LEARNING LISTWISE DOMAIN-INVARIANT REPRESENTATIONS FOR RANKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain adaptation aims to transfer models trained on data-rich domains to low-resource ones, for which a popular method is invariant representation learning. While they have been studied extensively for classification and regression problems, how they would apply to ranking problems, where the metrics and data follow a list structure, is not well understood. Theoretically, we establish a generalization bound for ranking problems under metrics including MRR and NDCG, leading to a method based on learning listwise invariant feature representations. The main novelty of our results is that they are tailored to the listwise approach of learning to rank: the invariant representations our method learns are for each list of items as a whole, instead of the individual items they contain. Our method is evaluated on the passage reranking task, where we adapt neural text rankers trained on a general domain to various specialized domains.

1 INTRODUCTION

Learning to rank applies machine learning to solve ranking problems that are at the core of many everyday applications and products, including search engines and recommendation systems (Liu, 2009). With the availability of ever increasing amounts of training data, state-of-the-art performance on more and more ranking tasks are achieved by larger and larger models. A prominent example is text retrieval and ranking, where fine-tuned language models with billions of parameters easily outperform traditional ranking models (Nogueira et al., 2020). But the need for abundant data means that large neural models may not benefit tasks with little to no annotated data, where they could fare worse than baselines such as gradient boosted decision trees (Qin et al., 2021).

A popular technique for extending the benefits of large neural models is *domain adaptation*. It builds on *zero-shot learning:* where instead of directly optimizing for the task of interest with limited data, referred to as the target domain, the model is trained on a data-rich source domain with a similar underlying data distribution. Domain adaptation considers the scenario where (unlabeled) data from the target domain is available, which can be leveraged to estimate the domain shift and improve transferability, e.g. via learning invariant feature representations. This setting and its training methods have been actively studied for classification and regression problems (Ben-David et al., 2007; Ganin et al., 2016; Zhao et al., 2018). For ranking problems, however, existing studies are mostly limited to specific tasks and applications. In fact, due to the inherent list structure of the metrics and data, theoretical explorations of domain adaptation for ranking are only nascent.

To this end, in this paper we provide the first analysis of domain adaptation for listwise learning to rank via invariant representation learning (Section 3). Our result builds upon the foundational work by Ben-David et al. (2007) for domain adaptation in the binary classification setting. One of the insights from our theory is that, when the domain shift is small in terms of the Wasserstein distance, a ranking model optimized for the source is transferable to the target domain, whose performance under ranking metrics such as MRR and NDCG can be bounded.

Inspired by our theory, we propose an adversarial training method for learning domain-invariant representations, called ListDA, which aims at minimizing the source and target distributional shifts in the feature space and thereby improving generalization over the target domain. Different from the traditional classification and regression settings where each input could be modeled as a fixed dimensional feature vector, in ranking, each input consists of a list of items. Technically, the main novelty of ListDA is that it learns invariant representations of each *list* as a whole instead of the

individual *items* they contain, as approaches for invariant *item* representations are not necessarily suitable for *listwise* learning to rank.[1] We evaluate ListDA on unsupervised domain adaptation for passage reranking, a fundamental research task in the field of information retrieval (Craswell et al., 2019), where the goal is to rerank a list of candidate documents retrieved by a first-stage retrieval model in response to a search query (Section 5). We adapt T5 neural rerankers (Raffel et al., 2020) fine-tuned on the general domain MS MARCO dataset (Bajaj et al., 2018) to two specialized domains: biomedical and news articles. Our results demonstrate the benefits of invariant representations on the generalization of rankers trained with ListDA. In particular, the key to the improved performance is the learning of listwise representations.

2 PRELIMINARIES

Learning to Rank. A ranking problem is defined by a joint distribution over lists[2] $X \in \mathcal{X}$ of items and their non-negative relevance scores $Y = (Y_1, \cdots, Y_\ell) \in \mathbb{R}^\ell_{\geq 0}$, where all lists are length-ℓ. Furthermore, we assume that the ground-truth scores are a function of the lists, $y(X)$, so that we equivalently define ranking problems by a distribution μ^X of lists X along with the scoring function $y : \mathcal{X} \to \mathbb{R}^\ell_{\geq 0}$.

The goal is to train a *ranker* $f : \mathcal{X} \to S_\ell$ that maps each list $x \in \mathcal{X}$ to rank assignments $r := f(x) \in S_\ell$, where r_i represents the predicted rank of item i and S_ℓ denotes the set of permutations on $[\ell] := \{1, 2, \cdots, \ell\}$, such that r recovers the descending ordering of the relevance scores y_i: $y_i > y_j \iff r_i < r_j$ for all $i \neq j$. The more common setup is to train a scoring function $h : \mathcal{X} \to \mathbb{R}^\ell$ whose output is a list of ranking scores, s.t. $s := h(x)$ correlates with y and their ordering follows that of the ground-truth scores. Besides taking the descending ordering, rankings of the items could also be obtained from the ranking scores with a probabilistic model, as will be used in Section 3.

The quality of the predicted ranks is measured by ranking metrics $u : S_\ell \times \mathbb{R}^\ell_{\geq 0} \to \mathbb{R}_{\geq 0}$, which are functions that take as inputs the ranking along with the ground-truth relevance scores of the list and output a non-negative utility score. Two popular metrics in information retrieval include reciprocal rank and normalized discounted cumulative gain:

Definition 1 (Voorhees, 1999)**.** *Suppose the ground-truth relevance scores $y \in \{0, 1\}^\ell$ are binary, then the reciprocal rank* (RR) *of the rank assignments $r \in S_\ell$ is*

$$\mathrm{RR}(r, y) := \max\{r_i^{-1} : i \in [\ell], y_i = 1\} \cup \{0\}.$$

Mean reciprocal rank (MRR) is the expectation of RR over the dataset, $\mathbb{E}_{X \sim \mu^X}[\mathrm{RR}(f(X), y(X))]$.

Definition 2 (Järvelin & Kekäläinen, 2002)**.** *The discounted cumulative gain* (DCG) *and the normalized* DCG *of the rank assignments $r \in S_\ell$ are*[3]

$$\mathrm{DCG}(r, y) := \sum_{i \in [\ell]} \frac{y_i}{\log(r_i + 1)} \quad \textit{and} \quad \mathrm{NDCG}(r, y) := \frac{\mathrm{DCG}(r, y)}{\mathrm{IDCG}(y)},$$

where the ideal DCG (IDCG) *is defined as the maximum* DCG *value for fixed y,* $\mathrm{IDCG}(y) := \max_{r' \in S_\ell} \mathrm{DCG}(r', y)$*, attained by the descending ordering of the y_i's.*

Domain Adaptation. This work considers the setting where there is a source and a target domain, $(\mu^X_S, y_S), (\mu^X_T, y_T)$, and the goal is to train a good scoring function for the target domain. When domain shift is small, i.e. $\mu^X_S \approx \mu^T_S$ and $y_S \approx y_T$, scorers trained on the source are expected to be transferable to the target without the explicit need of labeled target examples. In fact, in such

[1] Throughout this paper, by *listwise approach* to learning to rank we mean that the ranker is trained using listwise ranking losses (Cao et al., 2007; Xia et al., 2008), not that the ranker can necessarily model the interaction between items (Pang et al., 2020).

[2] A common choice for the space of lists $\mathcal{X}$ is the ℓ-times Cartesian product of $\mathbb{R}^d$, meaning that each list $x = (x_1, \cdots, x_\ell)$ is a concatenation of items represented by $\mathbb{R}^d$ feature vectors. But more generally and abstractly the lists need not be concatenations; $\mathcal{X}$ can also be $\mathbb{R}^k$ (not scaling with ℓ) provided a mechanism to represent lists by fixed-length vectors.

[3] W.l.o.g. the gain function in DCG is set to the identity map in this work.

cases, the target performance can be bounded by the source performance. As an example, for binary classification with Bernoulli models, we have the following generalization bound:

Theorem 1 (Shen et al., 2018)**.** *Let the source and target domain binary classification problems be given by joint distributions μ_S, μ_T over inputs and labels $(X, Y) \in \mathcal{X} \times \{0, 1\}$. Let $\mathcal{F} \subset [0, 1]^{\mathcal{X}}$ be a class of L-Lipschitz predictors, and define the error rate of $f \in \mathcal{F}$ on the source domain by*

$$\mathcal{E}_S(f) := \mathbb{E}_{(X,Y)\sim\mu_S}[\mathbb{1}(Y \neq \hat{Y})] := \mathbb{E}_{(X,Y)\sim\mu_S}[f(X) \cdot \mathbb{1}(Y \neq 1) + (1 - f(X)) \cdot \mathbb{1}(Y \neq 0)],$$

and $\mathcal{E}_T$ analogously, where $\mathbb{P}(\hat{Y} = 1 \mid X = x) = f(x)$. Define $\lambda^* := \min_{f'}(\mathcal{E}_S(f') + \mathcal{E}_T(f'))$, then for all $f \in \mathcal{F}$,

$$\mathcal{E}_T(f) \leq \mathcal{E}_S(f) + 2L \cdot W_1(\mu_S^X, \mu_T^X) + \lambda^*,$$

where μ^X denotes the marginal distribution of X.

