Deep Domain Specialisation for single-model multi-domain learning to rank

Abdelmaseeh Felfel[1], Paul Missault[2]

Abstract

Information Retrieval (IR) practitioners often train separate ranking models for different domains (geographic regions, languages, stores, websites,...) as it is believed that exclusively training on in-domain data yields the best performance when sufficient data is available. Despite their performance gains, training multiple models comes at a higher cost to train, maintain and update compared to having only a single model responsible for all domains. Our work explores consolidated ranking models that serve multiple domains. Specifically, we propose a novel architecture of Deep Domain Specialisation (DDS) to consolidate multiple domains into a single model. We compare our proposal against Deep Domain Adaptation (DDA) and a set of baseline for multi-domain models. In our experiments, DDS performed the best overall while requiring fewer parameters per domain as other baselines. We show the efficacy of our method both with offline experimentation and on a large-scale online experiment on Amazon customer traffic.

Keywords

Information Retrieval, Neural IR, Learning To Rank, Domain Adaptation, Transfer Learning

1. Introduction

Production systems often use different ranking models for many different domains: ranking books products is not typically done with the same ranker as ranking electronics. Similarly, for a global store it is common for customers in the US to be served with a different ranker as customers in France. There are many of these *domains*, and when they differ sufficiently, researchers report that training exclusively on in-domain data outperforms models that are trained on data that includes out-of-domain data [1, 2]. Despite their performance gains, training multiple models comes at a higher cost to train, maintain and update compared to having only a single model responsible for all domains.

One solution to this increased maintenance cost is to create multi-domain models. Deep Domain Adaptation (DDA) [3] is a widely popular technique to create multi-domain neural models by forcing intermediary layers to learn domain agnostic representations. Models trained with Deep Domain Adaptation learn representations that are robust across all domains, but the in-domain performance drops since the robust representations can not leverage domain-specific features and patterns [4].

With the recent transition from tree-based to neural ranking models [5, 6] the idea of Deep Domain Adaptation came to mind as a method to create multi-domain rankers. However, in this

SIGIR'24: ReNeurIR'24: Workshop on Reaching Efficiency in Neural Information Retrieval, July 18, 2024, Washington, DC

afelfel@amazon.com (A. Felfel); pmissaul@amazon.com (P. Missault)

paper we do not aim to create robust domain agnostic representations, instead we propose to force domain *specific* representations. With this technique, we achieve the benefits of combining multiple models into a single model, while maintaining the performance of an in-domain model. We call our technique **Deep Domain Specialisation (DDS)**.

It is important to note that we are inspired by research in domain generalisation, but our goal is different. We aspire to create a multi-domain model that performs as well as or better than multiple single-domain models, but we make no claims about out-of-domain generalisation. We hypothesise that a single network can be made equivalent to multiple networks of single-domain models while being cheaper to train, store, and maintain.

In this paper we will train a DDS model and compare it with multiple baselines including a set of domain-specific baseline models. We will show our methodology can replace multiple in-domain models in terms of performance compared to the baselines, while having lower model complexity than other consolidated approaches.

2. Related work

The approach of *domain generalisation* was introduced to address multi-domain learning in cases where not all domains had enough domain-specific data available [7]. Early methods for domain generalisation relied on kernel methods [8], but for Deep Neural Networks the idea of Deep Domain Adaptation (DDA) is a popular baseline. It was first applied in computer vision where it showed significant out-of-domain performance increases [3]. This generalisation however comes at a cost to the source domain, since some of the domain-specific representations might be beneficial for the in-domain learning [4]. Domain generalisation is used in many fields[7], and has been formalised in search ranking across stores [1].

Next to the generalisation benefits of *domain generalisation*, we specifically call out its benefits in terms of *technical debt*. A compelling reason to reduce the amount of models is the CACE principle: "changing Anything Changes Everything" [9]. Responding quickly to changes (deprecation of a feature, legal compliance work, bringing improved performance) becomes more difficult as the amount of models in production increases. Alongside CACE, a single model is also more efficient to store (weights are reused). Finally, inference with a single model is easier to optimise since peak usage differs geographically and across domains.

