Learning List-Level Domain-Invariant Representations for Ranking*

Ruicheng Xian[1†] Honglei Zhuang[2] Zhen Qin[2] Hamed Zamani[3†] Jing Lu[2]
Ji Ma[2] Kai Hui[2] Han Zhao[1] Xuanhui Wang[2] Michael Bendersky[2]

February 10, 2023

Abstract

Domain adaptation aims to transfer the knowledge acquired by models trained on (data-rich) source domains to (low-resource) target domains, for which a popular method is invariant representation learning. While they have been studied extensively on classification and regression problems, how they apply on ranking problems, where data and metrics have a list structure, is not well understood. In this paper, we establish a domain adaptation generalization bound for ranking via representation invariance, and instantiate it to listwise metrics including MRR and NDCG. The key insight from our result is that on ranking problems, invariance should be analyzed on list-level representations, which respects the problem structure, as opposed to e.g. item-level representations. Based on the theory, we propose an adaptation method via learning list-level domain-invariant feature representations, and empirically demonstrate its benefits for unsupervised domain adaptation on real-world ranking tasks, including passage reranking.

1 Introduction

Learning to rank applies machine learning to solving ranking problems that are at the core of many everyday products and applications, including search engines and recommendation systems (Liu, 2009). The availability of ever-increasing amounts of training data has enabled larger and larger models to improve the state-of-the-art on more ranking tasks. A prominent example is text ranking and retrieval, where neural language models with billions of parameters easily outperform traditional ranking models (Nogueira et al., 2020). But on domains that have little to no annotated data for training, larger models may fare worse than simpler ones, such as gradient boosted decision trees (Qin et al., 2021).

To extend the benefits of large models to low-resource domains, a popular transfer learning technique is *domain adaptation* (Pan and Yang, 2010). Provided data-rich source training domains relevant to the target domain of interest, along with (unlabeled) target data, domain adaptation methods optimize the model on the source domain, while making the knowledge transferable to the target by estimating and adjusting for the source-target domain shift using target data, e.g., via learning invariant feature representations (Muandet et al., 2013; Ganin et al., 2016). While they have been studied extensively on classification and regression problems (Ben-David et al., 2007; Zhao et al., 2018), most domain adaptation works on ranking are application-specific, or limited in

*Comparison to *arXiv:2212.10764*: improved presentation.

†Work performed while at Google Research.

[1]University of Illinois Urbana-Champaign. {rxian2,hanzhao}@illinois.edu.

[2]Google Research. {hlz,zhenqin,ljwinnie,maji,kaihuibj,xuanhui,bemike}@google.com.

[3]University of Massachusetts Amherst. zamani@cs.umass.edu.

theoretical understanding. For instance, the methods proposed in (Ma et al., 2021; Sun et al., 2021; Wang et al., 2022) are only applicable to text retrieval and ranking with neural language models.

Furthermore, unlike traditional settings, both data and metrics on ranking have a *list structure* defined on the items to be ranked. Yet, a number of existing adaptation methods are applied rather on the item level, ignoring the intrinsic structure of the problem (Cohen et al., 2018; Tran et al., 2019; Xin et al., 2022). This raises the question of *whether the list structure of ranking shall play a role in domain adaptation for learning to rank?*

Our Contributions. This paper, therefore, aims to provide a theoretical understanding of domain adaptation on ranking problems, and in particular, examine the significance of the list structure therein. Then, we explore principled adaptation methods based on the study. Specifically:

1. We provide the first domain adaptation generalization bound on ranking problems via analyzing representation invariance, and instantiate it to metrics including MRR and NDCG (Section 3).

   The distinct feature of our bound is that it considers representation invariance on the *list level*, so that the problem structure is respected. It also suggests that *item-level* domain invariance, which ignores the list structure, is inappropriate for domain adaptation for ranking; it is corroborated by our experiments.

2. Based on the theory, we propose an adaptation method for ranking, called ListDA (Section 4). The method is based on learning list-level domain-invariant feature representations via adversarial training, and is generally applicable to any ranking problems and tasks.

   Compared to existing domain-invariant feature learning methods, the novelty of ours is that it aims for invariance at the list-level, as opposed to the item-level, and is more suited for learning to rank.

3. We demonstrate the benefits ListDA for unsupervised domain adaptation on the passage reranking task, where RankT5 models trained on MS MARCO are adapted to various specialized text domains (Section 5).

   ListDA is additionally evaluated on the numerical Yahoo! LETOR dataset to illustrate its versatility and general applicability (Appendix C).

2 Preliminaries

Learning to Rank. A ranking problem is defined by a joint distribution of list[4] (of items) $X \in \mathcal{X}$ and nonnegative relevance scores $Y = (Y_1, \cdots, Y_\ell) \in \mathbb{R}^\ell_{\geq 0}$ associated with the items in each list,[5] where all lists are assumed to be length-ℓ. We also assume that the ground-truth scores are a function of the list, $Y = y(X)$, so that the problem is equivalently defined by a marginal distribution μ^X of lists and scoring function $y : \mathcal{X} \to \mathbb{R}^\ell_{\geq 0}$.

The goal is to learn a *ranker*, $f' : \mathcal{X} \to S_\ell$, that maps each list x to rank assignments $r := f'(x) \in S_\ell$, where S_ℓ denotes the set of permutations on $\{1, 2, \cdots, \ell\}$ and r_i represents the predicted rank of item i within the list, so that r recovers the descending order of the ground-truth relevance scores

[4] More precisely, each input is a *multiset* of items, because it is permutation-invariant, i.e., switching the order of the items (and the scores correspondingly) does not alter the list.

[5] A common choice for $\mathcal{X}$ is the ℓ-fold Cartesian product of $\mathbb{R}^k$, i.e., $x \in \mathbb{R}^{\ell \times k}$ is a stack of k-dimensional feature vectors, each corresponds to an item in the list. But generally, the representation of x need not be decomposable into the items it contains.

y, i.e., $y_i \geq y_j \iff r_i \leq r_j$ for all i, j. The more common setup is to train a *scorer*, $f : \mathcal{X} \to \mathbb{R}^\ell$, wherein we first compute the ranking scores on each list, with $s_i \coloneqq f(x)_i$ being the predicted score of item i, then map s to rank assignments r, either by taking the descending order of the s_i's or via probabilistic models (Section 3).

The quality of the predicted ranks is measured by ranking metrics. We consider listwise metrics, $u : S_\ell \times \mathbb{R}^\ell_{\geq 0} \to \mathbb{R}_{\geq 0}$, which take as inputs the rank assignments on each list along with the ground-truth relevance scores and output a utility score. Popular metrics in information retrieval include reciprocal rank and normalized discounted cumulative gain (Voorhees, 1999; Järvelin and Kekäläinen, 2002):

Definition 2.1 (Reciprocal Rank)**.** Suppose the ground-truth relevance scores are binary, $y \in \{0, 1\}^\ell$, then the reciprocal rank (RR) of the rank assignments $r \in S_\ell$ is

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

The average RR on the dataset, $\mathbb{E}_{(X,Y)\sim\mu}[\mathrm{RR}(f(X), Y)]$, is often referred to as the mean reciprocal rank (MRR).

Definition 2.2 (NDCG)**.** The discounted cumulative gain (DCG) and the normalized DCG (with identity gain functions, w.l.o.g.) of the rank assignments $r \in S_\ell$ are

$$\mathrm{DCG}(r, y) = \sum_{i=1}^{\ell} \frac{y_i}{\log(r_i + 1)}, \quad \mathrm{NDCG}(r, y) = \frac{\mathrm{DCG}(r, y)}{\mathrm{IDCG}(y)},$$

where the ideal DCG, $\mathrm{IDCG}(y) = \max_{r' \in S_\ell} \mathrm{DCG}(r', y)$, is the maximum DCG value of a list, which is attained by the descending order of the ground-truth y_i's.

Domain Adaptation. We study the adaptation of a scorer trained on a source domain (μ_S^X, y_S) to a target domain (μ_T^X, y_T) of interest (to which the source is relevant, i.e. $y_S \approx y_T$). When their domain shift is small, in the sense that the representations are *domain-invariant*, $\mu_S^X \approx \mu_T^X$, scorers optimized on the source are generally expected to be transferable to the target without the explicit need for training on labeled target data. Specifically, in such scenarios, target performance can be bounded in terms of source performance. As an example, for binary classification, prior work has established the following generalization bound for randomized classifiers (Shen et al., 2018):

Theorem 2.3. *Let binary classification problems on a source and a target domain be given by joint distributions μ_S, μ_T of inputs and labels, $(X, Y) \in \mathcal{X} \times \{0, 1\}$, and $\mathcal{F} \subset [0, 1]^\mathcal{X}$ be a class of L-Lipschitz predictors. Define the error rate of a predictor $f \in \mathcal{F}$ on μ by*

$$\mathcal{R}(f) \coloneqq \mathbb{E}_{(X,Y)\sim\mu}[\mathbb{1}(Y \neq \hat{Y})] \coloneqq \mathbb{E}_{(X,Y)\sim\mu}[f(X) \cdot \mathbb{1}(Y \neq 1) + (1 - f(X)) \cdot \mathbb{1}(Y \neq 0)],$$

where $\mathbb{1}(\cdot)$ denotes the indicator function; it implies that the output class assignments are probabilistic with $\mathbb{P}(\hat{Y} = 1 \mid X = x) = f(x)$. Define the minimum joint error by $\lambda^* \coloneqq \min_{f'}(\mathcal{R}_S(f') + \mathcal{R}_T(f'))$, then for all $f \in \mathcal{F}$,

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

The domain shift in Theorem 2.3 is measured by the Wasserstein-1 distance, a probability metric, between the source and target marginal input distributions. The definition of Wasserstein-1 distance under the Kantorovich-Rubinstein dual formulation is (Edwards, 2011):

Definition 2.4 (Wasserstein-1 Distance)**.** Let p, q be probability measures on a metric space $(X, d_{\mathcal{X}})$. The Wasserstein-1 distance between p, q is $W_1(p, q) = \sup_{\psi \in \mathrm{Lip}(1)} (\int_{\mathcal{X}} \psi(x)\, \mathrm{d}p(x) - \int_{\mathcal{X}} \psi(x)\, \mathrm{d}q(x))$.