In Theorem 1, the domain shift is measured by the Wasserstein-1 distance between source and target input distributions; its Kantorovich-Rubinstein dual formulation is stated below (Edwards, 2011):

Definition 3 (Wasserstein-1)**.** *Let $(X, d_{\mathcal{X}})$ be a metric space, and μ, ν be probability measures on $\mathcal{X}$. Their Wasserstein-1 distance is $W_1(\mu, \nu) := \sup_{f\in\text{Lip}(1)}(\int_{\mathcal{X}} f(x)\,\mathrm{d}\mu(x) - \int_{\mathcal{X}} f(x)\,\mathrm{d}\nu(x))$.*

Where the supremum is taken over 1-Lipschitz functionals $f : \mathcal{X} \to \mathbb{R}$:

Definition 4 (Lipschitz)**.** *Let $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{X}', d_{\mathcal{X}'})$ be metric spaces. A function $f : \mathcal{X} \to \mathcal{X}'$ is L-Lipschitz if $d_{\mathcal{X}'}(f(x_1), f(x_2)) \leq L d_{\mathcal{X}}(x_1, x_2)$ for all $x_1, x_2 \in \mathcal{X}$, which is denoted by $f \in \text{Lip}(L)$.*

Domain adaptation generalization bounds under the pointwise and pairwise approaches to ranking could be derived from Theorem 1, since they cast the ranking problem to one of binary classification. But the result would be too loose to give a statement on the learned ranker w.r.t. ranking metrics such as MRR or NDCG, which are defined on lists.

3 Domain Adaptation Generalization Bound for Ranking

In this section we establish a domain adaptation generalization bound for listwise learning to rank akin to Theorem 1. We consider the setting of learning representations in addition to scoring functions, so that our end-to-end scorer is the composition $h \circ g$ of a feature mapping $g : \mathcal{X} \to \mathcal{Z}$ and a scoring function $h : \mathcal{Z} \to \mathbb{R}^\ell$ on the learned list representations. For instance, we could train an m-layer neural network and treat the first $(m - 1)$ layers as g and the last as h.

For our bound, the rank assignments $r \in S_\ell$ are obtained from the output scores $s := h \circ g(x)$ in a probabilistic manner: by sampling from the *Plackett-Luce model* with the exponentiated scores $\exp(s_i)$ as its parameters (Cao et al., 2007; Guiver & Snelson, 2009).

Definition 5 (Plackett, 1975; Luce, 1959)**.** *A Plackett-Luce model with parameters $v \in \mathbb{R}^\ell_{>0}$ defines a distribution over S_ℓ, whose probability mass function is denoted by p_v and given for all $r \in S_\ell$ by*

$$p_v(r) = \prod_{i=1}^{\ell} \frac{v_{I(r)_i}}{\sum_{j=i}^{\ell} v_{I(r)_j}},$$

where $I(r)_i$ is the index of the item ranked at i, i.e. $r_{I(r)_i} = i, \forall i$.

For a ranking metric u, the source domain performance of a scorer $h \circ g$ is evaluated via

$$\mathcal{E}_S(h \circ g) := \mathbb{E}_{X\sim\mu_S^X}\left[\max_{r\in S_\ell} u(r, y_S(X)) - \mathbb{E}_{R\sim p_{\exp(h\circ g(X))}}[u(R, y_S(X))]\right],$$

which computes its suboptimality relative to the maximum utility. $\mathcal{E}_T$ is defined analogously.

Finally, we make the following Lipschitz assumptions for our result.

Assumption 1. *The ranking metric $u : S_\ell \times \mathbb{R}^\ell_{\geq 0} \to \mathbb{R}_{\geq 0}$ is bounded by B and is L_u-Lipschitz w.r.t. its second input, the ground-truth relevance scores $y \in \mathbb{R}^\ell_{\geq 0}$, under Euclidean distance.*

Assumption 2. *The ground-truth scoring functions $y_S, y_T : \mathcal{X} \to \mathbb{R}^\ell$ are L_y-Lipschitz under Euclidean distance on the output space $\mathbb{R}^\ell_{\geq 0}$.*

Assumption 2 is satisfied when $\mathcal{X}$ is discrete and the scores are bounded (this argument is used in the proof of Corollary 3). As an example, the input to most neural language models is a sequence of one-hot encodings of the vocabulary.

Assumption 3. *The spaces of input lists $\mathcal{X}$ and feature representations $\mathcal{Z}$ are metric spaces. The function class $\mathcal{H}$ of the scoring functions h is L_h-Lipschitz under Euclidean distance on the output space $\mathbb{R}^\ell$, and that of the feature mappings, $\mathcal{G}$, is such that $\forall g \in \mathcal{G}$, the restrictions of g to the supports of μ_S^X and μ_T^X are injective with L_g-Lipschitz inverses, $g^{-1}|_{g(\mathrm{supp}(\mu_S^X))}, g^{-1}|_{g(\mathrm{supp}(\mu_T^X))}$.*

When both h and g are neural networks, Lipschitzness can be enforced via ℓ_2-regularization. This assumption in fact underlies most neural network generalization and complexity analyses (Anthony & Bartlett, 1999; Bartlett et al., 2017). Injection is a technical assumption that we argue is reasonable on g, because the feature representations should retain as much information from the inputs of each domain as possible, whereas injection and Lipschitzness of the inverse are violated when different inputs are mapped to the same point in $\mathcal{Z}$, and causing information loss.

With the above definitions and assumptions in place, we are now ready to state our generalization bound for learning to rank.

Theorem 2. *Under Assumptions 1 to 3, for any $g \in \mathcal{G}$, define $\lambda_g^* := \min_{h'}(\mathcal{E}_S(h' \circ g) + \mathcal{E}_T(h' \circ g))$, then for all $h \in \mathcal{H}$,*

$$\mathcal{E}_T(h \circ g) \leq \mathcal{E}_S(h \circ g) + 2(2L_u L_y L_g + BL_h\sqrt{\ell}) \cdot W_1(\mu_S^Z, \mu_T^Z) + \lambda_g^*,$$

where μ_S^Z denotes the distribution of the learned source features, $\mu_S^Z(z) := \mu_S^X(g^{-1}(z))$, and μ_T^Z analogously.

The bound suggests that, if the feature representations of $g : \mathcal{X} \to \mathcal{Z}$ are domain-invariant, meaning $\mu_S^Z = \mu_T^Z$, and the optimal joint error λ_g^* remains low, then a scorer $h : \mathcal{Z} \to \mathbb{R}^\ell$ optimized for the source will also perform well on the target. Note that in order for λ_g^* to remain small, then the feature mapping g should also preserve the necessary information and learn the correct correspondence between source and target domains for recovering the ground-truth rankings on both domains. This forms the basis of the majority of domain adaptation approaches based on invariant representation learning, including ours, whichs use labeled source and unlabeled target data to minimize source error $\mathcal{E}_S$ and align the feature distributions μ_S^Z, μ_T^Z for lowering their W_1 distance. Generally without labeled target data, λ_g^* is not theoretically guaranteed to remain low, but it does not prevent the empirical success of these methods in many applications, from vision (Zhao et al., 2022) to language (Ramponi & Plank, 2020). The proof is deferred to Appendix A.

Proof Sketch. The main idea of the proof is that under our setup and assumptions we could write $\mathcal{E}_S$ and $\mathcal{E}_T$ as expectations of Lipschitz functions of $Z \sim \mu_S^Z$ and μ_T^Z, respectively, so their difference can be upper bounded by the W_1 distance between μ_S^Z, μ_T^Z by Definition 3. With a simple modification of the proof, the result extends to the cutoff version of the ranking metric u. A final remark is that a finite sample generalization bound could be obtained from Theorem 2 using existing results on Rademacher complexities, due to the Lipschitzness of the feature mapping as well as the scoring function (Blitzer et al., 2008; Shalev-Shwartz & Ben-David, 2014).

To instantiate our bound to MRR and NDCG, we verify their Lipschitzness:

Corollary 3 (Bound for MRR)**.** RR *is* 1*-Lipschitz in y, thereby*

$$\mathbb{E}_T[\mathrm{RR}(h \circ g)] \geq \mathbb{E}_S[\mathrm{RR}(h \circ g)] - 2(2L_y L_g + L_h\sqrt{\ell}) \cdot W_1(\mu_S^Z, \mu_T^Z) - \lambda_g^*,$$

where for brevity we wrote $\mathbb{E}[\mathrm{RR}(h \circ g)] := \mathbb{E}_{X \sim \mu^X, R \sim p_{\exp(h \circ g(X))}}[\mathrm{RR}(R, y(X))]$.

Corollary 4 (Bound for NDCG)**.** *Suppose $U_{\min} \leq \mathrm{IDCG}(y) \leq U_{\max}$ for some $U_{\min}, U_{\max} \in (0, \infty)$ and all $y \in y_S(\mathrm{supp}(\mu_S^X)) \cup y_T(\mathrm{supp}(\mu_T^X))$, then* NDCG *is $\widetilde{O}(\sqrt{\ell})$-Lipschitz in y, thereby*

$$\mathbb{E}_T[\mathrm{NDCG}(h \circ g)] \geq \mathbb{E}_S[\mathrm{NDCG}(h \circ g)] - \widetilde{O}(\sqrt{\ell}(L_y L_g + L_h)) \cdot W_1(\mu_S^Z, \mu_T^Z) - \lambda_g^*.$$

where for brevity we wrote $\mathbb{E}[\mathrm{NDCG}(h \circ g)] := \mathbb{E}_{X \sim \mu^X, R \sim p_{\exp(h \circ g(X))}}[\mathrm{NDCG}(R, y(X))]$.

Because the last two terms on the r.h.s. are nonnegative, target domain MRR and NDCG are improved by maximizing the source performance of h and minimizing W_1 via learning a feature mapping g that is both invariant and informative.

4 LEARNING LISTWISE DOMAIN-INVARIANT REPRESENTATIONS

Inspired by Theorem 2, we propose an adversarial training method for learning domain-invariant representations of list. Specifically, we consider the setup where each feature representation $z := g(x)$ of the input list of ℓ items, $x \in \mathcal{X}$, is the concatenation of ℓ feature vectors, i.e. $\mathcal{Z} = \mathbb{R}^{\ell \times d}$, each $z = (z_1, \cdots, z_\ell)$, and $z_i \in \mathbb{R}^d$ is the learned feature vector of the i-th item in the list. This is standard in many learning to rank applications, e.g. in neural text ranking, each feature vector is an embedding of the input text computed by a language model (Guo et al., 2020).

