	
		MS MARCO		MS MARCO	
		MRR@10	R@100	MRR@10	R@100
BM25	NA	18.74%	67.01%	19.89%	73.46%
BE	NA	24.09%	75.55%	21.52%	75.21%
CAT	BM25	35.29%	82.08%	35.23%	85.72%
	BE	34.88%	81.39%	35.50%	85.71%
CAT-CL	bmtf	35.98%	82.25%	35.91%	86.02%
	be	36.52%	81.80%	37.22%	86.47%
Bi-CAT	BM25	34.53%	82.05%	34.63%	85.58%
	BE	35.69%	81.41%	36.28%	86.02%

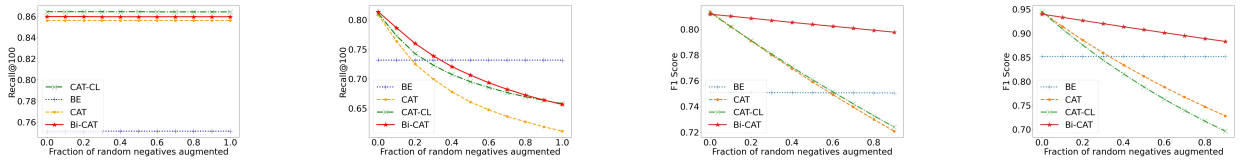
5.3 Performance of models on CIDS Data

To understand the performance of these models on CIDS data, we perform evaluation on two types of CIDS test data: 1) on MS MARCO and PES, we evaluate on a test set with negatives sampled from retriever different from the one used during training, 2) on all datasets we augment different types of negatives to the test dataset.

5.3.1 CIDS test set. For MS MARCO dataset we train two variants of CAT rankers, one with negatives from BM25 and another with negatives from BE. We evaluate model performance on BE dataset if the model was trained with BM25 and vice-versa. In Table 3 we observe that this makes very little difference in MS MARCO dataset, i.e., a model trained with BE has similar performance on BM25 and vice versa. This we believe is because MS-MARCO dataset has ≈ 1 truly relevant item per query which makes the retrieval set from both retrievers generate mostly random item in top 1000 leading to no CIDS in the dataset. To truly see the impact of CIDS we look at the PES dataset. Here, we train the model using BE output and evaluate model performance on the matchset provided by the production system. In Table 1 we observe that while evaluating in-distribution, the CAT, CAT-CL, and Bi-CAT models outperform BE significantly. However, when evaluated on the production matchset which is a CIDS dataset, there is a significant drop in performance and BE model has the best performance confirming the robustness of BE models to CIDS. We also observe that the drop in performance of the Bi-CAT model is lower than CAT showing that Bi-CAT is more robust to CIDS. This shows that CAT rankers while effective on test set, are also sensitive to CIDS and hence not robust in production.

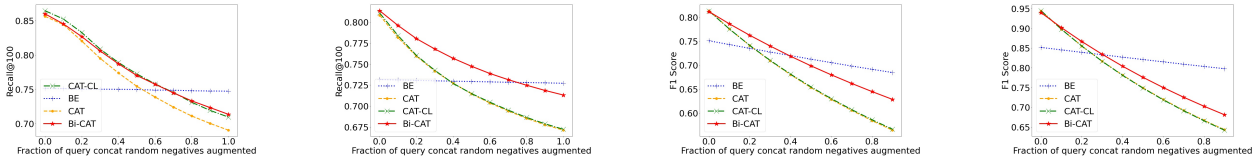
5.3.2 Induce CIDS through augmentation. To further understand the robustness of different models, we gradually augment two types of negatives to the dataset. 1) We augment random negatives (RN) which should be easy to identify for models, and 2) we choose a random item for a query and concatenate 50% of the query text to the beginning of the item text (QCRN). We augment from 10% to 100% of the dataset size. We do this process for all four tasks and plot a metric for all the four models.

Figure 2 shows the recall or F1 metric based on the task when RNs are augmented to the test dataset. We observe as expected in all tasks the BE model has almost no impact to such augmentation and the most robust to this CIDS. This is because, even though BE is trained with same data as all other models, training in-batch negatives makes almost any RN introduction almost become 'in-distribution'. Next, we observe that the Bi-CAT model is the second



(a) **MS-MARCO task.** (b) **PES task .** (c) **E-SCI task.** (d) **ESC-I task.**

Figure 2: Performance of all models on RN augmented CIDS test data.