The supremum above is taken over 1-Lipschitz functionals, $\psi : \mathcal{X} \to \mathbb{R}$:

Definition 2.5 (Lipschitz Function)**.** Let $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{X}', d_{\mathcal{X}'})$ be metric spaces. A function $\psi : \mathcal{X} \to \mathcal{X}'$ is L-Lipschitz, denoted by $\psi \in \mathrm{Lip}(L)$, if $d_{\mathcal{X}'}(\psi(x_1), \psi(x_2)) \leq L d_{\mathcal{X}}(x_1, x_2)$ for all $x_1, x_2 \in \mathcal{X}$.

However, the domain adaptation bound for binary classification would not extend to ranking, where the metrics are computed on the list level. This is because such list structure does not arise on classification problems nor in the computation of error rate, and is therefore not handled in the above analysis. While it is possible to cast ranking into a classification problem then apply Theorem 2.3, this conversion is too loose to give a meaning bound in terms of listwise ranking metrics such as MRR or NDCG.

3 Generalization Bound for Ranking

We establish a domain adaptation generalization bound for ranking via analyzing representation invariance. Our analysis respects the list structure of the problem, tailored to which we will introduce and provide intuitions to the techniques and assumptions leading up to the result. Readers may skip directly to Theorem 3.6 and its discussions.

Problem Setup. Given a source and a target domain, $(\mu_S^X, y_S), (\mu_T^X, y_T)$, we consider the setting of jointly learning a scoring function and (transferable) representations, i.e., the end-to-end scorer $f = h \circ g$ is composed of a feature map $g : \mathcal{X} \to \mathcal{Z}$ and a scoring function $h : \mathcal{Z} \to \mathbb{R}^\ell$ on the learned list feature representations. For instance, if the scorer is an m-layer multilayer perceptron (MLP), then we could treat the first $(m-1)$ layers as g and the last as h.

We map the output ranking scores $s \coloneqq f(x) \in \mathbb{R}^\ell$ to rank assignments $r \in S_\ell$ in a probabilistic manner via a *Plackett-Luce model* (Plackett, 1975; Luce, 1959), using the exponentiated scores $\exp(s)$ as its parameters (Cao et al., 2007; Guiver and Snelson, 2009):

Definition 3.1 (Plackett-Luce Model)**.** A Plackett-Luce (P-L) model with parameters $w \in \mathbb{R}^\ell_{>0}$ specifies a distribution over S_ℓ, whose probability mass function p_w is

$$p_w(r) = \prod_{i=1}^{\ell} \frac{w_{I(r)_i}}{\sum_{j=i}^{\ell} w_{I(r)_j}}, \quad \forall r \in S_\ell,$$

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

Whereby, the utility is computed by taking the expectation of the metric $u : S_\ell \times \mathbb{R}^\ell_{\geq 0} \to \mathbb{R}_{\geq 0}$ w.r.t. $p_{\exp(s)}$ (take exp coordinate-wise), and the suboptimality of a scorer f on (μ^X, y) is evaluated by the *difference from the maximum attainable utility*, given by

$$\mathcal{R}(f) \coloneqq \mathbb{E}_{X \sim \mu^X} \left[\max_{r \in S_\ell} u(r, y(X)) \right] - \mathbb{E}_\mu[u(f)],$$

where we overloaded u for brevity, and defined $\mathbb{E}_\mu[u(f)] \coloneqq \mathbb{E}_{X \sim \mu^X} \mathbb{E}_{R \sim p_{\exp(f(X))}}[u(R, y(X))]$.

The purpose of using the P-L model for generating rank assignments rather than simply taking the descending order of s is technical—it makes the computation of utility scores (Lipschitz) continuous w.r.t. the raw scores output from the model (combined with the exponentiation). This

is analogous to the randomized class assignments in Theorem 2.3 on classification problems (via a Bernoulli model).

Also similar to Theorem 2.3, our bound is based on analyzing representation invariance, a framework that is first established by Ben-David et al. (2007) for binary classification. However, our analysis differs from that on classification; we point out that when the optimal joint error λ^* is zero, the tightness of Theorem 2.3 is due entirely to the uniqueness of the optimal classifier. In contrast, because of the list structure, the optimal ranker is generally nonunique: for example, note that the two rank assignments $(1, 2, 3)$ and $(2, 1, 3)$ both achieve maximum utility on the list with ground-truth scores $(1, 1, 0)$. Therefore, extending their analysis naïvely to ranking would result in a loose bound. Instead, we use an analysis that does not rely on uniqueness by leveraging the following Lipschitz assumptions:

Assumption 3.2. The ranking metric $u : S_\ell \times \mathbb{R}^\ell_{\geq 0} \to \mathbb{R}_{\geq 0}$ is upper bounded by B, and is (Euclidean) L_u-Lipschitz in the second argument, the ground-truth relevance scores y.

Assumption 3.3. The input lists X reside in a metric space $\mathcal{X}$, and the ground-truth scoring function $y : \mathcal{X} \to \mathbb{R}^\ell_{\geq 0}$ is L_y-Lipschitz (Euclidean on the output space).

We will show that Assumption 3.2 is satisfied by both RR and NDCG. Assumption 3.3 says that similar lists (i.e., close in $\mathcal{X}$) should have similar ground-truth scores, and is satisfied, e.g., when $\mathcal{X}$ is finite (this argument is used in Corollary 3.7); this is typically the case with text data, which are one-hot encoded as part of tokenization.

Assumption 3.4. The list features Z reside in a metric space $\mathcal{Z}$, and the class $\mathcal{H}$ of scoring functions, $h : \mathcal{Z} \to \mathbb{R}^\ell$, is L_h-Lipschitz (Euclidean on the output space).

Assumption 3.5. The class $\mathcal{G}$ of feature maps, $g : \mathcal{X} \to \mathcal{Z}$, satisfies that $\forall g \in \mathcal{G}$, the restrictions of g to the supports of μ^X_S and μ^X_T, $g|_{\text{supp}(\mu^X_S)}, g|_{\text{supp}(\mu^X_T)}$ respectively, are both invertible with L_g-Lipschitz inverses.

Assumption 3.4 is standard in generalization and complexity analyses, and could be enforced e.g. with L^2-regularization (Anthony and Bartlett, 1999; Bartlett et al., 2017). The last assumption is technical, which says that the original inputs can be recovered from their feature representations by a Lipschitz inverse g^{-1}, hence, conceptually, requires the feature map g to retain as much information from the inputs (on each domain). Note that this assumption does not conflict with domain-invariant representation learning; as long as $\mathcal{G}$ is sufficiently expressive, $\exists g \in \mathcal{G}$ satisfying $\mu^Z_S = \mu^Z_T$.

We are now ready to state our domain adaptation generalization bound for ranking:

Theorem 3.6. *Under Assumptions 3.2 to 3.5, for any $g \in \mathcal{G}$, define the minimum joint risk by $\lambda^*_g \coloneqq \min_{h'}(\mathcal{R}_S(h' \circ g) + \mathcal{R}_T(h' \circ g))$, then for all $h \in \mathcal{H}$,*

$$\mathcal{R}_T(h \circ g) \leq \mathcal{R}_S(h \circ g) + \lambda^*_g + 4(L_u L_y L_g + B\ell L_h) \cdot W_1(\mu^Z_S, \mu^Z_T),$$

where μ^Z denotes the marginal distribution of the list features Z, $\mu^Z(z) \coloneqq \mu^X(g^{-1}(z))$.

We instantiate it to MRR and NDCG by simply verifying the Lipschitz condition L_u of Assumption 3.2:

Corollary 3.7 (Bound for MRR)**.** RR *is 1-Lipschitz in y, and thereby*

$$\mathbb{E}_{\mu_T}[\text{RR}(h \circ g)] \geq \mathbb{E}_{\mu_S}[\text{RR}(h \circ g)] - \lambda^*_g - 4(L_y L_g + \ell L_h) \cdot W_1(\mu^Z_S, \mu^Z_T).$$

Corollary 3.8 (Bound for NDCG). *If* $u_{\min} \leq \text{IDCG} \leq u_{\max}$ *for some* $u_{\min}, u_{\max} \in (0, \infty)$ *on both* (μ_S^X, y_S) *and* (μ_T^X, y_T) *almost surely, then* NDCG *is* $\widetilde{O}(\sqrt{\ell})$*-Lipschitz in* y *almost surely, and thereby*

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

IDCG is lower bounded e.g. if every list contains at least one relevant item, and upper bounded when the ground-truth relevance scores are.

The main message of this bound is that, if g learns domain-invariant feature representations, i.e. $W_1(\mu_S^Z, \mu_T^Z) \approx 0$, and retains (domain-invariant) information that are useful for ranking so that the optimal joint risk $\lambda_g^* \approx 0$, then an end-to-end scorer $h \circ g$ with good target performance can be obtained by simply optimizing the scoring head h on labeled source data. It naturally suggests an adaptation method based on learning list-level domain-invariant feature representations, which we will explore in Sections 4 and 5.

It should be emphasized that the domain invariance of features considered in our Theorem 3.6 is on the list level, $\mu_S^Z = \mu_T^Z$, which is its key distinction from prior results. As a concrete illustration, suppose the setup where the *list feature* is a stack of k-dimensional feature vectors, each associated with an item in the list, $z = (v_1, \cdots, v_\ell) \in \mathbb{R}^{\ell \times k}$. Then, a possible alternative to list-level invariance is *item-level invariance*, $\mu_S^V = \mu_T^V$, where $\mu^V(v) \coloneqq \mathbb{P}_{Z \sim \mu^Z}(v \in Z)$ denotes the distribution of *item feature* vectors. However, while $\mu_S^Z = \mu_T^Z \implies \mu_S^V = \mu_T^V$, the converse is generally not true, $\mu_S^V = \mu_T^V \not\implies \mu_S^Z = \mu_T^Z$ (see Example A.2)! In other words, list-level invariance is a stronger requirement than item-level invariance, and the latter, which ignores the list structure of the data, is insufficient to guarantee domain transferability. We note that this is not an artifact of our analysis; bounds based on item-level invariance cannot be tight under listwise ranking metrics.