We want to learn a feature mapping $g \in \mathcal{G}$ that minimizes the distributional shifts between source and target on the feature space under some probability metric, $D(\mu_S^Z, \mu_T^Z)$, for which a well-known technique is adversarial training (Goodfellow et al., 2014; Arjovsky et al., 2017). It solves a minimax problem of two players: the feature mapping g, and the domain discriminator $f_{\text{ad}} \in \mathcal{F}_{\text{ad}}$, $\mathcal{F}_{\text{ad}} \subset [0,1]^{\mathcal{Z}}$, taking as input a feature sample $z := g(x)$ of either the source or target domain and predicting its domain identity, $\hat{a} := f_{\text{ad}}(z) = f_{\text{ad}} \circ g(x)$. The minimax objective is defined with an adversarial loss function $\ell_{\text{ad}} : [0,1] \times \{0,1\} \to \mathbb{R}$ that takes as inputs the prediction $\hat{a}$ along with the true domain identity a (w.l.o.g. $a = 1$ for target):

$$\mathcal{L}_{\text{ad}}(g, f_{\text{ad}}) := \mathbb{E}_{x \sim \mu_S^X}[\ell_{\text{ad}}(f_{\text{ad}} \circ g(x), 0)] + \mathbb{E}_{x \sim \mu_T^X}[\ell_{\text{ad}}(f_{\text{ad}} \circ g(x), 1)], \tag{1}$$

and the optimization is w.r.t. $\min_g \max_{f_{\text{ad}}} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}})$.

With the 0-1 loss, $\ell_{\text{ad}}(\hat{a}, a) = -a \cdot \mathbb{1}(\hat{a} < \frac{1}{2}) - (1-a) \cdot \mathbb{1}(\hat{a} \geq \frac{1}{2})$, the adversarial loss becomes the (total) classification accuracy of f_{ad} on the task of predicting the domain identity, $\mathcal{L}_{\text{ad}}(g, f_{\text{ad}}) = \mathbb{P}_{x \sim \mu_S^X}(f_{\text{ad}} \circ g(x) < \frac{1}{2}) + \mathbb{P}_{x \sim \mu_T^X}(f_{\text{ad}} \circ g(x) \geq \frac{1}{2})$. Then the goal of the discriminator is to distinguish the features output by g, and that of the feature mapping is to fool the discriminator by learning invariant representations. Indeed, under optimality of f_{ad}, $\mathcal{L}_{\text{ad}}$ upper bounds $W_1(\mu_S^Z, \mu_T^Z)$:

Proposition 5. *Let $\|\cdot\|$ be a metric on $\mathbb{R}^{\ell \times d}$, define $B := \sup_{z \in \text{supp}(\mu_S^Z), z' \in \text{supp}(\mu_T^Z)} \|z - z'\|$. With $\ell_{\text{ad}}(\hat{a}, a) = -a \cdot \mathbb{1}(\hat{a} < \frac{1}{2}) - (1-a) \cdot \mathbb{1}(\hat{a} \geq \frac{1}{2})$, we have $W_1(\mu_S^Z, \mu_T^Z) \leq B(\max_{f_{\text{ad}}} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}}) - 1)$.*

In practice, to train f_{ad} for minimizing the classification error, the 0-1 loss is replaced by a surrogate loss, and for which we use the cross-entropy loss in our experiments,

$$\ell_{\text{ad}}(\hat{a}, a) = a \log(\hat{a}) + (1-a) \log(1 - \hat{a}). \tag{2}$$

And by setting $\mathcal{F}_{\text{ad}}$ to a parameterized function class e.g. neural networks, the minimax problem can be solved with gradient descent-ascent (w.r.t. g and f respectively). This is typically implemented with a gradient reversal layer on g (Ganin et al., 2016). To prevent g from converging to trivial solutions like $x \mapsto 0$ that cause information loss in the learned features and increase the minimum achievable $\mathcal{E}_S$ and therefore λ_g^*, g is optimized together with the scorer h under a joint objective

$$\mathcal{L}_{\text{joint}}(h, g) = \min_{h \in \mathcal{H}, g \in \mathcal{G}} \left(\mathcal{L}_{\text{rank}}(h \circ g) + \lambda \max_{f_{\text{ad}} \in \mathcal{F}_{\text{ad}}} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}}) \right), \tag{3}$$

where $\mathcal{L}_{\text{rank}}$ is the ranking loss of choice (a surrogate to the ranking metric), and the hyperparameter $\lambda \geq 0$ controls the strength of domain-invariant feature learning.

Choice of Discriminator Function Class. The final missing piece is the choice of $\mathcal{F}_{\text{ad}}$ that can model lists $z = (z_1, \cdots, z_\ell)$ of feature vectors $z_i \in \mathbb{R}^d$ and is continuously differentiable. A naïve design choice would be to flatten the list into a single ℓd-dimensional vector and set $\mathcal{F}_{\text{ad}}$ to dense neural networks. But note that the output of $f_{\text{ad}} \in \mathcal{F}_{\text{ad}}$ may change if the items in z are swapped, despite this does not alter the list as far as ranking is concerned, so the capacity of f_{ad} is wasted on modeling the permutation invariance property of the inputs z. To avoid this waste, one could attempt at replacing flattening with an information-preserving permutation-invariant operation, but such an operation that is also continuously differentiable is nontrivial to handcraft.

Therefore, we propose using transformers (no positional encoding) with mean-pooling as our discriminator function class $\mathcal{F}_{\text{ad}}$ (Vaswani et al., 2017), which are permutation-invariant, continuously differentiable, and expressive for their success in many language tasks. We refer to our approach as ListDA, for which a block diagram is in Fig. 1.

As a final remark, note that aligning the distributions of $\mathbb{R}^{\ell \times d}$ **lists** of feature vectors, which ListDA performs, is not the same as those of the individual $\mathbb{R}^d$ feature vectors (**items**), or ItemDA: $\mu_S^{U(Z)} = \mu_T^{U(Z)}$ where $\mu_S^{U(Z)}(z_i) := \mathbb{P}_{Z \sim \mu_S^Z}(z_i \in Z)$. Concretely, $\mu_S^{U(Z)} = \mu_T^{U(Z)} \not\Longrightarrow \mu_S^Z = \mu_T^Z$.

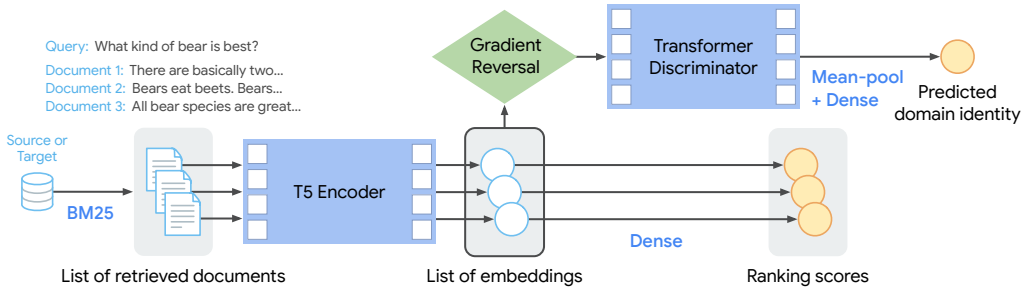


Figure 1: Block diagram of ListDA instantiated on the T5 neural ranker for text ranking.

5 Experiments on Passage Reranking

We evaluate ListDA on the passage[4] reranking task, where the goal is to rank candidate passages in a retrieved set based on their relevance to a given text query q. The candidate set of ℓ passages is usually retrieved from the corpus using an efficient model, such as sparse retrievers including BM25 (Robertson & Zaragoza, 2009), or dense retrievers e.g. DPR (Karpukhin et al., 2020). Then a more accurate but expensive ranking model is applied to refine and improve the ranks. Currently, SOTA performance is achieved by cross-attention rankers based on large language models.

Training neural rerankers requires a large amount of queries and document-relevance annotations. While such data can be obtained from search engines under weak supervision for general domain text retrieval, annotations on specialized domains such as scientific literature are costly. However, unannotated documents are almost always readily available regardless of domain, making them a suitable candidate for applying unsupervised domain adaptation: specialized rerankers are obtained via adapting ones that are trained on general domain annotated data.

Models. For our ranking model, we use BM25 as the first-stage retriever for simplicity (details in Appendix C) and focus on the adaptation of the reranker, which is fine-tuned from the T5 1.1 Base checkpoint with 250 million parameters. Given candidate documents $d_1, \cdots, d_\ell$ for a query q, the input list is formed by concatenating each document with the query (and the title if available), $x = ([q, d_1], [q, d_2], \cdots, [q, d_\ell])$. We treat the T5 encoder as the feature mapping g, and set the listwise feature representation to the list of first-token output embeddings that g computes on each query-document (q-d) pair,[5] $z = g(x) = (g(x_1), \cdots, g(x_\ell)) \in \mathbb{R}^{\ell \times 1024}$. Ranking scores are then projected from each q-d embedding by a dense layer, $s_i = h(z_i)$. We use the *listwise* softmax cross-entropy loss for model training, as in (Jagerman et al., 2022b):

$$\ell_{\text{rank}}(s, y) = -\sum_{i=1}^{\ell} y_i \log\left(\frac{\exp(s_i)}{\sum_{j\in[\ell]} \exp(s_j)}\right). \tag{4}$$

The setup for adversarial training follows Section 4. The domain discriminator f_{ad} consists of three transformer blocks with the same architecture as those in the T5 encoder. The prediction of domain identity given a list feature $z = (z_1, \cdots, z_\ell)$ is the projection of the mean-pool of f_{ad}'s output embeddings on z as a sequence, followed by sigmoid. To minimize the sensitivity of f_{ad} to random initializations, we use an ensemble of five discriminators as in (Elazar & Goldberg, 2018).

