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]

[1]University of Illinois Urbana-Champaign
[2]Google Research
[3]University of Massachusetts Amherst

Paper link: https://openreview.net/forum?id=m21rQusNgb

Learning List-Level Domain-Invariant

Contributions

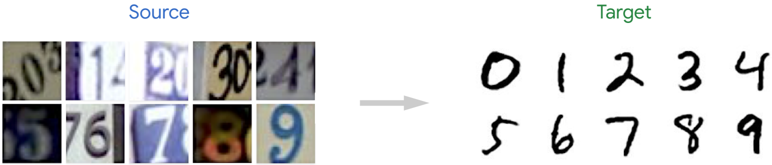
Revisit domain adaptation on ranking problems via invariant representation learning.

Whereas prior implementations perform *item-level alignment*, we

- propose *list-level alignment*;
- establish a domain adaptation generalization bound for ranking based on list-level alignment, and
- demonstrate the its empirical benefits.

Paper link: https://openreview.net/forum?id=m21rQusNgb

Domain Adaptation and Invariant Representations



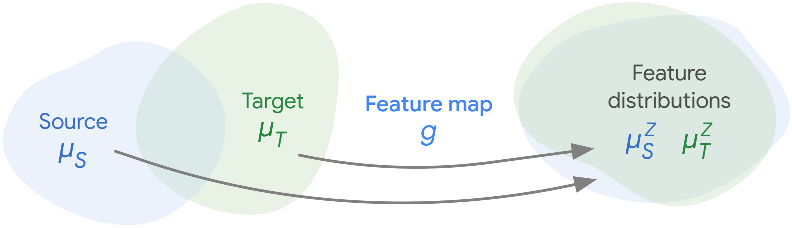
Problem Setup for Domain Adaptation

A (low-resource) target domain μ_T, and a source domain μ_S.

Goal. Train a good model for μ_T using available resources.

Example. In *unsupervised* domain adaptation, have labeled data from source domain, but only unlabeled data from target domain.

Domain Adaptation and Invariant Representations

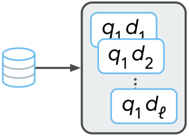


Invariant Representation Learning

1. Learn a mapping $g : \mathcal{X} \to \mathcal{Z}$ that *matches and aligns* the source and target data distributions on the feature space $\mathcal{Z}$.
2. Train the model on the learned (and *transferrable*) features.

Example. In *unsupervised* domain adaptation, the feature map is learned on unlabeled data from source and target, and the model is trained on source labeled data.

Learning to Rank



List of *q*-*d* pairs

Query: What kind of bear is best?

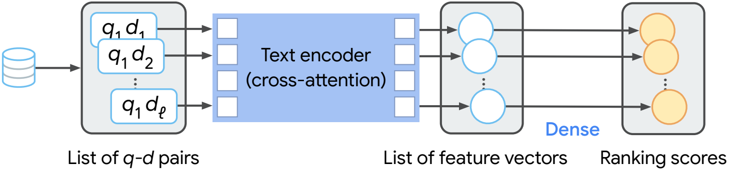
Document 1: There are basically two...
Document 2: Bears eat beets. Bears...
⋮
Document ℓ: All bear species are great...

A ranking problem is given by joint distribution μ over length-ℓ lists of items $(X_1, \cdots, X_\ell)$ and scores $(Y_1, \cdots, Y_\ell)$.

Measure model performance by ranking metrics, e.g., MRR, NDCG.

Goal. Train a ranking model f that ranks the items in agreement with the descending order of the scores.

Learning to Rank



A ranking problem is given by joint distribution μ over length-ℓ lists of items $(X_1, \cdots, X_\ell)$ and scores $(Y_1, \cdots, Y_\ell)$.

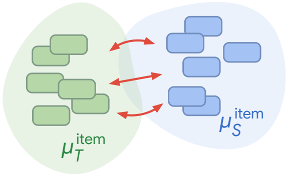
Measure model performance by ranking metrics, e.g., MRR, NDCG.

Goal′. Train a scoring model f that scores the items in agreement with the descending order of the scores.

Feature Space. Model computes a list of feature vectors, $(Z_1, \cdots, Z_\ell) \in \mathbb{R}^{\ell \times k}$, where $Z_i \in \mathbb{R}^k$ corresponds to item i.