We will corroborate the discussions above by the experiments in Section 5 and Appendix C, where we demonstrate the empirical benefits of learning list-level domain-invariant feature representations for domain adaptation (ListDA), and in comparison, highlight the deficiencies of learning item-level invariant features (ItemDA)—used in several (recent) work on domain adaptation for ranking (Cohen et al., 2018; Tran et al., 2019; Xin et al., 2022).

4 Learning List-Level Domain-Invariant Representations

In this section, we describe a domain adaptation method for ranking via learning list-level invariant feature representations, called ListDA. Same as in Section 3, we consider training a composite scorer, $h \circ g$, and specifically, assume that the list feature is a stack of k-dimensional feature vectors, $g(x) = (v_1, \cdots, v_\ell) \in \mathbb{R}^{\ell \times k}$, with v_i corresponding to the i-th item in x. This setup appears in listwise ranking models (Cao et al., 2007), and is common in many ranking systems; e.g., in neural text ranking, each feature vector is the embedding of the input text by a language model (Nogueira and Cho, 2020; Guo et al., 2020).

It is discussed in the remark following Theorem 3.6 that under the setting of domain adaptation, the composite scorer will perform well on the (low-resource or unlabeled) target domain if the features learned by g is domain-invariant (i.e., $W_1(\mu_S^Z, \mu_T^Z)$ and $\lambda_g^* \approx 0$) and the model is optimized on the source ($\mathcal{R}_S \approx 0$). To achieve these two goals, a well-known approach is adversarial training (Goodfellow et al., 2014; Ganin et al., 2016; Arjovsky et al., 2017). A description is provided below for completeness; readers familiar with this method may skip to the paragraph Parameterization of f_{ad}.

In adversarial training, we have the joint minimax objective of

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

where $\min_{f_{\text{ad}}} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}})$ is the adversarial component that we will relate to $W_1(\mu_S^Z, \mu_T^Z)$ below, $\mathcal{L}_{\text{rank}}$ is a ranking loss of choice, and $\lambda > 0$ is a hyperparameter that controls the strength of invariant feature learning. Optimization is typically performed by gradient descent-ascent (w.r.t. h, g and f_{ad}, respectively) with a gradient reversal layer added on top of g.

Under this objective, the model learns domain-invariant features by minimizing their distributional discrepancy, $\mu_S^Z \approx \mu_T^Z$, and is simultaneously trained on the source ranking task to ensure that the learned features are useful. We note that it does not require any labeled target data since the adversarial component (described below) only needs unlabeled input data, so it is applicable to *unsupervised domain adaptation*. A potential concern is that λ_g^* is not guaranteed to be small without supervision, but it does not preclude the empirical success of this method in fields ranging from vision (Zhao et al., 2022) to language (Ramponi and Plank, 2020), and in our ranking experiments.

Now, the adversarial component consists of an adversary $f_{\text{ad}} : \mathcal{Z} \to \mathbb{R}$, usually parameterized by neural networks, and an adversarial loss function $\ell_{\text{ad}} : \mathbb{R} \times \{0, 1\} \to \mathbb{R}$, whose inputs are the output of f_{ad} and the domain identity a (set to 1 for target domain). Then, the adversarial loss is

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

When ℓ_{ad} is the 0-1 loss of $\ell_{\text{ad}}(\hat{a}, a) = (1 - a) \cdot \mathbb{1}(\hat{a} \geq 0) + a \cdot \mathbb{1}(\hat{a} < 0)$, f_{ad} acts as a *domain discriminator* for predicting the domain identities, and $\mathcal{L}_{\text{ad}}$ is the balanced classification error of f_{ad}, $\mathcal{L}_{\text{ad}}(g, f_{\text{ad}}) = \mathbb{P}_{\mu_S^X}(f_{\text{ad}} \circ g(X) \geq 0) + \mathbb{P}_{\mu_T^X}(f_{\text{ad}} \circ g(X) < 0)$; it upper bounds $W_1(\mu_S^Z, \mu_T^Z)$:

Proposition 4.1. *Denote the metric on $\mathbb{R}^{\ell \times k}$ by d, and define $D \coloneqq \sup_{(z,z') \in \text{supp}(\mu_S^Z \times \mu_T^Z)} d(z, z')$. If ℓ_{ad} is the 0-1 loss, then $W_1(\mu_S^Z, \mu_T^Z) \leq D(1 - \min_{f_{\text{ad}}} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}}))$.*

To train f_{ad} for predicting domain identity, in practice, the 0-1 loss is replaced by a surrogate loss, for which we use the logistic loss in our experiments (Goodfellow et al., 2014):

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

Parameterization of f_{ad}. The final missing piece is the specification of f_{ad}, whose goal here, unlike traditional settings, is to model lists[4] of feature vectors, $z = (v_1, \cdots, v_\ell)$. A possible design choice is to flatten each $z \in \mathbb{R}^{\ell \times k}$ into a single (ℓk)-dimensional vector and set f_{ad} to an MLP. However, this model does not intrinsically capture the permutation invariance property of lists, and could make the optimization data inefficient. Instead, we set f_{ad} to transformers with mean-pooling as a novelty in our implementation (Vaswani et al., 2017), which satisfies permutation invariance (without positional encoding). A block diagram of ListDA can be found in Fig. 1.

To our knowledge, this is the first application of adversarial training to learning domain-invariant feature representations on the list level, as inspired by our Theorem 3.6.

5 Experiments on Passage Reranking

To demonstrate the benefits of *list-level* invariant feature representations, we evaluate ListDA for unsupervised domain adaptation on the passage[6] reranking task. Our method is not specialized to text (re)ranking; in Appendix C, we additionally evaluate ListDA on a web ranking task from the Yahoo! Learning to Rank Challenge. Furthermore, ListDA is compared to ItemDA, a method based on learning *item-level* invariant features, to illustrate the deficiencies of item-level invariance for domain adaptation discussed in Section 3.

[6]We use the terms *document*, *text* and *passage* interchangeably.

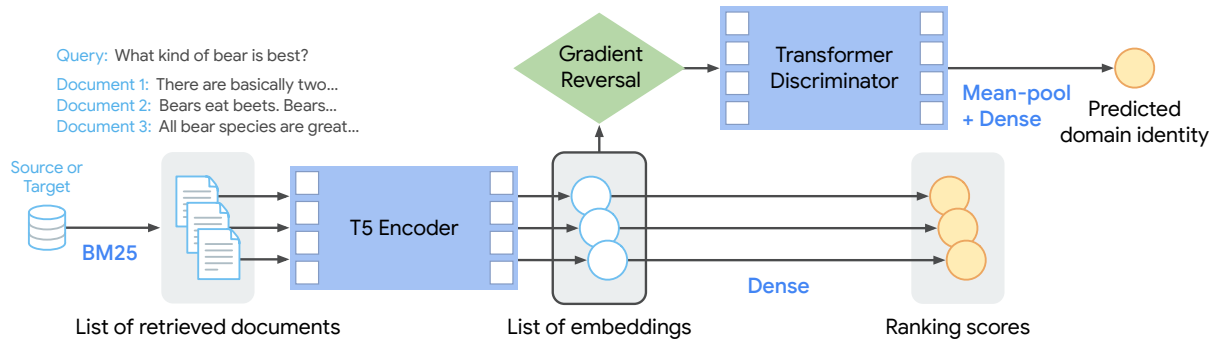


Figure 1: Block diagram of ListDA instantiated on the cross-attention RankT5 text ranking model.

In passage reranking, given a text query, the goal is to rank candidate passages in a retrieved set based on their relevance to the query. Reranking is employed in scenarios where the corpus is too large for all (millions of) documents to be ranked exhaustively by accurate but expensive models such as cross-attention rankers. Rather, a simple but efficient first-stage model such as sparse or dense retrievers (e.g., BM25 and DPR; Robertson and Zaragoza, 2009; Karpukhin et al., 2020) is used to retrieve a candidate set of (hundreds of) passages, whose ranks are then refined and improved by more sophisticated ranking models.

Datasets. The source domain in our experiments is MS MARCO for passage ranking, a large-scale dataset containing 8 million passages crawled from the web that covers a wide range of topics, and 532,761 search query and relevant passage pairs (Bajaj et al., 2018). The target domains are biomedical (TREC-COVID, BioASQ) and news articles (Robust04) (Voorhees et al., 2021; Tsatsaronis et al., 2015; Voorhees, 2005), where annotations could be costly to acquire. The data are collected and preprocessed according to the BEIR benchmark (Thakur et al., 2021); their paper also contains statistics of the datasets.

In our unsupervised setting, we do not assume the availability of queries and document-relevance annotations on the target domains. However, queries are required by (cross-attention) ranking models in the computation of feature representations (whose inputs are concatenations of query-document pairs), and access to model features is needed for adaptation methods such as ListDA to operate. So to make up for the queries on the target domain, we synthesize training queries on all target domains in a zero-shot manner following (Ma et al., 2021), with a T5 XL query generator (QGen) trained on MS MARCO relevant q-d pairs. QGen synthesizes each query as a seq-to-seq task given a passage from the target corpus as input, and the synthesized queries are expected to be related to the input passages. See Table 7 for samples of QGen q-d pairs.

Models. We use BM25 as the first-stage retriever for simplicity, and focus on the adaptation of the RankT5 listwise reranker (Zhuang et al., 2022)—a cross-attention model derived from the T5 base language model with 250 million parameters (Raffel et al., 2020). We treat the list of query-document embeddings output from the T5 encoder to be the list feature, which is consistent with the setting in Section 4. For the ranking loss, we use softmax cross-entropy:

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

The adversarial component follows Section 4. For the discriminator f_{ad}, we use a stack of three transformer blocks with the same architecture as those of the T5 encoder. To predict the domain

Table 1: Transfer performance of RankT5 on top 1000 BM25-retrieved passages.

Target Domain	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***‡

Source domain is MS MARCO. Gain function in NDCG is the identity map. *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.

of a list feature $z = (v_1, \cdots, v_\ell)$, we feed all vectors v_i through the transformer blocks at once as a sequence, take the mean-pool of the outputs and project to a logit with a dense layer. To reduce the sensitivity to the randomness in the initialization and the training process, we use an ensemble of five discriminators as in (Elazar and Goldberg, 2018), $\sum_{i=1}^{5} \mathcal{L}_{\text{ad}}(g, f_{\text{ad}}^{(i)})$. Figure 1 provides a block diagram of our model.