The joint objective for ListDA is obtained from combining Eqs. (1) to (4), except that the max in Eq. (3) is taken over the ensemble of discriminators $f_{\text{ad}}^{(1)}, \cdots, f_{\text{ad}}^{(5)}$ w.r.t. the aggregated adversarial loss of $\sum_{i=1}^{5} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}}^{(i)})$. See Fig. 1 for a diagram, and Appendix C for hyperparameter settings.

Datasets. The source domain of our experiments is the MS MARCO dataset for passage ranking, a large-scale public dataset consisting of 8 million passages from the web covering a wide range

[4]The terms *document, text,* and *passage* are used interchangeably in this paper.

[5]Here, the feature vectors are computed from each item independently, but generally and ideally g would also model the interaction between items in the same list (Pang et al., 2020).

of topics and 532,761 pairs of search queries and relevant passages (Bajaj et al., 2018). The target domains are biomedical (TREC-COVID and BioASQ) and news articles (Robust04) (Voorhees et al., 2021; Tsatsaronis et al., 2015; Voorhees, 2005). The data are collected and preprocessed as in the BEIR benchmark (Thakur et al., 2021), whose paper also includes the statistics of the datasets. Details on the preparation of training lists, including the negative sampling procedure for irrelevant q-d pairs, are in Appendix C.

Since not all target datasets contain training queries, and neither do most unannotated domains in the wild, we discard the real training queries they contain and instead synthesize queries in a zero-shot manner using a query generator (Ma et al., 2021). With MS MARCO source domain relevant q-d pairs, the natural language generation model (QGen) is fine-tuned from the T5 1.1 XL checkpoint under the sequence-to-sequence task of generating the query given the relevant document as input, prepended with the prompt "`Generate question >>>`" and the title if available. To synthesize queries on the target, we feed QGen with documents in the target corpus and treat its outputs as queries to which the input documents are likely relevant. See Table 5 for samples of QGen q-d pairs.

Baselines and Methods. ListDA is compared to two baseline methods, zero-shot and QGen PL. In **zero-shot** learning, the reranker is trained on MS MARCO source domain only and directly evaluated on the targets. In **QGen PL**, we treat the QGen q-d pairs synthesized on the target domain as relevant pairs (PL as in these q-d pairs being "pseudolabeled" by QGen), and train the reranker on them in addition to MS MARCO q-d pairs. This method underlies several recent work on domain adaptation of text retrievers and rankers (Ma et al., 2021; Sun et al., 2021; Wang et al., 2022).

To illustrate that learning invariant representations of individual items is not suitable for listwise learning to rank as discussed in Section 4, i.e. aiming for $\mu_S^{U(Z)} = \mu_T^{U(Z)}$, we perform a set of **ItemDA** experiments, where the transformer domain discriminator is replaced by a three-layer MLP (no improvements from going larger) that takes as inputs individual q-d embeddings instead of entire lists of items (same as the DANN model by Ganin et al. (2016) in effect). This is the approach taken by prior work based on invariant representation learning (Cohen et al., 2018; Tran et al., 2019; Xin et al., 2022). Although ItemDA is appropriate for *pointwise* learning to rank, they are not included here because pointwise ranking losses are typically less performant than listwise.

5.1 Results

The results are presented in Table 1. For evaluation, the items are ranked in descending order of the ranking scores, and the metrics include ones that are commonly reported in the literature (e.g. TREC-COVID uses NDCG@20). Because TREC-COVID and Robust04 are annotated with 3-level relevancy, they are binarized for mean average precision (MAP) and MRR as follows: for TREC-COVID, 0 (not relevant) and 1 (partially relevant) are negative, and 2 (fully relevant) is positive; for Robust04, 0 (not relevant) is negative, and 1 (relevant) and 2 (highly relevant) are positive.

Across all datasets, the best performing reranker is trained with ListDA, and the fact that it shares the same training resource with QGen PL demonstrates the benefits of invariant representations. Furthermore, the favorable comparison of ListDA with ItemDA confirms our earlier analysis and discussion that under the listwise approach, the invariant representations the model learns should be of each list as a whole, on which the ranking loss ℓ_{rank} is computed, and not of the item individually.

Quality of QGen. An explanation for why QGen PL underperforms ListDA despite them sharing the same resources is that the random sampling of documents for gathering irrelevant q-d pairs could lead to false negatives being sampled and included in the training lists (Appendix C). In particular, Sun et al. (2021) observed that the queries synthesized by QGen lack specificity and could therefore have many relevant documents. While QGen pseudolabels are treated as ground-truth for supervised training under QGen PL, they are not assumed by ListDA and hence is less affected by the false negatives or false positives, as some synthesized queries may also not be entirely relevant. While out of scope in the present work, improving the query generation procedure could boost the performance of QGen PL as well as ListDA (Sun et al., 2021).

Table 1: Reranker performance on ranking the top 1000 documents retrieved by BM25.

Dataset	Method	MAP	MRR@10	NDCG@5	NDCG@10	NDCG@20
Robust04	BM25	0.2282	0.6801	0.4396	0.4088	0.3781
	Zero-shot	0.2759	0.7977†	0.5857†	0.5340†	0.4856†
	QGen PL	0.2693	0.7644	0.5406	0.5034	0.4694
	ItemDA	0.2822*†	0.8037†	0.5822†	0.5396†	0.4922†
	ListDA	**0.2901***†‡	**0.8234***†	**0.5979**†‡	**0.5573***†‡	**0.5126***†‡
TREC-COVID	BM25	0.2485	0.8396	0.7163	0.6559	0.6236
	Zero-shot	0.3083	0.9217	0.8328	0.8200	0.7826
	QGen PL	0.3180*‡	0.8907	0.8373	0.8118	0.7861
	ItemDA	0.3087	0.9080	0.8276	0.8142	0.7697
	ListDA	**0.3187***‡	**0.9335**	**0.8693***‡	**0.8412**†‡	**0.7985**‡
BioASQ	BM25	0.4088	0.5612	0.4580	0.4653	0.4857
	Zero-shot	0.5008‡	0.6465	0.5484‡	0.5542‡	0.5796‡
	QGen PL	0.5143*‡	0.6551	0.5538‡	0.5643‡	0.5915*‡
	ItemDA	0.4781	0.6383	0.5315	0.5343	0.5604
	ListDA	**0.5191***‡	**0.6666***‡	**0.5639***‡	**0.5714***‡	**0.5985***‡

*Improves upon zero-shot baseline with statistical significance ($p \leq 0.05$) under the two-tailed Student's t-test. †Improves upon QGen PL. ‡Improves upon ItemDA.

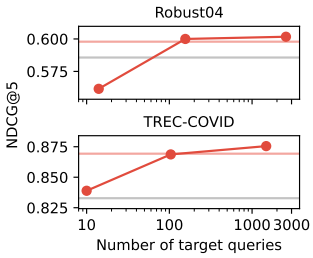
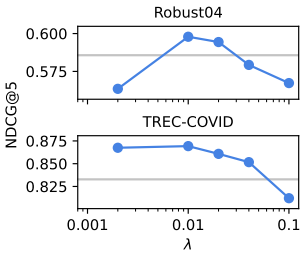


Figure 2: ListDA under different target sizes. Lower grey horizontal line is zero-shot, upper red line is ListDA using all QGen queries.



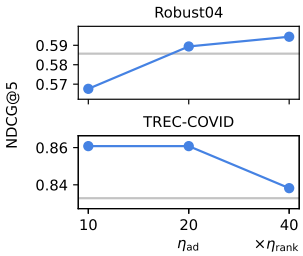


Figure 3: ListDA under different hyperparameter settings of λ and η_{ad}. Grey horizontal line is zero-shot. On the left, $\eta_{\text{ad}} = 0.004$ is fixed and λ varies. On the right, $\lambda = 0.02$ is fixed and η_{ad} varies.

5.2 Analysis of ListDA

Size of Target Data. Unsupervised domain adaptation assumes access to sufficient unlabeled target data for domain shift estimation, but not all domains have the same amount of resource. BioASQ contains 14 million documents (and thereby QGen queries), but Robust04 has 528,155, and TREC-COVID only 171,332. Therefore, we study how target data size affects ListDA on Robust04 and TREC-COVID by reducing the number of target QGen queries and thereby that of target documents for invariant feature learning (we retrieve 1,000 documents for each query using BM25, the majority of which are likely irrelevant). The results are shown in Fig. 2.

Surprisingly, using only around 100 target QGen queries (0.03% and 0.06% of all, respectively) is sufficient for ListDA to achieve full performance on both domains! Although the number of pseudolabeled q-d pairs is reduced, the size of target document is still substantial—up to 100,000, or 29.5% and 60.7% of the entire corpus, although most of them are irrelevant to the synthetic queries. So we hypothesize that the reason for this phenomenon, and the primary source of the performance gain brought by ListDA under our training setup (Appendix C), is the feature alignment of irrelevant q-d pairs in the lists. Indeed, when reduced to around 10 queries so that the number of documents is at most 10,000 (2.7% and 5.8%, respectively), ListDA performance begins to drop. Therefore, a direction for future empirical explorations is to apply ListDA on lists that contain more than one (pseudolabeled) relevant q-d pair (most pipelines for neural text rankers, including ours, uses only one), under which it may exhibit a different (and perhaps better) behavior.

Sensitivity to Hyperparameters. ListDA introduces two new hyperparameters for the domain discriminators f_{ad}: the strength of invariant feature learning $\lambda > 0$, and the discriminator learning rate η_{ad}. We study the sensitivity to their settings on Robust04 and TREC-COVID by fixing one and varying the other; the results are shown in Fig. 3. It is observed that the choice of λ is fairly important in eliciting the best performance from ListDA, but the same choice could work well across datasets because of the consistency of the trends. We set η_{ad} to be multiples of the reranker learning rate η_{rank}. The results show that while η_{ad} is more tolerant to misspecifications, the discriminators prefer different settings on each target dataset, probably due to distinct characteristics of the datasets.