(a) **MS-MARCO task.** (b) **PES task.** (c) **E-SCI task.** (d) **ESC-I task.**

Figure 3: Performance of all models on QCRN CIDS test data.

most robust model as it combines BE and CAT. While there is a degradation in performance in general, we do observe that the model is always better than CAT and sometimes worse than BE. We also observe that Bi-CAT is least robust on PES dataset with RN. We believe, this is because the dataset has more diverse item distribution compared to others making it hard to be robust without drop in 'in-distribution' CAT performance. CAT and CAT-CL have very similar performance throughout and is always affected by the CIDS the most. As mentioned earlier, we observe that in MS-MARCO dataset there is almost no drop with augmentation of RNs. This validates our previous hypothesis that the negative generated for this dataset are all likely RNs hence making the augmented dataset similar to the train data.

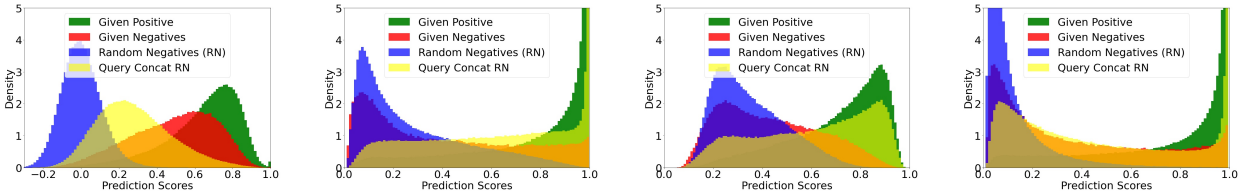
Figure 3 shows the performance of the models to harder negatives being injected. Since query text is partially embed in the title here, we observe that even the BE model drops performance significantly. However, still the BE models are the most robust to such injections and CAT rankers are the least. We again observe that Bi-CAT models are consistently more robust than CAT. On the MS-MARCO dataset, we observe a drastic drop in model performance for non-BE models compared to previous experiment. We believe this is because the augmentation of the query emulates a true CIDS scenario making the results resemble the results from other datasets. Overall, these experiments show that CAT rankers are significantly more vulnerable to CIDS than BE models and that Bi-CAT models improve the robustness of CAT models to CIDSanswering our questions two and three.

5.4 Understanding the Behavior of Models on CIDS Data

To answer our final question, we take a look at the score distribution of different models to different type of positive and negative examples. We use the E-SCI task to do this analysis. Figure 4 shows the distribution of the scores produced by all models on different segments of the data, i.e., given positives, given negatives, RNs, and QCRNs. We observe that for the BE model, score distribution of the given positives and negatives, while separate, have a large overlap. However, the score distribution of both RNs and QCRNs have clear separation from positives indicating the robustness of the model to such negatives. Next, for the CAT models, we observe that while there is a clear separation between given positives and negatives, the distribution of RNs have higher overlap compared to BE model and further the QCRNs have an almost uniform distribution indicating the poor robustness. Finally, on the Bi-CAT model we observe that the model inherits the goodness from both the models and the score distribution are clearly separated for given positives and negatives like the CAT model and for the RNs and QCRNs, we see the scores becoming more separate from positives as it makes use of the BE model. As mentioned earlier, we believe that the reason for BE to be more robust is that the model is both geometrically constrained and have an advantage with negatives and has learned from a lot more data which helps the model easily distinguish RNs and QCRNs. This also makes it hard for the BE models pay too much attention to individual tokens making it harder to separate given negative. On the flip side, CAT models are able to pay attention to all tokens making them more capable of separating given negatives from positives than BE. However, this comes at the cost of not being able to identify more diverse or easy negatives. Because Bi-CAT models take both these as input, they are able to make use of both signals to get the best of both worlds.

5.5 Abalation Study of Bi-CAT

In all of our experiments using Bi-CAT models, we only used the dot product score and kept the parameters of the encoders frozen. In this section, we will use the E-SCI task to perform some ablation study of trying different combination. We consider freezing encoders and using the full embedding in this ablation. Table 4 shows the F1 score of the models evaluated on test set and test set augmented with equal number of RNs. We observe that training the Bi-CAT model with frozen encoders and using dot product only performs best on both datasets. Whenever the encoders are unfrozen the model tends to loose the robustness that exists when encoders are frozen and using dot only. We believe this is because when using the dot only instead of embeddings and having encoders frozen, the model



(a) **BE model scores.** (b) **CAT model scores.** (c) **CAT-CL model scores.** (d) **Bi-CAT model scores.**

Figure 4: Score distribution of BE, CAT, and Bi-CAT models on E-SCI task for positives and negatives from test data and augmented RNs and QCRNs.