Baseline Methods.[7] To demonstrate the benefits of *list-level* invariant representations, we compare ListDA to zero-shot learning and a method called QGen PL. In **zero-shot**, the reranker is trained on MS MARCO only and directly evaluated on the target; it serves as a sanity check for the adaptation methods. In **QGen PL**, we treat QGen q-d pairs synthesized on the target domain as relevant, and train the reranker on both MS MARCO and target QGen q-d pairs (PL as in these pairs are "pseudolabeled" by QGen as relevant). This method is specific to text ranking, and underlies several recent works on domain adaptation of text retrievers and rerankers (Ma et al., 2021; Sun et al., 2021; Wang et al., 2022).

Also, to illustrate the deficiencies of *item-level* invariant representations for domain adaptation on ranking, we compare ListDA to the variant method that learns item-level invariant features, referred to as **ItemDA**. It uses a three-layer MLP discriminator (no improvements from going larger) and aims for $\mu_S^V = \mu_T^V$ rather than $\mu_S^Z = \mu_T^Z$ as discussed in Section 3. This method ignores the list structure of the data; yet, it has been applied for (unsupervised) domain adaptation on ranking in (Cohen et al., 2018; Tran et al., 2019; Xin et al., 2022).

Experiment details, including hyperparameter settings and the construction of training example lists, are relegated to Appendices B.1 and B.2. We also include case studies, and additional results with the pairwise logistic ranking loss and a hybrid method that combines ListDA and QGen PL.

[7] All adaptation methods (excluding zero-shot) are applied on each source-target domain pair separately.

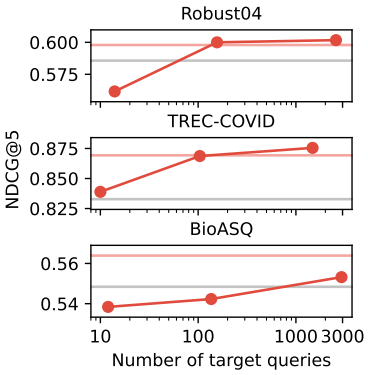


Figure 2: ListDA under different target sizes. Lower grey horizontal line is zero-shot, upper red line is ListDA using all QGen queries.

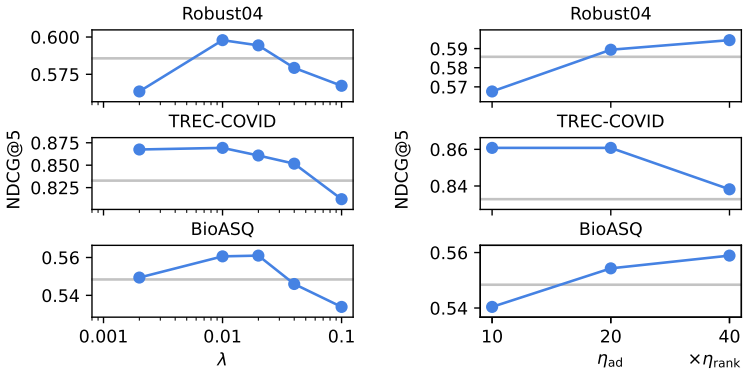


Figure 3: ListDA performance under different hyperparameter settings for λ and η_{ad}. Grey horizontal line is zero-shot. On the left, $\eta_{\text{ad}} = 0.004$ is fixed and λ varies. On the right, $\lambda = 0.02$ is fixed and η_{ad} varies.

5.1 Results

The main results are presented in Table 1, where we report metrics that are standard in the literature (e.g., TREC-COVID uses NDCG@20), and evaluate rank assignments by the descending order of the predicted scores. Since TREC-COVID and Robust04 are annotated with 3-level relevancy, the scores are binarized for mean average precision (MAP) and MRR as follows: on TREC-COVID, we map 0 (not relevant) and 1 (partially relevant) to negative, and 2 (fully relevant) to positive; on Robust04, 0 (not relevant) to negative, and 1 (relevant) and 2 (highly relevant) to positive.

Across all three datasets, ListDA achieves the best performance, and the fact that it uses the same training resource as QGen PL demonstrates the added benefits of list-level invariant representations for domain adaptation. In addition, the favorable comparison of ListDA to ItemDA corroborates the discussion in Section 3 that for domain adaptation on ranking problems, item-level invariance is insufficient for transferability, sometimes even resulting in negative transfer (vs. zero-shot). Rather, aiming for list-level invariance should be the more appropriate approach.

5.2 Analysis of ListDA

Last but not least, we include empirical analyses of ListDA from a practical perspective as a method for (unsupervised) domain adaptation on ranking problems.

Quality of QGen. An explanation for why ListDA outperforms QGen PL despite sharing the same resources is that the negative sampling of irrelevant q-d pairs could lead to the inclusion of false negatives in the training data. This is supported by the observation in (Sun et al., 2021) that queries synthesized by QGen lack specificity and could be relevant to many documents, implying a higher likelihood of sampling false negatives. While these false negatives will be treated as true negatives and trained on by QGen PL, they are not assumed by ListDA, which is thereby less likely to be affected by false negatives, or even false positives—when synthesized queries turn out to be irrelevant to the input passages (see Table 9 for samples). Although out of the scope, we expect both QGen PL and ListDA to benefit from better query generation.

Size of Target Data. Unsupervised domain adaptation requires sufficient unlabeled data, but not all domains have the same amount: BioASQ has 14 million documents (also the total number of QGen queries, as we synthesize one per document), but Robust04 only 528,155, and TREC-COVID 171,332. In Fig. 2, we plot the performance of ListDA under varying numbers of target QGen queries (also the number of target training lists). Surprisingly, on Robust04 and TREC-COVID, using just ~100 target QGen queries (0.03% and 0.06% of all, respectively) is sufficient for ListDA to achieve full performance! Although the number of queries is small, since 1,000 documents are retrieved per query, the total number of distinct target documents is still substantial—up to 100,000, or 29.5% and 60.7% of the respective corpora. The performance begins to drop when reduced to ~10 queries, capping the number of documents at 10,000 (2.7% and 5.8%, respectively). The same data efficiency, however, is not observed on BioASQ, likely due to the hardness of the dataset from e.g. the extensive use of specialized biomedical terms (Tables 7 to 9).

Sensitivity to Hyperparameters. ListDA introduces two main hyperparameters for the discriminator f_{ad}: the learning rate η_{ad} and the strength of invariant feature learning λ. We plot in Fig. 3 the sensitivity of their settings by fixing one and varying the other. It is observed that a balanced choice of λ is needed to elicit the best performance from ListDA, but the same choice largely works well across datasets. We set η_{ad} to be multiples of the reranker learning rate η_{rank}, and the results show that each dataset prefers different settings of η_{ad}, probably due to their distinct domain characteristics.

6 Related Work

Learning to Rank and Text Ranking. Traditional learning to rank focuses on tabular datasets with numerical features, for which, a wide array of models are developed (Liu, 2009), ranging 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 research direction is the design of ranking losses (surrogate to ranking metrics), which are categorized into pointwise, pairwise, and listwise approaches (Cao et al., 2007; Bruch et al., 2020; Zhu and Klabjan, 2020; Jagerman et al., 2022a).

Recent advances in large neural language models have spurred interest in applying them on text ranking tasks (Lin et al., 2022), leading to cross-attention models (Han et al., 2020; Nogueira and Cho, 2020; Nogueira et al., 2020; Pradeep et al., 2021) and generative models based on query likelihood (dos Santos et al., 2020; Zhuang and Zuccon, 2021; Zhuang et al., 2021; Sachan et al., 2022). A different line of work is neural text retrieval models, which emphasizes efficiency, and has seen the development of dual-encoder (Karpukhin et al., 2020; Zhan et al., 2021), late-interaction (Khattab and Zaharia, 2020; Hui et al., 2022), and models based on transformer memory (Tay et al., 2022).

Domain Adaptation in Information Retrieval. Work on this subject is categorized into supervised and unsupervised domain adaptation. The former assumes access to labeled source data and (a small amount of; few-shot) labeled target data (Sun et al., 2021). The present work focuses on the latter, only assuming access to unannotated target documents. Cohen et al. (2018) 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. However, unlike our ListDA, their method learns item-level invariant representations. Another family of methods is based on query generation (Ma et al., 2021; Wang et al., 2022), originally proposed for dense retrieval.

Invariant Representation Learning. Learning (adversarial) domain-invariant feature representations underlies a popular family of domain adaptation methods (Long et al., 2015; Ganin et al., 2016; Courty et al., 2017), to which our ListDA also belongs. Besides ranking, these methods are also applied in fields including vision and language, 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).

Recently, Zhao et al. (2019) and Tachet des Combes et al. (2020) point out that on classification problems, achieving perfect (marginal) feature alignment and high source accuracy are insufficient to guarantee good target performance. This occurs when the marginal distributions of labels differ, or the learned features include domain-specific components. Although their findings do not directly apply to ranking, still, they suggest two potential directions for future investigations: one is whether Theorem 3.6 would admit a fundamental lower bound under distributional shifts in the relevance scores, and the other is to explore a variant of ListDA by adding a component that could isolate nontransferable features, as in (Bousmalis et al., 2016).

7 Conclusion

In this paper, we established a domain adaptation generalization bound for ranking via analyzing representation invariance, and based on which, proposed and evaluated an adaptation method. More importantly, our results illustrate the theoretical and empirical significance of the list structure of the problem in (learning) invariant representations.

A higher-level message is that when analyzing representations, they should be considered at the same level (or structure) at which the data is defined and the metric is computed. Our work is an example, where we demonstrated the necessity of respecting the list structure when analyzing representation invariance for ranking.

Acknowledgments

This research was supported in part by the Google Visiting Researcher program and in part by the Center for Intelligent Information Retrieval. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

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 *Proceedings of the 34th International Conference on Machine Learning*, pages 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.CL]*.

Peter L. Bartlett, Dylan J. Foster, and Matus 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.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain Separation Networks. In *Advances in Neural Information Processing Systems*, 2016.

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*, pages 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*, pages 89–96, 2005.