6 Related Work

Learning to Rank and Text Ranking. Traditional learning to rank concerns tabular datasets with numerical features (Liu, 2009), for which a range of models are developed in the past decades: from SVMs (Joachims, 2006), gradient boosted decision trees (Burges, 2010), to neural rankers (Burges et al., 2005; Pang et al., 2020; Qin et al., 2021). Another focus is the design of ranking losses (surrogate to the ranking metrics), categorized into pointwise, pairwise, and listwise approaches (Cao et al., 2007; Bruch et al., 2020; Zhu & Klabjan, 2020; Jagerman et al., 2022a).

Recent advances in large neural language models have spurred interests in applying them on text ranking tasks (Lin et al., 2022), leading to the development of models including cross-attention (Han et al., 2020; Nogueira & Cho, 2020; Nogueira et al., 2020; Pradeep et al., 2021) and generative ones based on query likelihood (dos Santos et al., 2020; Zhuang & Zuccon, 2021; Zhuang et al., 2021; Sachan et al., 2022). Another family of work focuses on neural text retrieval models that emphasize efficiency, including dual-encoder (Karpukhin et al., 2020; Zhan et al., 2021), late-interaction (Khattab & Zaharia, 2020; Hui et al., 2022), and transformer memory (Tay et al., 2022).

Domain Adaptation. Following (Ben-David et al., 2007; Blitzer et al., 2008), a family of domain adaptation methods is based on learning (adversarial) domain-invariant feature representations (Long et al., 2015; Ganin et al., 2016; Courty et al., 2017). These methods are applied in fields including NLP, and on tasks ranging from cross-domain sentiment analysis, question-answering (Li et al., 2017; Vernikos et al., 2020), to unsupervised cross-lingual learning and machine translation (Xian et al., 2022; Lample et al., 2018). Our method, ListDA, belongs to this family, but to the best of our knowledge no prior work considers learning invariant representations of lists/sets.

Domain Adaptation for Information Retrieval. Existing work on this subject can be categorized into supervised and unsupervised domain adaptation. The former assumes access to labeled source data and (a small amount of; i.e. few-shot) labeled target data (Sun et al., 2021). The focus of this paper is on the latter, which assumes access to target documents but not annotated data (queries and relevance judgements). Cohen et al. (2018) is the first to apply invariant representation learning to unsupervised domain adaptation for text ranking, followed by Tran et al. (2019) for enterprise email search and Xin et al. (2022) for dense retrieval. Another family of approaches is based on query generation (Ma et al., 2021; Wang et al., 2022), originally proposed for dense retrieval.

7 Conclusion

We theoretically analyze domain adaptation for learning to rank and establish a generalization bound under ranking metrics. Our bound leads to a method based on learning invariant listwise representations that is generally applicable to any ranking problem. The method, called ListDA, demonstrates improved results on unsupervised domain adaptation for passage reranking on various domains.

The novelty of our results is that they are tailored to the listwise approach for learning to rank. Although existing work on binary classification and learning invariant item representations could immediately apply to domain adaptation under the pointwise approach, there are several limitations. Theoretically, pointwise generalization bounds would be too loose for a statement on ranking metrics defined on lists. Empirically, models trained with pointwise ranking losses are often less performant than listwise. In contrast, ListDA is a listwise approach, and works in tandem with listwise ranking losses that currently achieve SOTA performance on many ranking tasks. We believe our theoretical and empirical contributions provide a foundation for future studies on domain adaptation for ranking.

REFERENCES

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, 2018. *arXiv:1611.09268 [cs]*.

Peter L. Bartlett, Dylan J. Foster, and Matus J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, 2007.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, 2008.

Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. A Stochastic Treatment of Learning to Rank Scoring Functions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 61–69, 2020.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005.

Chris J.C. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 129–136, 2007.

Daniel Cohen, Bhaskar Mitra, Katja Hofmann, and W. Bruce Croft. Cross Domain Regularization for Neural Ranking Models using Adversarial Learning. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1025–1028, 2018.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, 2017.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of the Twenty-Eighth Text Retrieval Conference*, 2019.

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

D. A. Edwards. On the Kantorovich–Rubinstein theorem. *Expositiones Mathematicae*, 29(4):387–398, 2011.

Yanai Elazar and Yoav Goldberg. Adversarial Removal of Demographic Attributes from Text Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 11–21, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

John Guiver and Edward Snelson. Bayesian inference for Plackett-Luce ranking models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 377–384, 2009.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. A Deep Look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6), 2020.

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. Learning-to-Rank with BERT in TF-Ranking, 2020. *arXiv:2004.08476 [cs]*.

Kai Hui, Honglei Zhuang, Tao Chen, Zhen Qin, Jing Lu, Dara Bahri, Ji Ma, Jai Gupta, Cicero Nogueira dos Santos, Yi Tay, and Donald Metzler. ED2LM: Encoder-Decoder to Language Model for Faster Document Re-ranking Inference. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3747–3758, 2022.

Rolf Jagerman, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 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*, pp. 2303–2307, 2022a.

Rolf Jagerman, Xuanhui Wang, Honglei Zhuang, Zhen Qin, Michael Bendersky, and Marc Najork. Rax: Composable Learning-to-Rank Using JAX. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3051–3060, 2022b.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 217–226, 2006.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*, 2018.

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2237–2243, 2017.

Chen Liang, Haoming Jiang, Simiao Zuo, Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. No Parameters Left Behind: Sensitivity Guided Adaptive Learning Rate for Training Large Transformer Models. In *International Conference on Learning Representations*, 2022.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Pretrained Transformers for Text Ranking: BERT and Beyond*. Springer International Publishing, 2022.

Tie-Yan Liu. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning Transferable Features with Deep Adaptation Networks. In *International Conference on Machine Learning*, pp. 97–105, 2015.

R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1075–1088, 2021.

Rodrigo Nogueira and Kyunghyun Cho. Passage Re-ranking with BERT, 2020. *arXiv:1901.04085 [cs]*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 708–718, 2020.

Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 499–508, 2020.

Robin L. Plackett. The Analysis of Permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, pp. 193–202, 1975.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models, 2021. *arXiv:2101.05667 [cs]*.

Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees? In *International Conference on Learning Representations*, 2021.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5835–5847, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Alan Ramponi and Barbara Plank. Neural Unsupervised Domain Adaptation in NLP—A Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6838–6855, 2020.

Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. Improving Passage Retrieval with Zero-Shot Question Generation, 2022. *arXiv:2204.07496 [cs]*.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pp. 4058–4065, 2018.

Axel Suarez, Dyaa Albakour, David Corney, Miguel Martinez, and José Esquivel. A Data Collection for Evaluating the Retrieval of Related Tweets to News Articles. In *Advances in Information Retrieval*, pp. 780–786, 2018.

Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. Few-Shot Text Ranking with Meta Adapted Synthetic Weak Supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5030–5043, 2021.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer Memory as a Differentiable Search Index, 2022. *arXiv:2202.06991 [cs]*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Brandon Tran, Maryam Karimzadehgan, Rama Kumar Pasumarthi, Michael Bendersky, and Donald Metzler. Domain Adaptation for Enterprise Email Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 25–34, 2019.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artiéres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 2017.

Giorgos Vernikos, Katerina Margatina, Alexandra Chronopoulou, and Ion Androutsopoulos. Domain Adversarial Fine-Tuning as an Effective Regularizer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3103–3112, 2020.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. TREC-COVID: Constructing a pandemic information retrieval test collection. *ACM SIGIR Forum*, 54:1–12, 2021.

Ellen M. Voorhees. The TREC-8 Question Answering Track Report. In *Proceedings of the Eighth Text Retrieval Conference*, pp. 77–82, 1999.

Ellen M. Voorhees. The TREC robust retrieval track. *ACM SIGIR Forum*, 39:11–20, 2005.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2345–2360, 2022.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1192–1199, 2008.

Ruicheng Xian, Heng Ji, and Han Zhao. Cross-Lingual Transfer with Class-Weighted Language-Invariant Representations. In *International Conference on Learning Representations*, 2022.

Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul Bennett. Zero-Shot Dense Retrieval with Momentum Adversarial Domain Invariant Representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 4008–4020, 2022.

Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1253–1256, 2017.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing Dense Retrieval Model Training with Hard Negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1503–1512, 2021.

Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, 2018.

Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E. Gonzalez, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Kurt Keutzer. A Review of Single-Source Deep Unsupervised Visual Domain Adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):473–493, 2022.

Xiaofeng Zhu and Diego Klabjan. Listwise Learning to Rank by Exploring Unique Ratings. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 798–806, 2020.

Shengyao Zhuang and Guido Zuccon. TILDE: Term Independent Likelihood moDEl for Passage Re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1483–1492, 2021.

Shengyao Zhuang, Hang Li, and Guido Zuccon. Deep Query Likelihood Model for Information Retrieval. In *Advances in Information Retrieval*, pp. 463–470, 2021.

A OMITTED PROOFS

Before proving the generalization bounds for binary classification (Theorem 1) and learning to rank (Theorem 2), recall the following properties of Lipschitz functions:

Fact 6.