3. Proposed models

3.1. Single-Domain baseline

The baseline in-domain model combines the latest techniques of Neural information retrieval. Specifically, we follow the RankerFormer architecture which incorporates a listwise transformer component for modeling query-product pairs jointly and enabling listwise aware ranking[6]. On top of this architecture, we add text encoders based on the BERT architecture[10] that were pretrained on the document corpus. The architecture thus consists of four components, which were trained end-to-end:

- **text similarity encoders:** BERT-based encoders are used to independently encode the search query and product titles. The encoder outputs are used to compute query-product similarity feature.
- **Trunk multilayer perceptron (MLP) component:** A feedforward neural network takes the text similarity features along with numerical and categorical query-product features as inputs. It generates pointwise relevance scores for each query-product pair.
- **Listwise transformer component:** A transformer-based listwise scoring architecture. Uses the previous component's pointwise scores for all products per query as input. Scores products by modeling the query's product and other product listed for the same query jointly leveraging attention mechanism.
- **Final MLP component:** Final feedforward neural network that takes both pointwise and listwise scores from the previous layers. It generates the final purchase probability score for each query-product pairs.

The architecture is depicted in Figure 1.

3.2. Multi-headed MLP

The multi-headed approach trains separate scoring head for each domain. In the base architecture of Section 3.1, we replace the head MLP with multiple MLP components: one for each domain. Each head outputs independent scores for each domain. By learning specialized domain-specific ranking heads, this approach aims to capture the nuances of each domain. No regularisation is placed on the lower layers, and as such the representation can capture domain-specific patterns. During inference, the right domain-specific scorer is selected by multiplication with a domain indicator feature ($store\_id$ in the Figure). The total score is the sum of the domain-specific predictions weighted with this domain indicator. Figure 2 shows the two headed architecture we use in the ranking experiment for 2 domains.

3.3. Deep Domain Adaptation

Deep Domain Adaptation (DDA) adds a domain classification head to the baseline model. Specifically, an additional MLP is appended to predict the domain of an input. Since the aim is to learn domain agnostic representations, we aim to *maximize* the classification loss of domain classification. Hence a Gradient Reversal layer (GRL) is placed between the domain classifier and the rest of the network. The Gradient Reversal Layer acts as an identity function during forward propagation but during back propagation it multiplies the gradient by -1. Intuitively, GRL forces lower layers to move in the direction that increases the loss, leading to representations where classification between domains becomes impossible. The goal is to learn representations in lower layers that are invariant to the different domains. The model architecture is shown in Figure 3.

3.4. DDS model

Deep Domain Specification (DDS) uses the same architecture from Section 3.3 but without the Gradient Reversal Layer. This causes the network to minimize it's classification loss instead of

maximising it. In turn, this forces the model to learn domain specific representations that are optimised simultaneous to do a specific task and to be maximally discriminative between input domains. The Model architecture is shown in Figure 4.

4. Experiments

4.1. Experiment definition

We will experiment in a Search relevance ranking task for 2 geographically different domains: a store in the United Arab Emirates (AE) and a store in Saudi Arabia (SA). Our goal is to create a single ranking models that can serve both the AE and SA stores, while maintaining the same or better performance as the independently trained models. This experiment is our first attempt at consolidating neural ranking models.

4.2. Datasets

	Train	Validation	Test
Dates	1 May - 31 May	1 June - 7 June	8 June - 15 June
SA #Queries	576,531	134,180	139,786
AE #Queries	1,856,515	378,498	418,932

Table 1
Breakdown of the experiment dataset by time period and store. This data is a uniform sample of production traffic to the stores in these geographic domains

The experiment used a dataset spanning May to June 2023. As seen in Table 1, the dataset was divided into train, validation, and test sets by time. We split by time instead of randomly to create a realistic dataset where query and user behavior patterns are not stationary. We use 1 month for training, 1 week for validation, and 1 final week held out for testing. Each query contained the customer feedback for the search query (clicks, purchases, video watches,...), and all the features, keywords and product descriptions of the top 130 ranked products shown to the customer.

4.3. Model training

We trained 5 models on this dataset: DDA, DDS, Two-headed MLP and 2 store specific baselines. All models were trained using the same listwise loss to optimize for the quality of the entire ranked list of products. DDA and DDS additionally had the domain (in this case the geographic location of the store) classification loss. The two-headed model used two independent but identical losses, one for each MLP component.