Christopher 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*, pages 129–136, 2007.

Olivier Chapelle and Yi Chang. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 14:1–24, 2011.

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*, pages 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.

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)*, pages 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*, pages 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 J. 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 International Conference on Machine Learning*, pages 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.IR]*.

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*, pages 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*, pages 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*, pages 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*, pages 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)*, pages 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*, pages 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*, pages 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 *Proceedings of the 32nd International Conference on Machine Learning*, pages 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*, pages 1075–1088, 2021.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 10–18, 2013.

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

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*, pages 708–718, 2020.

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

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*, pages 499–508, 2020.

Robin L. Plackett. The Analysis of Permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):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.IR]*.

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*, pages 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*, pages 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.CL]*.

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*, pages 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*, pages 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)*, pages 5030–5043, 2021.

Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoff Gordon. Domain Adaptation with Conditional Distribution Matching and Generalized Label Shift. In *Advances in Neural Information Processing Systems*, 2020.

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.CL]*.

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*, pages 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(1):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*, pages 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*, pages 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*, pages 2345–2360, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

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*, pages 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*, pages 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*, pages 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.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On Learning Invariant Representations for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7523–7532, 2019.

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*, pages 798–806, 2020.

Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. Feature Transformation for Neural Ranking Models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1649–1652, 2020.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses, 2022. *arXiv:2210.10634 [cs.IR]*.

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*, pages 1483–1492, 2021.

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

A Omitted Proofs

Before proving the generalization bounds for binary classification (Theorem 2.3) and ranking (Theorem 3.6), recall the following properties of Lipschitz functions:

Fact A.1 (Properties of Lipschitz Functions)**.**

1. *$f : \mathbb{R}^d \to \mathbb{R}$ is differentiable, then it is (Euclidean) L-Lipschitz if and only if $\|\nabla f\|_2 \le L$.*
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 $z \coloneqq (1 - t)x + ty$, so by Cauchy-Schwarz,

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

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

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

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

For the second,

$$|af(x) + bg(x) - (af(y) + bg(y))| \le |a||f(x) - f(y)| + |b||g(x) - g(y)| \le (|a|L + |b|M)d_{\mathcal{X}}(x, y).$$

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

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

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

We first prove Theorem 2.3 as a warm-up, because it shares the same organization with the proof of our main result, Theorem 3.6 (but the analyses will be different).

Proof of Theorem 2.3. Define random variable $\eta \coloneqq \mathbb{1}(Y = 1)$, then $\mathcal{R}(f) = \mathbb{E}_{(X,Y)\sim\mu}[\eta - (2\eta - 1)f(X)]$. Note that

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

because $2\eta - 1 \in \{-1, +1\}$. On the other hand,

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

Then by Fact A.1, the fact that taking absolute value is 1-Lipschitz, and Definition 2.4, for all $f, f' \in \mathcal{F}$,

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

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

Next, we prove Theorem 3.6. The main idea behind the proof is that under our setup and assumptions, $\mathcal{R}_S$ and $\mathcal{R}_T$ can be written as expectations of Lipschitz functions of $Z \sim \mu_S^Z$ and μ_T^Z, respectively, so by Definition 2.4 their difference is upper bounded by $W_1(\mu_S^Z, \mu_T^Z)$ times the Lipschitz constant.

While omitted, Theorem 3.6 can be extended to the cutoff version of the ranking metric u with a simple modification of the proof. Also, a finite sample generalization bound could be obtained using e.g. Rademacher complexity and additionally assuming Lipschitzness of the end-to-end scorer (Ben-David et al., 2007; Shalev-Shwartz and Ben-David, 2014).

Proof of Theorem 3.6. Fix $g \in \mathcal{G}$, which by Assumption 3.5, when restricted to $\mathrm{supp}(\mu_S^X)$, has an L_g-Lipschitz inverse g^{-1}. Define function $\epsilon_{h,g} : \mathcal{Z} \to \mathbb{R}_{\ge 0}$ for a given $h : \mathcal{Z} \to \mathbb{R}^\ell$ as

$$\begin{aligned}
\epsilon_{h,g}(z) &\coloneqq \max_{r\in S_\ell} u(r, y_S \circ g^{-1}(z)) - \mathbb{E}_{R\sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))] \qquad (2) \\
&= \max_{r\in S_\ell} u(r, y_S \circ g^{-1}(z)) - \sum_{r\in S_\ell} u(r, y_S \circ g^{-1}(z)) \prod_{i=1}^{\ell} \frac{\exp(h(z)_{I(r)_i})}{\sum_{j=i}^{\ell} \exp(h(z)_{I(r)_j})},
\end{aligned}$$

and note that $\mathcal{R}_S(h\circ g) = \mathbb{E}_{X\sim\mu_S^X}[\epsilon_{h,g}(g(X))] =: \mathbb{E}_{Z\sim\mu_S^Z}[\epsilon_{h,g}(z)]$. Analogous statements and analyses in the following discussion hold for $\mathcal{R}_T$.

We show that $\epsilon_{h,g}$ is Lipschitz provided that h is Lipschitz, from establishing the Lipschitzness of both terms in Eq. (2). 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 in turns $L_y L_g$-Lipschitz in z by Assumptions 3.2 and 3.3, $z \mapsto u(r, y_S \circ g^{-1}(z))$ is $L_u L_y L_g$-Lipschitz in z for any fixed $r \in S_\ell$, and by Fact A.1 so is $z \mapsto \max_{r\in S_\ell} u(r, y_S \circ g^{-1}(z))$.

Next, we show that the second term is (Euclidean) Lipschitz in both $y_S \circ g^{-1}(z) =: y$ and $h(z) =: s$. By Jensen's inequality and Fact A.1,

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

And by the definition that $\nabla_x f(x) = [\frac{\partial}{\partial x_1} f(x), \cdots, \frac{\partial}{\partial x_d} f(x)]$ for any $f : \mathbb{R}^d \to \mathbb{R}$,

$$
\begin{aligned}
\|\nabla_s \mathbb{E}_{R\sim p_{\exp(s)}}[u(R,y)]\|_2 &\le \|\nabla_s \mathbb{E}_{R\sim p_{\exp(s)}}[u(R,y)]\|_1 \\
&= \sum_{m=1}^{\ell} \left| \sum_{r\in S_\ell} u(r,y) \left(\frac{\partial}{\partial s_{I(r)_m}} \prod_{i=1}^{\ell} \frac{\exp(s_{I(r)_i})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \right| \\
&= \sum_{m=1}^{\ell} \left| \sum_{r\in S_\ell} u(r,y) \sum_{i=1}^{\ell} \left(\frac{\partial}{\partial s_{I(r)_m}} \frac{\exp(s_{I(r)_i})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \prod_{n\neq i} \frac{\exp(s_{I(r)_n})}{\sum_{j=n}^{\ell} \exp(s_{I(r)_j})} \right| \\
&= \sum_{m=1}^{\ell} \sum_{r\in S_\ell} u(r,y) \left(\prod_{n=1}^{\ell} \frac{\exp(s_{I(r)_n})}{\sum_{j=n}^{\ell} \exp(s_{I(r)_j})} \right) \left| \sum_{i=1}^{m} \left(\mathbb{1}(m=i) - \frac{\exp(s_{I(r)_m})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \right| \\
&\le B \sum_{m=1}^{\ell} \sum_{r\in S_\ell} \left(\prod_{n=1}^{\ell} \frac{\exp(s_{I(r)_n})}{\sum_{j=n}^{\ell} \exp(s_{I(r)_j})} \right) \sum_{i=1}^{m} \left(\mathbb{1}(m=i) + \frac{\exp(s_{I(r)_m})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \\
&= B \sum_{r\in S_\ell} \left(\prod_{n=1}^{\ell} \frac{\exp(s_{I(r)_n})}{\sum_{j=n}^{\ell} \exp(s_{I(r)_j})} \right) \sum_{i=1}^{\ell} \sum_{m=i}^{\ell} \left(\mathbb{1}(m=i) + \frac{\exp(s_{I(r)_m})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \\
&= B \sum_{r\in S_\ell} \left(\prod_{n=1}^{\ell} \frac{\exp(s_{I(r)_n})}{\sum_{j=n}^{\ell} \exp(s_{I(r)_j})} \right) \left(\ell + \sum_{i=1}^{\ell} \frac{\sum_{m=i}^{\ell} \exp(s_{I(r)_m})}{\sum_{j=i}^{\ell} \exp(s_{I(r)_j})} \right) \\
&= 2B\ell \sum_{r\in S_\ell} \left(\prod_{n=1}^{\ell} \frac{\exp(s_{I(r)_n})}{\sum_{j=n}^{\ell} \exp(s_{I(r)_j})} \right) \\
&= 2B\ell,
\end{aligned}
$$

where the second equality is due to the product rule, the third equality to the identity $\frac{\mathrm{d}}{\mathrm{d}x_j}\mathrm{softmax}(x)_i = \mathrm{softmax}(x)_i(\mathbb{1}(i=j) - \mathrm{softmax}(x)_j)$, the last inequality to Assumption 3.2, and the last equality to recognizing the pmf of the P-L model (Definition 3.1). Provided that $h \in \mathrm{Lip}(L_h)$ by Assumption 3.4, the two results above imply that the second term in Eq. (2) is Lipschitz: for all $z, z' \in \mathcal{Z}$,

$$
\begin{aligned}
&\left| \mathbb{E}_{R\sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))] - \mathbb{E}_{R\sim p_{\exp(h(z'))}}[u(R, y_S \circ g^{-1}(z'))] \right| \\
&\quad \le \left| \mathbb{E}_{R\sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z))] - \mathbb{E}_{R\sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z'))] \right| \\
&\qquad\quad + \left| \mathbb{E}_{R\sim p_{\exp(h(z))}}[u(R, y_S \circ g^{-1}(z'))] - \mathbb{E}_{R\sim p_{\exp(h(z'))}}[u(R, y_S \circ g^{-1}(z'))] \right| \\
&\quad \le L_u \|y_S \circ g^{-1}(z) - y_S \circ g^{-1}(z')\|_2 + 2B\ell \|h(z) - h(z')\|_2 \\
&\quad \le (L_u L_y L_g + 2B\ell L_h) d_{\mathcal{Z}}(z, z').
\end{aligned}
$$

Combining the two parts, we have that $\epsilon_{h,g}$ is $2(L_u L_y L_g + B\ell L_h)$-Lipschitz in z.