1. *$f : \mathbb{R}^d \to \mathbb{R}$ is differentiable, then it is L-Lipschitz under Euclidean distance if and only if $\|\nabla f\|_2 \leq L$.*
2. *If $f : \mathcal{X} \to \mathbb{R}$ is L-Lipschitz and $g : \mathcal{X} \to \mathbb{R}$ is M-Lipschitz, then $af + bg$ is $(|a|L + |b|M)$-Lipschitz, and $\max(f, g)$ is $\max(L, M)$-Lipschitz.*
3. *If $f : \mathcal{X} \to \mathcal{Y}$ is L-Lipschitz and $g : \mathcal{Y} \to \mathcal{Z}$ is M-Lipschitz, then $g \circ f$ is LM-Lipschitz.*

Proof. For the first statement, suppose bounded gradient norms, then by mean value theorem $\exists t \in [0, 1]$ s.t. $f(y) - f(x) = \nabla f(z)^\top (y - x)$ with $z := (1 - t)x + ty$, so by Cauchy-Schwarz,

$$\|f(y) - f(x)\|_2 \leq \|\nabla f(z)\|_2 \|y - x\|_2 \leq L\|y - x\|_2.$$

Next, suppose L-Lipschitzness, then by differentiability, $\nabla f(x)^\top z = f(x + z) - f(x) + o(\|z\|_2)$. Set $z := t\nabla f(x)$, we have

$$t\|\nabla f(x)\|_2^2 = f(x + t\nabla f(x)) - f(x) + o(t\nabla f(x)) \leq Lt\|\nabla f(x)\|_2 + o(t\|\nabla f(x)\|_2),$$

and the result follows by dividing both sides by $t\|\nabla f(x)\|_2$ and taking $t \to 0$.

For the second,

$$\begin{aligned}
&|af(x) + bg(x) - (af(y) + bg(y))| \\
&\qquad \leq |a||f(x) - f(y)| + |b||g(x) - g(y)| \leq (|a|L + |b|M)d_{\mathcal{X}}(x, y).
\end{aligned}$$

Next, assume w.l.o.g. $\max(f(x), g(x)) - \max(f(y), g(y)) \geq 0$, then

$$\begin{aligned}
&|\max(f(x), g(x)) - \max(f(y), g(y))| \\
&\quad = \begin{cases} f(x) - \max(f(y), g(y)) \leq f(x) - f(y) \leq Ld_{\mathcal{X}}(x, y) & \text{if } \max(f(x), g(x)) = f(x) \\ g(x) - \max(f(y), g(y)) \leq g(x) - g(y) \leq Md_{\mathcal{X}}(x, y) & \text{else} \end{cases} \\
&\quad \leq \max(L, M)d_{\mathcal{X}}(x, y).
\end{aligned}$$

For the third, $d_{\mathcal{Z}}(g \circ f(x), g \circ f(y)) \leq Md_{\mathcal{Y}}(f(x), f(y)) \leq LMd_{\mathcal{X}}(x, y)$. □

We first prove Theorem 1, as it shares the same organization of the arguments with Theorem 2.

Proof of Theorem 1. Define $\eta := \mathbb{1}(Y = 1)$, then $\mathcal{E}(f) = \mathbb{E}_{(X,Y)\sim\mu}[\eta - (2\eta - 1)f(X)]$. Note that

$$\begin{aligned}
\mathcal{E}(f) - \mathcal{E}(f') &= \mathbb{E}_{(X,Y)\sim\mu}[\eta - (2\eta - 1)f(X)] - \mathbb{E}_{(X,Y)\sim\mu}[\eta - (2\eta - 1)f'(X)] \\
&= \mathbb{E}_{(X,Y)\sim\mu}[(2\eta - 1) \cdot (f'(X) - f(X))] \\
&\leq \mathbb{E}_{X\sim\mu^X}[|f'(X) - f(X)|]
\end{aligned}$$

because $(2\eta - 1) = \pm 1$. On the other hand,

$$\begin{aligned}
\mathbb{E}_{X\sim\mu^X}[|f(X) - f'(X)|] &= \mathbb{E}_{(X,Y)\sim\mu}[|(2\eta - 1) \cdot (f(X) - f'(X)) - \eta + \eta|] \\
&\leq \mathbb{E}_{(X,Y)\sim\mu}[|(2\eta - 1)f(X) - \eta|] + \mathbb{E}_{(X,Y)\sim\mu}[|-(2\eta - 1)f'(X) + \eta|] \\
&= \mathbb{E}_{(X,Y)\sim\mu}[\eta - (2\eta - 1)f(X)] + \mathbb{E}_{(X,Y)\sim\mu}[\eta - (2\eta - 1)f'(X)] \\
&= \mathcal{E}(f') + \mathcal{E}(f).
\end{aligned}$$

Then by Fact 6, the fact that taking absolute value is 1-Lipschitz, and Definition 3, for all $f, f' \in \mathcal{F}$,

$$\begin{aligned}
\mathcal{E}_T(f) &= \mathcal{E}_S(f) + (\mathcal{E}_T(f) - \mathcal{E}_T(f')) - (\mathcal{E}_S(f) + \mathcal{E}_S(f')) + (\mathcal{E}_S(f') + \mathcal{E}_T(f')) \\
&\leq \mathcal{E}_S(f) + \Big(\mathbb{E}_{X\sim\mu_T^X}[|f(X) - f'(X)|] - \mathbb{E}_{X\sim\mu_S^X}[|f(X) - f'(X)|]\Big) + (\mathcal{E}_S(f') + \mathcal{E}_T(f')) \\
&\leq \mathcal{E}_S(f) + \sup_{q\in\mathrm{Lip}(2L)} (\mathbb{E}_{X\sim\mu_T^X}[q(X)] - \mathbb{E}_{X\sim\mu_S^X}[q(X)]) + (\mathcal{E}_S(f') + \mathcal{E}_T(f')) \\
&\leq \mathcal{E}_S(f) + 2L \cdot W_1(\mu_S^X, \mu_T^X) + (\mathcal{E}_S(f') + \mathcal{E}_T(f')).
\end{aligned}$$

and the result follows by taking the min over f'. □

Proof of Theorem 2. Fix $g \in \mathcal{G}$, which has a L_g-Lipschitz inverse g^{-1} to $\mathrm{supp}(\mu_S^X)$ by Assumption 3. Define $\epsilon_{h,g} : \mathcal{Z} \to \mathbb{R}_{\geq 0}$ for any given $h : \mathcal{Z} \to \mathbb{R}^\ell$ by

$$\begin{aligned}\epsilon_{h,g}(z) &:= \max_{r \in S_\ell} u(r, y_S \circ g^{-1}(z)) - \mathbb{E}_{R \sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))] \\ &= \max_{r \in S_\ell} u(r, y_S \circ g^{-1}(z)) - \sum_{r \in S_\ell} u(r, y_S \circ g^{-1}(z)) \prod_{i=1}^{\ell} \frac{\exp(h(z)_{I(r)_i})}{\sum_{j=i}^{\ell} \exp(h(z)_{I(r)_j})},\end{aligned}$$

and note that $\mathcal{E}_S(h \circ g) = \mathbb{E}_{X \sim \mu_S^X}[\epsilon_{h,g}(g(X))] =: \mathbb{E}_{Z \sim \mu_S^Z}[\epsilon_{h,g}(z)]$. An analogous analysis holds for $\mathcal{E}_T$.

We show that $\epsilon_{h,g}$ as written above is a Lipschitz function of z if h is Lipschitz. For the first term, because u is L_u-Lipschitz in $y_S \circ g^{-1}(z)$ and $y_S \circ g^{-1}(z)$ is $L_y L_g$-Lipschitz in z, so u is $L_u L_y L_g$-Lipschitz in z, and so is $z \mapsto \max_{r \in S_\ell} u(r, y_S \circ g^{-1}(z))$ by Fact 6. Now we bound the second term. We show that it is Lipschitz in both $y_S \circ g^{-1}(z) =: y$ and $h(z) =: s$ under the Euclidean distance. By triangle inequality and Jensen's inequality,

$$\|\nabla_y \mathbb{E}_{R \sim p_{\exp(s)}}[u(R, y)]\|_2 = \|\mathbb{E}_{R \sim p_{\exp(s)}}[\nabla_y u(R, y)]\|_2 \leq \mathbb{E}_{R \sim p_{\exp(s)}}[\|\nabla_y u(R, y)\|_2] \leq L_u.$$

Next,

$$\|\nabla_s \mathbb{E}_{R \sim p_{\exp(s)}}[u(R, y)]\|_2 = \sqrt{\sum_{k=1}^{\ell} \nabla_s \left(\sum_{r \in S_\ell} u(r, y) \prod_{i=1}^{\ell} \frac{\exp(s_{I(r)_i})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right)_k^2} \leq B\sqrt{\ell},$$

where the inequality is due to product rule and

$$\begin{aligned}&\left| \frac{\partial}{\partial s_{I(r)_m}} \left(\sum_{r \in S_\ell} u(r, y) \prod_{i=1}^{\ell} \frac{\exp(s_{I(r)_i})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \right| \\ &\quad = \left| \sum_{r \in S_\ell} u(r, y) \sum_{i=1}^{\ell} \prod_{k \neq i} \left(\frac{\partial}{\partial s_{I(r)_m}} \frac{\exp(s_{I(r)_i})}{\sum_{j=k}^{\ell} \exp(s_{I(r)_j})} \right) \frac{\exp(s_{I(r)_i})}{\sum_{j=k}^{\ell} \exp(s_{I(r)_j})} \right| \\ &\quad = \sum_{r \in S_\ell} u(r, y) \sum_{i=1}^{\ell} \mathbb{1}(m \leq i) \left(1 - \frac{\exp(s_{I(r)_i})}{\sum_{j=k}^{\ell} \exp(s_{I(r)_j})} \right) \prod_{k=1}^{\ell} \frac{\exp(s_{I(r)_i})}{\sum_{j=k}^{\ell} \exp(s_{I(r)_j})} \\ &\quad \leq B \sum_{r \in S_\ell} \sum_{i=1}^{\ell} \prod_{k=1}^{\ell} \frac{\exp(s_{I(r)_i})}{\sum_{j=k}^{\ell} \exp(s_{I(r)_j})} = B;\end{aligned}$$

recall that $\frac{\mathrm{d}}{\mathrm{d}x_i}\mathrm{softmax}(x)_j = \mathrm{softmax}(x)_i(\mathbb{1}(i = j) - \mathrm{softmax}(x)_j)$. Suppose $h \in \mathrm{Lip}(L_h)$, then $z \mapsto \mathbb{E}_{R \sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))]$ is Lipschitz because

$$\begin{aligned}&\left| \mathbb{E}_{R \sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))] - \mathbb{E}_{R \sim p_{\exp(h(z'))}}[u(R, y_S \circ g^{-1}(z'))] \right| \\ &\quad \leq \left| \mathbb{E}_{R \sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))] - \mathbb{E}_{R \sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z'))] \right| \\ &\qquad\qquad + \left| \mathbb{E}_{R \sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z'))] - \mathbb{E}_{R \sim p_{\exp(h(z'))}}[u(R, y_S \circ g^{-1}(z'))] \right| \\ &\quad \leq L_u \|y_S \circ g^{-1}(z) - y_S \circ g^{-1}(z')\|_2 + B\sqrt{\ell}\|h(z) - h(z')\|_2 \\ &\quad \leq (L_u L_y L_g + B L_h \sqrt{\ell}) d_{\mathcal{X}}(x, x').\end{aligned}$$

Putting everything together, $\epsilon_{h,g}$ is $(2L_u L_y L_g + B L_h \sqrt{\ell})$-Lipschitz in z for any $h \in \mathrm{Lip}(L_h)$.