5. Results

5.1. Offline evaluation

We evaluated all 5 models on the held-out test set described in Table 1. For each store we additionally compared the 3 consolidated approaches against the store specific baseline as shown in Table 2. Both the two-headed model and the DDS model outperformed the baseline where the two-headed achieved a gain of 0.48% NDCG@16 and the DDS model had a 0.51% gain. On SA test data, the two-headed and DDS models achieved very similar NDCG gains of respectively 0.23% and 0.22% over the baseline.

As expected, the DDA model showed deteriorated performance compared to the domain-specific baselines on the AE and SA test sets. DDA was optimized for generalization across stores rather than Specialisation to each store. As a domain-adapted model, DDA is likely over-regularized in a way that hurts domain-specific relevance. Since our goal is to optimise performance in known domains -not to generalize to unseen domains- DDA is not a good fit for further online testing with customers despite its consolidated architecture.

These offline results demonstrate the potential for consolidated models to not only save maintenance cost but also improve relevance over independent domain-specific models. By training on a larger dataset combined across stores, the consolidated models appear to leverage that data to perform better on each store.

store	Model	NDCG@16 (gain over store model)
AE	AE trained	0.535
	DDA	0.492 (-7.98%)
	two-headed	0.537 (0.48%)
	DDS	**0.538 (0.51%)**
SA	SA trained	0.524
	DDA	0.475 (-9.36%)
	two-headed	**0.526 (0.23%)**
	DDS	0.525 (0.22%)

Table 2
Offline analysis on SA and AE dataset. Full scorecard is linked here.

The two-headed and DDS models both increase performance compared to the baseline. Although, we can not conclude that a one is better than the other, see more details in the appendix B. However, DDS achieves this performance with the least model parameters: DDS uses 1 scoring head (+ 1 classification head that can be cut off after training) against the N scoring heads for an N-headed solution. As per Appendix **??** in our models this relates to 40K extra parameters per head.

6. Online experimentation

After seeing positive gains in offline evaluation, we launched an interleaving experiment[11] to validate the performance of the consolidated models online. We analyzed a range of online metrics, but we report the increases in sales as interleaving credit, which has high correlation to

corresponding metric from an A/B test [11]. The gain is reported against the production model, which are store-specific models. As such that results of the store-specific models can be seen as "refreshes" of the production models. The experiment was run for 2 weeks with the online results shown in Table 3

store	Model	Interleaving credit
AE	AE trained	9.15%; p=0.0000
	two-headed	12.92%; p=0.0000
	DDS	14.79%; p=0.0000
SA	SA trained	12.91%; p=0.0034
	two-headed	28.35%; p=0.0000
	DDS	24.67%; p=0.0000

Table 3
Online Interleaving experiment results on SA and AE stores.

The online results further validate our offline findings that a single consolidated models can replace multiple in-domain models. Additionally, we see that consolidation improved performance in this experiment. Specifically, both the DDS and two-headed consolidated models outperform the production model and the store specific baselines in both AE and SA.

7. Conclusion and Future Work

Both the Deep Domain Specialisation (DDS) and two-headed models managed to performed as good as or better than the domain-specific models in both offline and online experiments. This empirically demonstrates our claim that consolidated models can be used to reduce technical debt, and do not necessarily incur performance losses as previously thought. Comparing between DDS and the two-headed approach, DDS had better performance online and can be scaled easier to expand to additional stores. The multi-headed approach needs N heads for N domains. DDS on the other hand does not require any architectural change and will scale to N domains with the same parameters. Scaling to multiple domains, more diverse domains is left for future work. We additionally want to investigate the performance gains we saw, as these were unexpected.

References

[1] P. Missault, A. de Myttenaere, A. Radler, P.-A. Sondag, Addressing cold start with dataset transfer in e-commerce learning to rank (2021).

[2] D. Chen, Y. Xiong, J. Yan, G.-R. Xue, G. Wang, Z. Chen, Knowledge transfer for cross domain learning to rank, Information Retrieval 13 (2010) 236–253.

[3] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189.

[4] P. Chattopadhyay, Y. Balaji, J. Hoffman, Learning to balance specificity and invariance for in and out of domain generalization, in: Computer Vision–ECCV 2020: 16th European

Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16, Springer, 2020, pp. 301–318.

[5] Z. Qin, L. Yan, H. Zhuang, Y. Tay, R. K. Pasumarthi, X. Wang, M. Bendersky, M. Najork, Are neural rankers still outperformed by gradient boosted decision trees?, in: International Conference on Learning Representations, 2020.