Finally, by Fact A.1 and Definition 2.4, for all $g \in \mathcal{G}$ and $h, h' \in \mathcal{H}$,

$$\begin{aligned}
\mathcal{R}_T(h \circ g) &= \mathcal{R}_S(h \circ g) + \big(\mathcal{R}_T(h \circ g) - \mathcal{R}_T(h' \circ g)\big) - \big(\mathcal{R}_S(h \circ g) + \mathcal{R}_S(h' \circ g)\big) \\
&\qquad + \big(\mathcal{R}_S(h' \circ g) + \mathcal{R}_T(h' \circ g)\big) \\
&\le \mathcal{R}_S(h \circ g) + \big(\mathcal{R}_T(h \circ g) - \mathcal{R}_T(h' \circ g)\big) - \big(\mathcal{R}_S(h \circ g) - \mathcal{R}_S(h' \circ g)\big) \\
&\qquad + \big(\mathcal{R}_S(h' \circ g) + \mathcal{R}_T(h' \circ g)\big) \\
&= \mathcal{R}_S(h \circ g) + \mathbb{E}_{Z \sim \mu_T^Z}[\epsilon_{h,g}(Z) - \epsilon_{h',g}(Z)] - \mathbb{E}_{Z \sim \mu_S^Z}[\epsilon_{h,g}(Z) - \epsilon_{h',g}(Z)] \\
&\qquad + \big(\mathcal{R}_S(h' \circ g) + \mathcal{R}_T(h' \circ g)\big) \\
&\le \mathcal{R}_S(h \circ g) + \sup_{q \in \mathrm{Lip}(4(L_u L_y L_g + B\ell L_h))} (\mathbb{E}_{Z \sim \mu_T^Z}[q(Z)] - \mathbb{E}_{Z \sim \mu_S^Z}[q(Z)]) \\
&\qquad + \big(\mathcal{R}_S(h' \circ g) + \mathcal{R}_T(h' \circ g)\big) \\
&\le \mathcal{R}_S(h \circ g) + 4(L_u L_y L_g + B\ell L_h) \cdot W_1(\mu_S^Z, \mu_T^Z) + \big(\mathcal{R}_S(h' \circ g) + \mathcal{R}_T(h' \circ g)\big),
\end{aligned}$$

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

In the following, we verify the Lipschitzness of RR and NDCG.

Proof of Corollary 3.7. Because $\mathrm{RR} \le 1$ uniformly and $\|y - y'\|_2 \ge 1$ for all $y, y' \in \{0,1\}^\ell, y \ne y'$, so $y \mapsto \mathrm{RR}(r, y)$ is 1-Lipschitz. □

Proof of Corollary 3.8. We show that

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

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 for any $k \in \{1, \cdots, \ell\}$,

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

so NDCG is $\sqrt{\ell}(u_{\min}^{-1} + u_{\max} \log(\ell + 1)^2)$-Lipschitz by Fact A.1. □

Proof of Proposition 4.1. First,

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

because $\int \mu_T^Z(z) - \mu_S^Z(z) \,\mathrm{d}z = 0$. Note that the last term is the total variation between μ_S^Z, μ_T^Z.

On the other hand, define $\hat{Y}(z) \coloneqq \mathbb{1}(f_{\mathrm{ad}}(z) \geq 0)$. Then the balanced total error rate of $\hat{Y}$ on predicting the domain identities is

$$\mathrm{Err}(\hat{Y}) \coloneqq \int_{\mathcal{Z}} \Bigl(\hat{Y}(z)\mu_S^Z(z) + (1 - \hat{Y}(z))\mu_T^Z(z)\Bigr) \,\mathrm{d}z = 1 + \int_{\mathcal{Z}} \left(\hat{Y}(z) - \frac{1}{2}\right)\bigl(\mu_S^Z(z) - \mu_T^Z(z)\bigr) \,\mathrm{d}z.$$

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

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

The result then follows from an algebraic rearrangement of the terms. □

Finally, we give an example to illustrate the discussion in Theorem 3.6 that item-level alignment does not necessarily imply list-level alignment.

Example A.2. Consider two uniform distributions μ_S^Z, μ_T^Z over lists of length three, each supported on two lists:

$$\begin{aligned}
\mathrm{supp}(\mu_S^Z) &= \{(1, 2, 3), (4, 5, 6)\}, \\
\mathrm{supp}(\mu_T^Z) &= \{(1, 3, 5), (2, 4, 6)\}.
\end{aligned}$$

Then the item-level feature distributions μ_S^V, μ_T^V derived from the respective list-level distributions are the same uniform distribution supported on four items:

$$\mathrm{supp}(\mu_S^V) = \mathrm{supp}(\mu_T^V) = \{1, 2, 3, 4, 5, 6\}.$$

Note that item-level features are aligned since $\mu_S^V = \mu_T^V$, but list-level features are not, because $\mathrm{supp}(\mu_S^Z) \cap \mathrm{supp}(\mu_T^Z) = \emptyset$.

B Additional Experiments on Passage Reranking and Details

In this section, additional experiment results for unsupervised domain adaptation on the passage reranking task considered in Section 5 are provided, along with case studies on ListDA vs. zero-shot and QGen PL (Tables 8 and 9), hyperparameter settings (Appendix B.1) and details on the construction of training example lists (Appendix B.2).

Table 2: Transfer performance of RankT5 on top 1000 BM25-retrieved passages; in addition to Table 1 results.

(a) With pairwise logistic ranking loss in place of softmax cross-entropy on Robust04.

Target Domain	Method	MAP	MRR@10	NDCG@5	NDCG@10	NDCG@20
Robust04	Zero-shot	0.2656	0.7894	0.5671	0.5163	0.4729
	QGen PL	0.2776*	0.7975	0.5576	0.5267	0.4892*
	ItemDA	0.2766*	0.8021	0.5794	0.5355*	0.4917*
	ListDA	**0.2893***†‡	**0.8103**	**0.5935***†‡	**0.5524***†‡	**0.5044***†‡

Source domain is MS MARCO. Gain function in NDCG is the identity map. *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.

(b) With ListDA + QGen PL method.

Target Domain	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*‡

Source domain is MS MARCO. Gain function in NDCG is the identity map. *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.

Pairwise Logistic Ranking Loss. Besides the listwise softmax cross-entropy loss (Eq. (1)) in Section 5, we also experimented with the pairwise logistic ranking loss (Burges et al., 2005):

$$\ell_{\text{rank}}(s, y) = -\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \mathbb{1}(y_i > y_j) \log\left(\frac{\exp(s_i)}{\exp(s_i) + \exp(s_j)}\right).$$

The results with this training loss function on the Robust04 dataset are in Table 2a. It is observed that the pairwise logistic loss does not perform better than softmax cross-entropy (cf. Table 1; see also (Jagerman et al., 2022b)), hence further experiments with this loss were not pursued.

As an implementation remark, in this set of experiments, we did not perform pairwise comparisons to obtain the predicted rank assignments during inference due to the high time complexity (the loss is still aggregated pairwise during training, but on smaller truncated lists as described in Appendix B.2). Whether or not the forward pass of the model involves pairwise computations is orthogonal to our theory, which is applicable to any (pointwise, pairwise or listwise) model as long as we can gather list-level representations. While not pursued in this work, a list-level invariant representation learning method could be instantiated on pairwise models e.g. DuoT5 (Pradeep et al., 2021), as we did for listwise models in Section 4.

ListDA + QGen PL Method and Signal-1M Dataset. We experiment with supplementing ListDA with QGen pseudolabels by (uniformly) combining the training objectives of ListDA and QGen PL methods (**ListDA + QGen PL**). The results on the three datasets considered in Section 5 are included in Table 2b. This method is also applied on the Signal-1M (RT) dataset (Suarez et al., 2018), with results in Table 3. It is noted that reranking using neural rerankers transferred from MS MARCO source domain does not perform better than BM25 on Signal-1M, which is also observed

Table 3: Transfer performance of RankT5 on top 1000 BM25-retrieved passages on Signal-1M.

Target Domain	Method	MAP	MRR@10	NDCG@5	NDCG@10	NDCG@20
Signal-1M	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

Source domain is MS MARCO. Gain function in NDCG is the identity map.

in prior work (Thakur et al., 2021; Liang et al., 2022). This does not mean that neural rerankers are worse than BM25, but that MS MARCO is not a good choice as the source domain when Signal-1M is the target because of the arguably large domain shift between tweet retrieval and MS MARCO web search—qualitatively, it can be seen from Table 7 that the text styles and task semantics of the two domains are very different. Hence, the following discussions on Signal-1M results focus on reranking models.

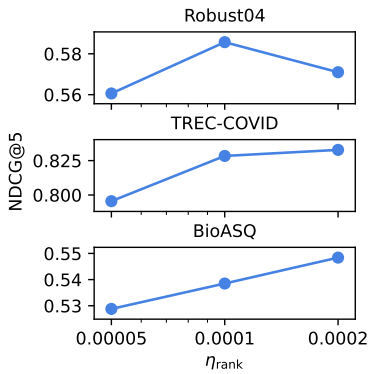
On Signal-1M, QGen PL improves upon the zero-shot baseline, but ListDA does not, which is likely due to the large domain shift between MS MARCO and Signal-1M that prevented ListDA from finding the correct source-target feature alignment without supervision. With QGen pseudolabels, ListDA performance 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.

Overall, ListDA + QGen PL is the only method that consistently improves upon the zero-shot baseline on all four datasets, including Signal-1M, although it underperforms ListDA on the other three. Further improvements to this method may be possible with better strategies for balancing the constituent training objectives of ListDA and QGen PL.

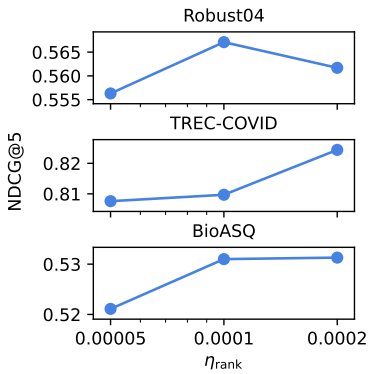
Case Studies. In Tables 8 and 9, we include examples where the ranks (top-1) predicted by models trained with ListDA have higher utilities v.s. zero-shot and QGen PL results, respectively, to provide a qualitative understanding of the benefits and advantages of ListDA.