Then by Fact 6 and Definition 3, for all $g \in \mathcal{G}$ and $h, h' \in \mathcal{H}$,

$$\begin{aligned}
\mathcal{E}_T(h \circ g) &= \mathcal{E}_S(h \circ g) + (\mathcal{E}_T(h \circ g) - \mathcal{E}_T(h' \circ g)) - (\mathcal{E}_S(h \circ g) + \mathcal{E}_S(h' \circ g)) \\
&\qquad + (\mathcal{E}_S(h' \circ g) + \mathcal{E}_T(h' \circ g)) \\
&\leq \mathcal{E}_S(h \circ g) + (\mathcal{E}_T(h \circ g) - \mathcal{E}_T(h' \circ g)) - (\mathcal{E}_S(h \circ g) - \mathcal{E}_S(h' \circ g)) \\
&\qquad + (\mathcal{E}_S(h' \circ g) + \mathcal{E}_T(h' \circ g)) \\
&= \mathcal{E}_S(h \circ g) + \mathbb{E}_{Z \sim \mu_T^Z}[\epsilon_{h,g}(Z) - \epsilon_{h',g}(Z)] - \mathbb{E}_{Z \sim \mu_S^Z}[\epsilon_{h,g}(Z) - \epsilon_{h',g}(Z)] \\
&\qquad + (\mathcal{E}_S(h' \circ g) + \mathcal{E}_T(h' \circ g)) \\
&\leq \mathcal{E}_S(h \circ g) + \sup_{q \in \mathrm{Lip}(2(2L_u L_y L_g + BL_h\sqrt{\ell}))} \Big(\mathbb{E}_{Z \sim \mu_T^Z}[q(Z)] - \mathbb{E}_{Z \sim \mu_S^Z}[q(Z)]\Big) \\
&\qquad + (\mathcal{E}_S(h' \circ g) + \mathcal{E}_T(h' \circ g)) \\
&\leq \mathcal{E}_S(h \circ g) + 2(2L_u L_y L_g + BL_h\sqrt{\ell}) \cdot W_1(\mu_S^Z, \mu_T^Z) + (\mathcal{E}_S(h' \circ g) + \mathcal{E}_T(h' \circ g)),
\end{aligned}$$

and the result follows by taking the min over h'. □

Finally, we verify the Lipschitz conditions for RR and NDCG.

Proof of Corollary 3. It suffices to verify that $y \mapsto \mathrm{RR}(r, y)$ is 1-Lipschitz, which follows from the fact that RR ≤ 1 and $\|y - y'\|_2 \geq 1$ for all $y, y' \in \{0,1\}^\ell, y \neq y'$. □

Proof of Corollary 4. It suffices to verify that

$$y \mapsto \mathrm{NDCG}(r, y) := \frac{\mathrm{DCG}(r, y)}{\mathrm{IDCG}(y)} = \left(\sum_{i \in [\ell]} \frac{y_i}{\log(r_i^* + 1)}\right)^{-1} \sum_{i \in [\ell]} \frac{y_i}{\log(r_i + 1)}$$

is Lipschitz. Note that $\mathrm{IDCG}(y) = \max_r \mathrm{DCG}(r, y)$, a max of continuous functions, is piecewise continuous in y where each piece is defined by an $r' \in S_\ell$: $\{y : r' = \arg\max_r \mathrm{DCG}(r, y)\}$.

Let $r \in S_\ell$, and $y, y' \in \mathbb{R}^\ell$ s.t. $\arg\max_{r'} \mathrm{DCG}(r', y) = \arg\max_{r'} \mathrm{DCG}(r', y') =: r^*$, i.e. they are on the same piece for IDCG. Then

$$\begin{aligned}
&\left|\frac{\partial}{\partial y_k} \mathrm{NDCG}(r, y)\right| \\
&\quad = \left|\mathrm{IDCG}(y)^{-1} \cdot \frac{\partial}{\partial y_k} \sum_{i \in [\ell]} \frac{y_i}{\log(r_i + 1)} - \mathrm{DCG}(r, y) \cdot \left(\frac{\partial}{\partial y_k} \sum_{i \in [\ell]} \frac{y_i}{\log(r_i^* + 1)}\right)^{-2}\right| \\
&\quad \leq \left|\mathrm{IDCG}(y)^{-1} \cdot \log(r_k + 1)^{-1}\right| + \left|\mathrm{DCG}(r, y) \cdot \log(r_k^* + 1)^2\right| \\
&\quad \leq \left|\mathrm{IDCG}(y)^{-1} \cdot \log(2)^{-1}\right| + \left|\mathrm{DCG}(r, y) \cdot \log(\ell + 1)^2\right| \\
&\quad \leq U_{\min}^{-1} + U_{\max} \log(\ell + 1)^2,
\end{aligned}$$

so NDCG is $\sqrt{\ell}(U_{\min}^{-1} + U_{\max} \log(\ell + 1)^2)$-Lipschitz by combining the above and the fact that IDCG is continuous. □

Proof of Proposition 5. First,

$$\begin{aligned}
W_1(\mu_S^Z, \mu_T^Z) &= \inf_{\gamma \in \Gamma(\mu_S^Z, \mu_T^Z)} \int_{\mathcal{Z} \times \mathcal{Z}} \|z - z'\| \,\mathrm{d}\gamma(z, z') \leq B \inf_{\gamma \in \Gamma(\mu_S^Z, \mu_T^Z)} \int_{\mathcal{Z} \times \mathcal{Z}} \mathbb{1}(z \neq z') \,\mathrm{d}\gamma(z, z') \\
&= B\left(1 - \sup_{\gamma \in \Gamma(\mu_S^Z, \mu_T^Z)} \int_{\mathcal{Z} \times \mathcal{Z}} \mathbb{1}(z = z') \,\mathrm{d}\gamma(z, z')\right) \\
&= B\left(1 - \int_{\mathcal{Z}} \min\big(\mu_S^Z(z), \mu_T^Z(z)\big) \,\mathrm{d}z\right) \\
&= B \int_{\mathcal{Z}} \max\big(0, \mu_T^Z(z) - \mu_S^Z(z)\big) \,\mathrm{d}z = \frac{B}{2} \int_{\mathcal{Z}} \big|\mu_T^Z(z) - \mu_S^Z(z)\big| \,\mathrm{d}z,
\end{aligned}$$

because $\int \mu_T^Z(z) - \mu_S^Z(z)\,\mathrm{d}z = 0$.

On the other hand, define $\hat{Y}(z) := \mathbb{1}(f_{\text{ad}}(z) \geq 1/2)$. Then the balanced total error rate of $\hat{Y}$ on predicting the domain identities is

$$\text{Err}(\hat{Y}) := \int_{\mathcal{Z}} \Big(\hat{Y}(z)\mu_S^Z(z) + (1-\hat{Y}(z))\mu_T^Z(z)\Big)\,\mathrm{d}z = 1 + \int_{\mathcal{Z}} \left(\hat{Y}(z) - \frac{1}{2}\right)\big(\mu_S^Z(z) - \mu_T^Z(z)\big)\,\mathrm{d}z.$$

This quantity is minimized with $\hat{Y}^*(z) = \mathbb{1}(\mu_T^Z(z) \geq \mu_S^Z(z))$, whereby

$$\text{Err}(\hat{Y}^*) = 1 - \frac{1}{2}\int_{\mathcal{Z}} \big|\mu_S^Z(z) - \mu_T^Z(z)\big|\,\mathrm{d}z \leq 1 - \frac{1}{B} W_1(\mu_S^Z, \mu_T^Z).$$

The conclusion follows from the balanced total classification accuracy of $\hat{Y}^*$ being $2 - \text{Err}(\hat{Y}^*)$. Note that $\text{Err}(\hat{Y}^*) \in [0,1]$, so $2 - \text{Err}(\hat{Y}^*) \in [1,2]$. □

Table 2: ListDA + QGen PL performance on ranking the top 1000 retrieved documents.

Dataset	Method	MAP	MRR@10	NDCG@5	NDCG@10	NDCG@20
Robust04	ListDA + QGen PL	0.2851*†	0.8039†	0.5761†	0.5386†	0.4975†
TREC-COVID		0.3168	0.8950	0.8539	0.8292	0.7820
BioASQ		0.6538*‡	0.5158	0.5547‡	0.5671*‡	0.5931*‡

*Improves upon zero-shot baseline with statistical significance ($p \leq 0.05$) under the two-tailed Student's t-test. †Improves upon QGen PL. ‡Improves upon ItemDA.