Table 4: Ablation study of Bi-CAT model on E-SCI task.

E-SCI Task					
Freeze BE	Freeze CAT	Use emb	Use dot	F1	
				Test	Test + 1x RNs
Y	Y	N	Y	81.16%	79.97%
N	Y	N	Y	80.03%	64.90%
Y	N	N	Y	81.26%	54.82%
N	N	N	Y	81.11%	54.49%
N	N	Y	N	81.29%	54.65%
Y	Y	Y	N	80.78%	54.31%

has lesser degrees of freedom and can hence learn faster with lesser data. The model can simply focus on relying on the dot-product score when it is hard to make a decision with the CAT embedding alone, and rely on CAT embedding for in-distribution data. On the other hand, when encoders are unfrozen or embeddings are used instead of dot, the *mlp* potentially requires a lot more data augmentation to attain robustness. We observe this further, when only the CAT encoder is unfrozen, we observe that the drop in F1 on the CIDS data is not as steep as the drop when both encoders are unfrozen.

6 CONCLUSION AND FUTUREWORK

In this paper, we aimed to address the issue of robustness in CAT models when used for ranking/classification tasks. We observed that the performance of CAT models drop significantly when evaluated on datasets where the answer (item) distribution given the question (query) is different from its training dataset. This highlights the model's inability to be robust to shifts in the conditional distribution. To overcome this issue, we proposed the Bi-CAT approach, which augments a BE model with a CAT model to significantly improve the robustness of the model. Our experiments on multiple datasets show that Bi-CAT models can maintain the superior performance of CAT while also improving robustness.

REFERENCES

[1] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report MSR-TR-2010-82.

[2] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List.

[3] Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA).*

[4] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. arXiv:2010.02666 [cs.IR]

[5] Rolf Jagerman, Zhen Qin, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2022. On Optimizing Top-K Metrics for Neural Ranking Models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[6] 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.

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

[8] Megan Leszczynski, Daniel Y Fu, Mayee F Chen, and Christopher Ré. 2022. TABi: Type-Aware Bi-Encoders for Open-Domain Entity Retrieval. *arXiv preprint arXiv:2204.08173* (2022).

[9] Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2021. Trans-Encoder: Unsupervised sentence-pair modelling through self-and mutual-distillations. *arXiv preprint arXiv:2109.13059* (2021).

[10] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. 3365–3375. https://doi.org/10.1145/3447548.3467149

[11] Xiaofei Ma, Cicero Nogueira dos Santos, and Andrew O Arnold. 2021. Contrastive fine-tuning improves robustness for neural rankers. *arXiv preprint arXiv:2105.12932* (2021).

[12] Aditya Menon, Sadeep Jayasumana, Ankit Singh Rawat, Seungyeon Kim, Sashank Reddi, and Sanjiv Kumar. 2022. In defense of dual-encoders for neural ranking. In *International Conference on Machine Learning.*

[13] Aashiq Muhamed, Sriram Srinivasan, Choon Hui Teo, Qingjun Cui, Belinda Zeng, Trishul Chilimbi, and S. V. N. Vishwanathan. 2023. Web-scale semantic product search with large language models. In *PAKDD 2023.* https://www.amazon.science/publications/web-scale-semantic-product-search-with-large-language-models

[14] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016.*

[15] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Ding Weitian (Allen), Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic Product Search.

[16] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[17] Jesus-German Ortiz-Barajas, Gemma Bel-Enguix, and Helena Gómez-Adorno. 2022. Sentence-CROBI: A Simple Cross-Bi-Encoder-Based Neural Network Architecture for Paraphrase Identification. *Mathematics* 10, 19 (2022), 3578.

[18] Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I.*

[19] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. (2022).

[20] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (apr 2009), 333–389.

[21] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488* (2021).

[22] Georgios Sidiropoulos and Evangelos Kanoulas. 2022. Analysing the Robustness of Dual Encoders for Dense Retrieval Against Misspellings. *arXiv preprint arXiv:2205.02303* (2022).

[23] Robin Swezey, Aditya Grover, Bruno Charron, and Stefano Ermon. 2020. PiRank: Scalable Learning To Rank via Differentiable Sorting. (2020).

[24] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240* (2020).

[25] Xuanhui Wang, Cheng Li, Nadav Golbandi, Mike Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *Proceedings of The 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*.

[26] Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[27] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[28] 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.

[29] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*. Association for Computational Linguistics, Online.