In zero-shot learning, the reranker is trained on MS MARCO general domain data only and directly evaluated on the target domains. When the target domain differs from MS MARCO stylistically or consists of passages containing domain-specific words, as in the TREC-COVID and BioASQ datasets for biomedical retrieval, the zero-shot model, which is barely exposed to the specialized language usages, may resort to keyword matching. Examples of such cases are presented in Table 8, where the top passages returned by the zero-shot model contain keywords from the query but are irrelevant.

In QGen PL, the reranker is trained on the pseudolabels generated during the query synthesis procedure, treating the passages from which the queries are generated as relevant. However, as remarked in Section 5, because QGen is deployed in a zero-shot manner, the pseudolabels are not guaranteed to be valid, meaning that they could be false positives. This is observed in cases presented in Tables 7 and 9. One specific scenario where false positives hurt transfer performance is when the synthesized queries (of false positive documents) coincide with queries from the evaluation set, which is indeed the case presented in Table 9 on TREC-COVID and BioASQ datasets. Since ListDA does not assume the pseudolabels in its training objective, it does not suffer the same pitfalls as QGen PL.



(a) With softmax cross-entropy ranking loss.



(b) With pairwise logistic ranking loss.

Figure 4: Zero-shot performance under different hyperparameter settings for η_{rank}.

B.1 Hyperparameter Settings

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), titles are indexed as a separate field with equal weight as the document body, if available.

For the RankT5, the model is fine-tuned from the T5 v1.1 base checkpoint on a Dragonfish TPU with 8x8 topology for 100,000 steps with a batch size of 32 (each example is a list containing 31 items) per domain. We tune the learning rate $\eta_{\text{rank}} \in \{5\text{e-}5, 1\text{e-}4, 2\text{e-}4\}$, and select the one that gives the best zero-shot performance to use on all models for each dataset (see Fig. 4 for zero-shot sweep results). We apply a learning rate schedule on η_{rank} that decays (exponentially) by a factor of 0.7 every 5,000 steps. Each concatenated query-document text input is truncated to 512 tokens.

For the domain discriminators, there are two hyperparameters: the strength of invariant feature learning λ, and the discriminator learning rate η_{ad}. We tune both by sweeping $\lambda \in \{0.01, 0.02\}$, and $\eta_{\text{ad}} \in \{10, 20, 40\} \times \eta_{\text{rank}}$, multiples of the reranker learning rate (see Fig. 3 in Section 5 for ListDA sweep results). The tuned hyperparameter settings for η_{rank}, η_{ad} and λ used in our experiments are included in Table 4.

As a remark on runtime efficiency, the adaptation methods of ListDA, ItemDA and QGen PL all have double the training time compared to zero-shot learning due to data loading: in addition to source domain data, the adaptation methods also requires target domain (unlabeled) data. Among ListDA, ItemDA and QGen PL, the training times are roughly the same, because the overhead of training the domain discriminators is not significant.

B.2 Training Example List Construction

Recall from the description in Section 5 that on ranking problems, the inputs are defined over lists and the invariant representations for domain adaptation are learned at the list level. In other words, the ranking loss and the adversarial loss need to be computed on ranking scores and feature representations that the model outputs on lists of documents (containing both relevant and irrelevant ones) for each query.

Under our reranking setup, each list would be the top-r documents retrieved by BM25 on a query, and we set $r = 1000$ in our experiments. However, there are two complications. The first is that due to memory constraints, it is not feasible to feed all 1000 documents from each list

Table 4: Hyperparameter settings of RankT5 and domain discriminators.

(a) With softmax cross-entropy ranking loss.

Target Domain	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	Zero-shot	5e-5	-	-
	QGen PL		-	-
	ListDA		2e-3	0.01
	ListDA + QGen PL		1e-3	0.02

(b) With pairwise logistic ranking loss.

Target Domain	Method	η_{rank}	η_{ad}	λ
Robust04	Zero-shot	1e-4	-	-
	QGen PL		-	-
	ItemDA		2e-3	0.01
	ListDA		2e-3	0.01

simultaneously through the T5 encoder during training. The second is that the source domain MS MARCO dataset only contains annotations of one relevant document per query, meaning that out of the 1000 documents retrieved by BM25 for each query, we would only know that one of them is relevant; the ground-truth relevance scores for the remaining 999 documents are unknown.

Example List Construction with Negative Sampling. To address both issues, we truncate the list to length $\ell = 31$ during training and perform random negative sampling from the BM25 results to gather irrelevant documents.

On the MS MARCO source domain, given a query q and top 1000 documents retrieved by BM25, $d_1, \cdots, d_{1000}$, we construct the example list with the one document d^* labeled as relevant in the dataset along with 30 randomly sampled documents $d_{N_1}, \cdots, d_{N_{30}}$ treated as irrelevant, and get $x = ([q, d^*], [q, d_{N_1}], \cdots, [q, d_{N_{30}}])$ and $y = (1, 0, \cdots, 0)$.

For the target domains, we perform the same procedure. Given a QGen synthesized query and the top 1000 documents retrieved by BM25, the example list consists of the pseudolabeled document $\hat{d}$ (i.e., the document with which the query was synthesized) and 30 randomly sampled irrelevant documents, so that $x = ([q, \hat{d}], [q, d_{N_1}], \cdots, [q, d_{N_{30}}])$ and $y = (1, 0, \cdots, 0)$. Note that the

pseudolabels y are used by QGen PL but discarded by ListDA.

Reducing MS MARCO False Negatives. One potential problem that arises from negative sampling is that the constructed lists may contain false negatives (i.e., relevant documents that are incorrectly marked as irrelevant); in fact, false negatives are prevalent in the MS MARCO dataset. While these false negatives are mostly harmless for source domain supervised training because the effects are canceled out by training on the true positive documents to which they are similar with positive labels, they negatively affect the alignment of the source and target domain features in unsupervised invariant representation learning.

One such negative effect is that the duplicates in the lists (which have identical feature vectors) will cause ListDA to collapse distinct documents on the target domain to the same feature vector for achieving alignment, which is an artifact that will cause information loss in the target domain feature representations. Another is that the inclusion of duplicates may alter the marginal distribution of scores on the source domain (see the discussion on (Zhao et al., 2019) in Section 6), resulting in complications for feature alignment.

To lower the chance of selecting false negatives on MS MARCO, we rerank each BM25-retrieved result using a ranker that is pre-trained on MS MARCO, and sample negatives from documents that are ranked at 300 or higher, since the duplicates and relevant documents will be concentrated at the top (Qu et al., 2021). We only apply this method when constructing source domain lists for feature alignment (namely, in ListDA and ItemDA), and it only affects the computation of adversarial loss. This sampling method does not apply to target domains because we do not have reliable pre-trained rankers. It is also not used for source domain supervised training (i.e., the computation of ranking loss), as reduced performance was observed in our preliminary experiments with this method, likely due to the exclusion of "hard" negatives.

C Experiments on Yahoo! LETOR

Our method is also evaluated on the ranking task from the Yahoo! Learning to Rank Challenge v2.0 (Chapelle and Chang, 2011). This is a web search ranking dataset in numerical format, where each item is represented by a 700-d vector with values in the range of $[0, 1]$. It has two subsets, called "Set 1" and "Set 2", whose data originate from the US and an Asian country, respectively. Among the 700 features, 415 are defined on both sets (shared), and the other 285 are defined only on Set 1 or 2 only (disjoint); we hence write each item $x \coloneqq [x_{\text{shared}}, x_{\text{disjoint}}]$ as a concatenation of shared features $x_{\text{shared}} \in \mathbb{R}^{415}$ and disjoint ones $x_{\text{disjoint}} \in \mathbb{R}^{285}$.

We consider unsupervised domain adaptation from Set 1 to Set 2. Our implementation uses the Hugging Face Transformers library (Wolf et al., 2020).

Models. Our models have the same setup as that of the passage reranking experiments in Section 5, except that the RankT5 text model is replaced by a (generic) three-hidden-layer MLP following (Zhuang et al., 2020), where we treat the list of 256-d outputs on the last hidden layer as feature representations:

$$
\begin{aligned}
g(x)_i =: v_i &= \mathrm{ReLU}\left(W_3 \begin{bmatrix} \mathrm{ReLU}(W_2 \mathrm{ReLU}(W_1 x_{i,\text{shared}} + b_1) + b_2) \\ \mathrm{ReLU}(W_2' \mathrm{ReLU}(W_1' x_{i,\text{disjoint}} + b_1') + b_2') \end{bmatrix} + b_3\right), \\
h(x)_i =: s_i &= W_4 v_i + b_4,
\end{aligned}
$$

where $W_1 \in \mathbb{R}^{1024 \times 415}$, $W_1' \in \mathbb{R}^{1024 \times 285}$, $W_2, W_2' \in \mathbb{R}^{256 \times 1024}$, $W_3 \in \mathbb{R}^{256 \times 512}$, and $W_4 \in \mathbb{R}^{1 \times 256}$, all randomly initialized.

Table 5: Transfer performance of 3-layer MLP ranker on Yahoo! LETOR (Set 2).

Target Domain	Method	MAP	MRR@10	NDCG@5	NDCG@10	NDCG@20
Yahoo! LETOR (Set 2)	Zero-shot	0.5138	0.6558	0.7302	0.7627	0.8199
	Supervised	0.5389	0.6785	0.7523	0.7829	0.8341
	ItemDA	0.5315*	0.6717*	0.7402*	0.7708*	0.8255*
	ListDA	**0.5370***‡	**0.6771***‡	**0.7442***‡	**0.7735***‡	**0.8269***‡

Source domain is Yahoo! LETOR (Set 1). Gain function in NDCG is the identity map. The best unsupervised results are in bold. Results are from ensembles of five models. *Improves upon zero-shot baseline with statistical significance ($p \leq 0.05$) under the two-tailed Student's t-test. ‡Improves upon ItemDA. Significance tests are not performed on supervised results.