Table 3: Reranker performance on ranking the top 1000 retrieved documents on Signal-1M (RT).

Dataset	Method	MAP	MRR@10	NDCG@5	NDCG@10	NDCG@20
Signal-1M (RT)	BM25	**0.1740**	**0.5765**	**0.3639**	**0.3215**	**0.2905**
	Zero-shot	0.1511	0.4804	0.3068	0.2685	0.2410
	QGen PL	0.1541	0.5043	0.3238	0.2799	0.2497
	ListDA	0.1456	0.4629	0.3002	0.2602	0.2328
	ListDA + QGen PL	0.1549	0.5170	0.3261	0.2817	0.2505

B Additional Experiments

Signal-1M and ListDA + QGen PL. In this section, we include passage reranking results on the Signal-1M dataset (Suarez et al., 2018), included in the BEIR benchmark as well. They are in Table 3. As in most published results of neural rerankers on Signal-1M (Thakur et al., 2021; Liang et al., 2022), reranking does not improve performance on Signal-1M. This does not mean that neural rerankers are worse than BM25, but that MS MARCO is not a good source domain choice for transfer learning with Signal-1M as the target, which is likely due to the large domain shift between tweet retrieval and MS MARCO web search data—qualitatively, the text styles and task semantics are very different; see Table 5 for samples. Hence we discussions and comparisons below are among the reranking models.

On Signal-1M, QGen PL improves upon the zero-shot baseline, but ListDA does not, which is likely because the domain shift is too large for ListDA to find the correct alignment of source and target features without supervision. Therefore, we experiment supplementing ListDA with QGen q-d pairs training (**ListDA + QGen PL**). It is observed that ListDA performance on Signal-1M improves with + QGen PL, which could have benefited from QGen q-d pairs acting as anchor points for ListDA to find the correct correspondence between source and target.

The results of ListDA + QGen PL on the other datasets are in Table 2. Although it is the only method that consistently improves the zero-shot baseline (because of Signal-1M), it underperforms ListDA on the other datasets. Further improvements to this method may be possible with better strategies of balancing the contributions of ListDA and QGen PL.

Table 4: Hyperparameter settings of T5 reranker and domain discriminators for experiment results.

Dataset	Method	η_{rank}	η_{ad}	λ
Robust04	Zero-shot	1e-4	-	-
	QGen PL		-	-
	ItemDA		1e-3	0.01
	ListDA		4e-3	0.01
	ListDA + QGen PL		1e-3	0.02
TREC-COVID	Zero-shot	2e-4	-	-
	QGen PL		-	-
	ItemDA		8e-3	0.01
	ListDA		4e-3	0.01
	ListDA + QGen PL		4e-3	0.02
BioASQ	Zero-shot	2e-4	-	-
	QGen PL		-	-
	ItemDA		4e-3	0.01
	ListDA		8e-3	0.01
	ListDA + QGen PL		4e-3	0.02
Signal-1M (RT)	Zero-shot	5e-5	-	-
	QGen PL		-	-
	ListDA		2e-3	0.01
	ListDA + QGen PL		1e-3	0.02

C Details of Experiment Setup

Training List Construction with Negative Sampling. To train the reranker under supervision for minimizing the listwise ranking loss (as in training on MS MARCO labeled data and QGen pseudolabeled data), the model needs to see training example lists that contain both relevant and irrelevant q-d pairs. Because the MS MARCO dataset and QGen only provide relevant q-d pairs, we perform negative sampling to gather irrelevant pairs for constructing training lists: given a relevant q-d pair, let $d_1, \cdots, d_k$, $k \leq 1000$ denote the unannotated or irrelevant documents in the top 1000 results returned by BM25 on query q (we use the implementation of Anserini; Yang et al., 2017), then we sample 30 documents at random and treat them as being irrelevant to q. This way, our training lists have length $\ell = 31$, each containing one relevant and 30 (likely) irrelevant q-d pairs for the same query. Setting a larger ℓ should improve performance, but it also demands more memory and compute resources for training.

Eliminating False Negatives. Note that the above procedure does not prevent unannotated relevant documents from being selected during negative sampling and causing the training lists to contain false negatives. In fact, they are prevalent in the MS MARCO dataset due to duplicates and the existence of many semantically similar documents in the web. A sampling method with a lower chance of selecting false negatives is to first use a pre-trained reranker to score and rank the BM25-retrieved top-1000 documents, so that all relevant documents are now concentrated at the top provided good reranker performance, and then sample negatives among the rank-300 or greater documents (Qu et al., 2021). We use a cross-attention reranker trained on MS MARCO.

This improved sampling method is not used in our experiments to construct example lists for supervised training, but is used to construct source domain lists for ListDA feature alignment. In other words, the source domain feature lists seen by the domain discriminator in Eq. (1) are constructed using the improved method. This follows our observation that while the ranking loss is not very sensitive to false negatives, they affect ListDA feature alignment, likely due to the domain shift caused by the existence of many duplicates/semantically similar documents on MS MARCO and the lack of them on our target domains. While the above method for eliminating false negatives is largely sufficient to allow ListDA to provide good performance in our experiments, fundamentally, source domain training data should be prepared with more care.

Hyperparameters. For BM25, we use the implementation of Anserini (Yang et al., 2017), set $k_1 = 0.82$ and $b = 0.68$ on MS MARCO source domain, and $k_1 = 0.9$ and $b = 0.4$ on all target domains without tuning. As in (Thakur et al., 2021), if titles are available, they are indexed as a separate field with equal weights as the document body.

For the T5 reranker, we tune the learning rate $\eta_{\text{rank}} \in \{5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$, and select the one that gives the best zero-shot performance to use on all models on each dataset. We also apply a learning rate schedule on η_{rank} that decays by 0.7 every 5,000 steps. All rerankers are fine-tuned for 50,000 steps from the pre-trained T5 checkpoint.

For the domain discriminators, there are two hyperparameters: the strength of invariant feature learning $\lambda > 0$, and the discriminator learning rate η_{ad}. We select $\lambda \in \{0.01, 0.02\}$ and $\eta_{\text{ad}} \in \{10, 20, 40\}$ times the reranker learning rate η_{rank}.

The reranker and discriminator hyperparameters mentioned above for the results in Tables 1, 2 and 3 are provided in Table 4.

Table 5: Samples of test relevant and QGen q-d pairs from the text reranking domains experimented on. Truncated or omitted texts are indicated by "*[...]*".

Dataset	Test Q-D Pairs	QGen Q-D Pairs
MS MARCO	Q: what is the science of mapmaking called D: What is cartography? A. the science of mapmaking B. the science of shipbuilding C. the science of charting direction on a ship D. the science of measuring distances on the ocean. Cartography is the science of map making A. Q: what's in the flu shot D: The flu shot also contains the following ingredients: sodium phosphate & buffered isotonic sodium chloride solution, formaldehyde, octylphenol ethoxylate, and gelatin, according to the FDA.	-
TREC-COVID	Q: what is known about an mRNA vaccine for the SARS-CoV-2 virus? D: An Evidence Based Perspective on mRNA-SARS-CoV-2 Vaccine Development. *[...]* The production of mRNA-based vaccines is a promising recent development in the production of vaccines. However, there remain significant challenges in the development *[...]* Q: What is the mechanism of cytokine storm syndrome on the COVID-19? D: The possible pathophysiology mechanism of cytokine storm in elderly adults with COVID-19 infection: the contribution of "inflame-aging". PURPOSE: Novel Coronavirus disease 2019 (COVID-19), is an acute respiratory distress syndrome (ARDS), *[...]*	Q: what is arterial load for pulse pressure analysis D: Impact of arterial load on the agreement between pulse pressure analysis and esophageal Doppler. INTRODUCTION The reliability of pulse pressure analysis to estimate cardiac output is known to be affected by arterial load changes. *[...]* Q: opportunity cost pacifism D: Opportunity Costs Pacifism. If the resources used to wage wars could be spent elsewhere and save more lives, does this mean that wars are unjustified? This article considers this question, which has been largely overlooked by Just War Theorists and pacifists. It focuses on whether the opportunity costs of war *[...]*
BioASQ	Q: Is amantadine ER the first approved treatment for akinesia? D: The role of extended-release amantadine for the treatment of dyskinesia in Parkinson's disease patients. *[...]* Extended-release amantadine (amantadine ER) is the first approved medication for the treatment of dyskinesia. When it is given at bedtime, it *[...]* Q: Can leuprorelin acetate be used as androgen deprivation therapy? D: *[...]* We investigated the health-related quality of life (HRQoL) of long-term prostate cancer patients who received leuprorelin acetate in microcapsules (LAM) for androgen-deprivation therapy (ADT). *[...]*	Q: average age of femoral subluxation D: Subluxation of the femoral head in coxa plana. Twenty-two patients who had severe coxa plana had closed reduction for lateral subluxation of the femoral head, *[...]* The average age when the patients were first seen was eight years and six months. *[...]* Q: which scale is used for proxy assessment of hrqol D: *[...]* a comparison of proxy assessment and patient self-rating using the disease-specific Huntington's disease health-related quality of life questionnaire (HDQoL). *[...]* Specific Scales of the HDQoL. On the Specific Hopes and Worries Scale, proxies on average rated HrQoL as better than patients' *[...]*
Signal-1M (RT)	Q: Party MP calls BJP 'Baura Jayewala Party' D: BJP terms party MP R.K Singh's allegation that money has changed hands for tickets in #BiharPolls as baseless. Q: Kerry: US plans military talks with Russia over Syria D: Kerry: US plans military talks with Russia over Syria	Q: where is black lives matter? D: Black lives matter: thoughts from the delivery ward in St. Louis: #mustread Q: brenda got a baby pac D: RETWEET if "Brenda's Got A Baby" is one of your favorite @2Pac songs. #RIP2Pac