Item-Level Alignment[1]

Feature Space. Model computes a list of feature vectors, $(Z_1, \cdots, Z_\ell) \in \mathbb{R}^{\ell \times k}$, where $Z_i \in \mathbb{R}^k$ corresponds to item i.



In item-level alignment, distributions of feature vectors aggregated from all lists are aligned, i.e., $\mu_S^{Z,\text{item}} \approx \mu_T^{Z,\text{item}}$,

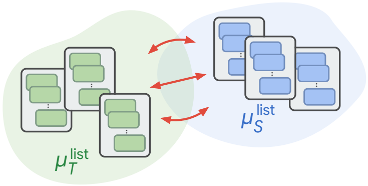
$$\operatorname{supp}(\mu^{Z,\text{item}}) \subseteq \mathbb{R}^k, \qquad \mu^{Z,\text{item}}(v) = \mathbb{P}(v \in (Z_1, \cdots, Z_\ell)).$$

List structure on data is ignored.

[1]Cohen et al., 2018; Tran et al., 2019; Xin et al., 2022.

List-Level Alignment

Feature Space. Model computes a list of feature vectors, $(Z_1, \cdots, Z_\ell) \in \mathbb{R}^{\ell \times k}$, where $Z_i \in \mathbb{R}^k$ corresponds to item i.



In list-level alignment, distributions of lists of feature vectors are aligned, i.e., $\mu_S^{Z,\text{list}} \approx \mu_T^{Z,\text{list}}$,

$$\operatorname{supp}(\mu^{Z,\text{list}}) \subseteq \mathbb{R}^{\ell \times k}, \qquad \mu^{Z,\text{list}}(z) = \mathbb{P}(z = (Z_1, \cdots, Z_\ell)).$$

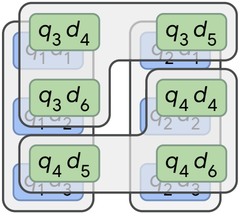
Preserves list structure on data.

List-Level vs. Item-Level Alignment

List-level alignment is stronger than item-level alignment.

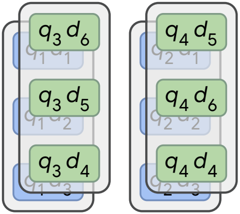
Example. Source and target both have 2 lists of length-3.

Representation 1



Aligned at item-level
but not at list-level

Representation 2



Aligned at item-level
and at list-level

Benefits of List-Level Alignment

1. We establish a domain adaptation generalization bound for ranking based on list-level alignment.

Theorem (Instantiated for mean reciprocal rank)

On ranking problems, under Lipschitz assumptions on model and scores, let $g : \mathcal{X} \to \mathcal{Z}$, then for all scoring models $h : \mathcal{Z} \to \mathbb{R}^{\ell}$,

$$\mathrm{MRR}_T(h \circ g) \geq \mathrm{MRR}_S(h \circ g) - \Theta(\ell)\, W_1(\mu_S^{Z,\mathrm{list}}, \mu_T^{Z,\mathrm{list}}) - \lambda_g^*,$$

where $\lambda_g^* = \min_{h'}(1 - \mathrm{MRR}_S(h' \circ g) + 1 - \mathrm{MRR}_T(h' \circ g))$. *Note that* $\mathrm{MRR} \in (0, 1]$.

Can also be instantiated for other ranking metrics, e.g., NDCG.

Benefits of List-Level Alignment

2. List-level alignment achieves better unsupervised domain adaptation performance vs. item-level alignment and zero-shot transfer.

Target domain	Method	MAP	MRR@10	NDCG@10
Robust04	BM25	0.2282	0.6801	0.4088
	Zero-shot	0.2759	0.7977	0.5340
	Item-level alignment	0.2822	0.8037	0.5396
	List-level alignment	**0.2901**	**0.8234**	**0.5573**
TREC-COVID	BM25	0.2485	0.8396	0.6559
	Zero-shot	0.3083	0.9217	0.8200
	Item-level alignment	0.3087	0.9080	0.8142
	List-level alignment	**0.3187**	**0.9335**	**0.8412**
BioASQ	BM25	0.4088	0.5612	0.4653
	Zero-shot	0.5008	0.6465	0.5542
	Item-level alignment	0.4781	0.6383	0.5343
	List-level alignment	**0.5191**	**0.6666**	**0.5714**

Source domain is MS MARCO.

References

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*. 2018, pp. 1025–1028.

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*. 2019, pp. 25–34.

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*. 2022, pp. 4008–4020.