Table 6: Hyperparameter settings of 3-layer MLP ranker and domain discriminators.

Target Domain	Method	η_{rank}	η_{ad}	λ
Yahoo! LETOR (Set 2)	Zero-shot	4e-4	-	-
	Supervised	4e-5	-	-
	ItemDA	2e-4	4e-4	0.4
	ListDA	2e-4	1.6e-3	0.1

Each model in the ensemble of five domain discriminators is a stack of three T5 encoder transformer blocks, with 4 attention heads (`num_heads`), size-32 key, query and value projections per attention head (`d_kv`), and size-1024 intermediate feedforward layers (`d_ff`).

Results. The results are presented in Table 5. Considering the small dataset size and number of training steps, each method is evaluated by an ensemble of five separately trained models to reduce the variance in the results due to the randomness in the initialization and the training process. Since Yahoo! LETOR is annotated with 5-level relevancy, the scores are binarized for MAP and MRR metrics by mapping 0 (bad) and 1 (fair) to negative, and 2 (good), 3 (excellent), 4 (perfect) to positive. Thanks to the availability of labeled data on Set 2, we also include **supervised** results as an upper bound for unsupervised domain adaptation, where the model is trained on labeled data from both Set 1 and 2.

ListDA achieves the best unsupervised transfer performance. In particular, the favorable comparison of ListDA to ItemDA again corroborates our discussion in Section 3 that list-level invariant representation learning is more appropriate on ranking problems (although their gap is smaller compared to the passage reranking results in Table 1, which we suspect is because the contextual (query) information for defining the list structure of the data is too weak in this numerical dataset).

Hyperparameters. The model is trained from scratch on an NVIDIA A6000 GPU for 5,000 steps with a batch size of 32 (each example is a list containing one to no more than 140 items) per domain. We apply a learning rate schedule on η_{rank} that decays (exponentially) by a factor of 0.7 every 500 steps.

We exhaustively tune the learning rates of the ranker and the domain discriminator, together with the strength of feature alignment λ, by performing grid search over their combinations: $\eta_{\text{rank}} \in \{$1e-5, 2e-5, 4e-5, 8e-5, 1e-4, 2e-4, 4e-4, 8e-4, 1e-3$\}$, $\eta_{\text{ad}} \in \{0.2, 0.4, 0.8, 1, 2, 4, 8, 10\} \times \eta_{\text{rank}}$, and $\lambda \in \{0.01, 0.02, 0.04, 0.08, 0.1, 0.2\}$. The tuned settings are included in Table 6.

Table 7: Samples of test set relevant q-d pairs and QGen synthesized q-d pairs from domains considered in Section 5 passage reranking experiments. Truncated or omitted texts are indicated by "[...]".

Dataset	Ground-Truth Relevant Q-D Pairs	QGen Q-D Pairs
MS MARCO	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 is the science of mapmaking called 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. Q: what's in the flu shot	-
TREC-COVID	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 known about an mRNA vaccine for the SARS-CoV-2 virus? 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 the mechanism of cytokine storm syndrome on the COVID-19?	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. [...] QGen: what is arterial load for pulse pressure analysis 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 [...] QGen: opportunity cost pacifism
BioASQ	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: Is amantadine ER the first approved treatment for akinesia? 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: Can leuprorelin acetate be used as androgen deprivation therapy?	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. [...] QGen: average age of femoral subluxation 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' [...] QGen: which scale is used for proxy assessment of hrqol
Signal-1M	D: BJP terms party MP R.K Singh's allegation that money has changed hands for tickets in #BiharPolls as baseless. Q: Party MP calls BJP 'Baura Jayewala Party' D: Kerry: US plans military talks with Russia over Syria Q: Kerry: US plans military talks with Russia over Syria	D: Black lives matter: thoughts from the delivery ward in St. Louis: #mustread QGen: where is black lives matter? D: RETWEET if "Brenda's Got A Baby" is one of your favorite @2Pac songs. #RIP2Pac QGen: brenda got a baby pac

Table 8: Samples of passage reranking results where ListDA achieves higher utilities v.s. zero-shot. Truncated or omitted texts are indicated by "[...]".

Dataset	Zero-Shot Top Results (Ground-Truth Irrelevant)	ListDA Top Results (Ground-Truth Relevant)
Robust04	Q: Find information on prostate cancer detection and treatment.	
	D: [...] FIRST PATIENT UNDERGOES GENE INSERTION IN CANCER TREATMENT [...] This first round of gene transfer experiments, in which a gene was inserted into a patient's white blood cells, is not expected to directly benefit an individual patient. Instead, the inserted gene is being used to track the movement in the body of the cancer-fighting white blood cells. [...] Inserting human genes to repair defects may one day help with a host of inherited disorders, [...]	D: [...] Little knowledge goes a long way - Cancer of the prostate need not be a killer / Health Check. Earlier this year, 13-year-old [...] died from cancer of the bladder and prostate. His death is a grim reminder that no male should consider himself immune from waterworks trouble. The prostate, a gland about the size and shape of a chestnut, lies deep in the pelvis just below the bladder. Because it surrounds the urethra, it has the potential to block the flow of urine completely. [...]
TREC-COVID	Q: What are the longer-term complications of those who recover from COVID-19?	
	D: [...] Our previous experience with members of the same corona virus family (SARS and MERS) which have caused two major epidemics in the past albeit of much lower magnitude, has taught us that the harmful effect of such outbreaks are not limited to acute complications alone. Long term cardiopulmonary, glucometabolic and neuropsychiatric complications have been documented following these infections. [...]	D: Up to 20-30% of patients hospitalized with coronavirus disease (COVID-19) have evidence of myocardial involvement. Acute cardiac injury in patients hospitalized with COVID-19 is associated with higher morbidity and mortality. There are no data on how acute treatment for COVID-19 may affect convalescent phase or long-term cardiac recovery and function. Myocarditis from other viral pathogens can evolve into overt or subclinical myocardial dysfunction, [...]
BioASQ	Q: What is the interaction between WAPL and PDS5 proteins?	
	D: Pds5 and Wpl1 act as anti-establishment factors preventing sister-chromatid cohesion until counteracted in S-phase by the cohesin acetyl-transferase Eso1. [...] Here, we show that Pds5 is essential for cohesin acetylation by Eso1 and ensures the maintenance of cohesion by promoting a stable cohesin interaction with replicated chromosomes. The latter requires Eso1 only in the presence of Wapl, indicating that cohesin stabilization relies on Eso1 only to neutralize the anti-establishment activity. [...]	D: [...] Here, we show that cohesin suppresses compartments but is required for TADs and loops, that CTCF defines their boundaries, and that the cohesin unloading factor WAPL and its PDS5 binding partners control the length of loops. In the absence of WAPL and PDS5 proteins, cohesin forms extended loops, presumably by passing CTCF sites, accumulates in axial chromosomal positions (vermicelli), and condenses chromosomes. [...]

Table 9: Samples of passage reranking results where ListDA achieves higher utilities v.s. QGen PL. Truncated or omitted texts are indicated by "[...]".

Dataset	QGen PL Top Results (Ground-Truth Irrelevant)	ListDA Top Results (Ground-Truth Relevant)
Robust04	Q: Identify outbreaks of Legionnaires' disease.	
	D: [...] 3. Care of Patients with Tracheostomy 4. Suctioning of Respiratory Tract Secretions III. Modifying Host Risk for Infection A. Precautions for Prevention of Endogenous Pneumonia 1. Prevention of Aspiration 2. Prevention of Gastric Colonization B. Prevention of Postoperative Pneumonia C. Other Prophylactic Procedures for Pneumonia 1. Vaccination of Patients 2. Systemic Antimicrobial Prophylaxis 3. Use of Rotating "Kinetic" Beds Prevention and Control of Legionnaires' Disease [...] QGen: what kind of precautions are used to prevent pneumonia	D: [...] LEGIONNAIRE'S DISEASE STRIKES 16 AT REUNION IN COLORADO; 3 DIE. An outbreak of legionnaire's disease at a 50th high school reunion was blamed Thursday for the deaths of three elderly celebrants and the pneumonia-like illness of 13 others. State health officials contacted 250 other people from 21 states who attended the Lamar High School reunion for the classes of 1937 through 1941 but found no new cases, Dr. Ellen Mangione, a Colorado Department of Health epidemiologist, said. [...]
TREC-COVID	Q: what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?	
	D: Different treatments are currently used for clinical management of SARS-CoV-2 infection, but little is known about their efficacy yet. Here we present ongoing results to compare currently available drugs for a variety of diseases to find out if they counteract SARS-CoV-2-induced cytopathic effect in vitro. [...] We will provide results as soon as they become available, [...] QGen: what is the treatment for sars	D: [...] the antiviral efficacies of lopinavir-ritonavir, hydroxychloroquine sulfate, and emtricitabine-tenofovir for SARS-CoV-2 infection were assessed in the ferret infection model. [...] all antiviral drugs tested marginally reduced the overall clinical scores of infected ferrets but did not significantly affect in vivo virus titers. Despite the potential discrepancy of drug efficacies between animals and humans, these preclinical ferret data should be highly informative to future therapeutic treatment of COVID-19 patients.
BioASQ	Q: What is the function of the Spt6 gene in yeast?	
	D: As a means to study surface proteins involved in the yeast to hypha transition, human monoclonal antibody fragments (single-chain variable fragments, scFv) have been generated that bind to antigens expressed on the surface of Candida albicans yeast and/or hyphae. [...] To assess C. albicans SPT6 function, expression of the C. albicans gene was induced in a defined S. cerevisiaespt6 mutant. Partial complementation was seen, confirming that the C. albicans and S. cerevisiae genes are functionally related in these species. QGen: what is the function of spt6 gene in candida albicans	D: Spt6 is a highly conserved histone chaperone that interacts directly with both RNA polymerase II and histones to regulate gene expression. [...] Our results demonstrate dramatic changes to transcription and chromatin structure in the mutant, including elevated antisense transcripts at >70% of all genes and general loss of the +1 nucleosome. Furthermore, Spt6 is required for marks associated with active transcription, including trimethylation of histone H3 on lysine 4, previously observed in humans but not Saccharomyces cerevisiae, and lysine 36. [...]