[6] M. Buyl, P. Missault, P.-A. Sondag, Rankformer: Listwise learning-to-rank using listwide labels, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 3762–3773.

[7] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, C. C. Loy, Domain generalization: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2022) 4396–4415.

[8] K. Muandet, D. Balduzzi, B. Schölkopf, Domain generalization via invariant feature representation, 2013. arXiv:1301.2115.

[9] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, Advances in neural information processing systems 28 (2015).

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[11] N. Bi, P. Castells, D. Gilbert, S. Galperin, P. Tardif, S. Ahuja, Debiased balanced interleaving at amazon search, in: CIKM 2022, 2022. URL: https://www.amazon.science/publications/debiased-balanced-interleaving-at-amazon-search.

Appendices

A. Model Figures

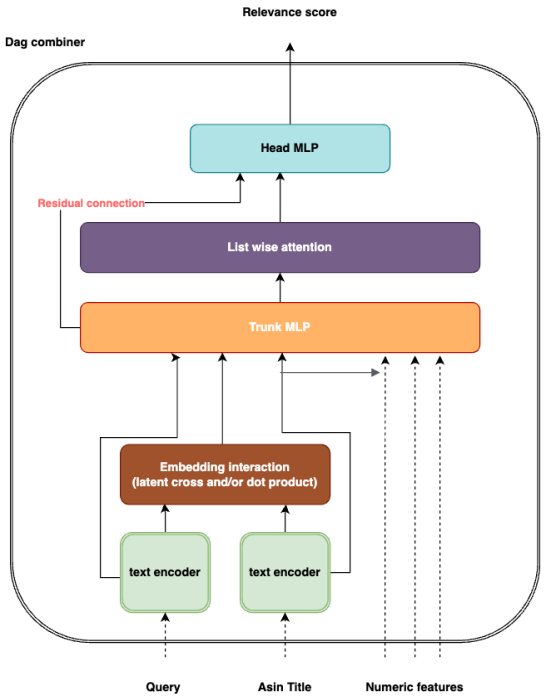


Figure 1
Single model architecture (baseline)

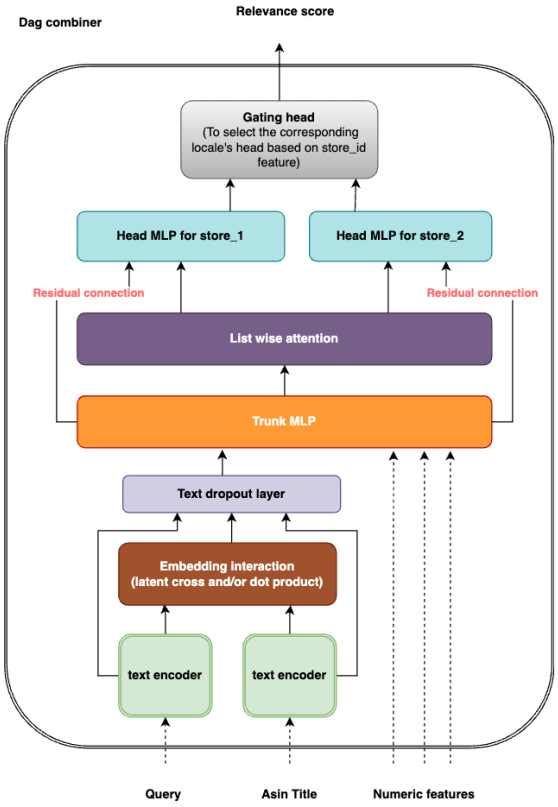


Figure 2
Two headed architecture

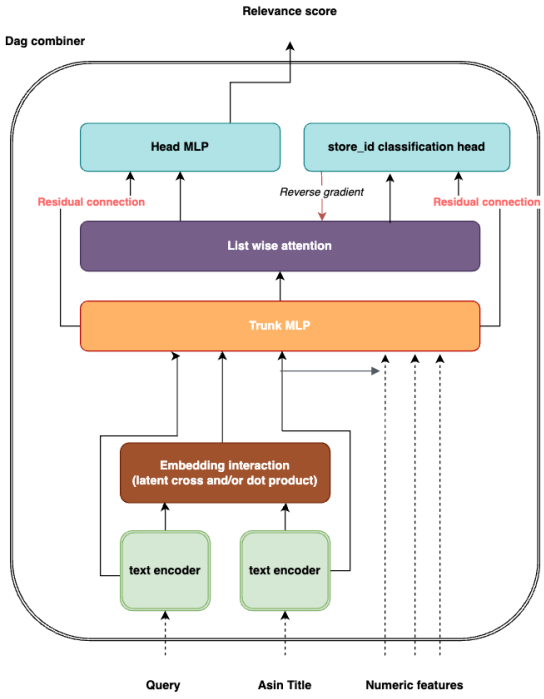


Figure 3
DDA architecture

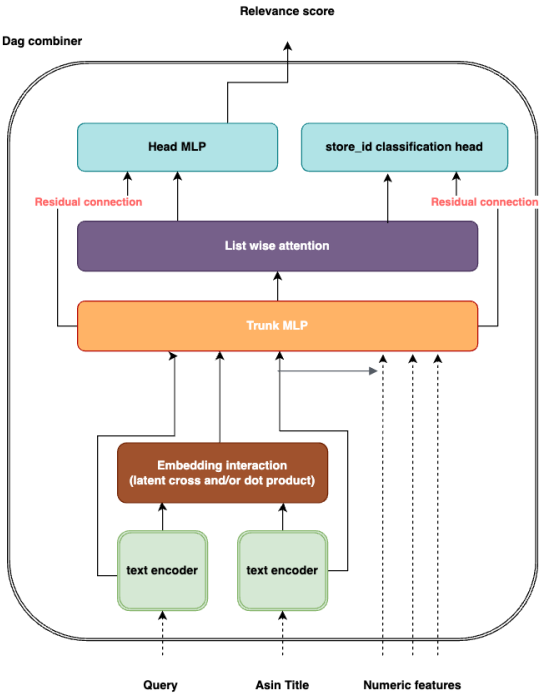
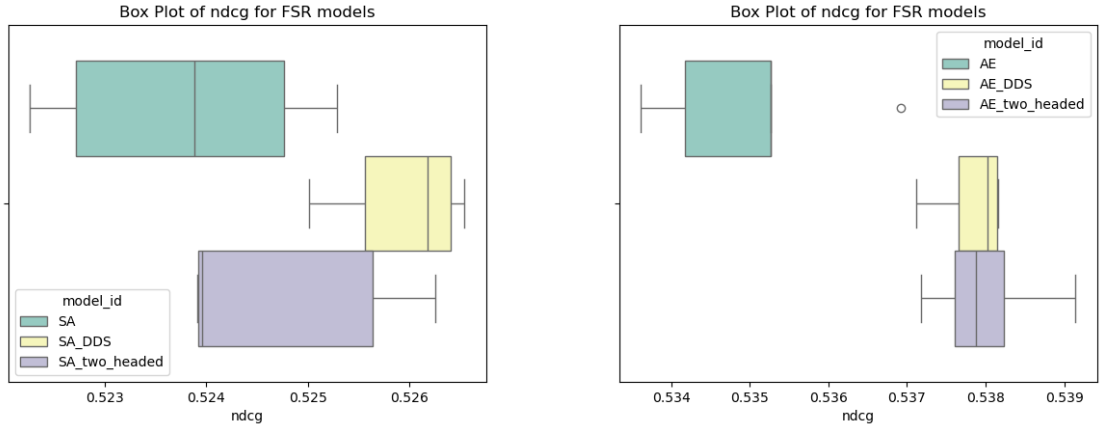


Figure 4
DDS architecture

B. Offline evaluation

To evaluate the models, we conducted 5 training runs for each model (the baselines, two-headed model, and DDS) with different random seeds. For each training run, we selected the model with the highest NDCG scores on a validation dataset. We then evaluated all chosen models from the 5 runs on the SA and AE test sets. Figure 5 below shows boxplots of the 1st and 3rd quartiles alongside the median NDCG scores across the 5 runs for each model on the two test sets.

The results show that both consolidated models (DDS and multi-headed) have better or equal performance than the single domain models on both the SA and AE test sets. When comparing DDS and the two-headed models, the quartiles overlap and so we cannot conclusively determine if one model performs better than the other. As highlighted in our work, we propose DDS based on its better scaling in terms of model parameters.



(a) Models evaluation against SA test set

(b) Models evaluation against AE test set

Figure 5: NDCG scores of 5 runs of the different methods and baselines used in this paper.