Diversification-Aware Learning to Rank using Distributed Representation

Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky
{lyyanle,zhenqin,ramakumar,xuanhui,bemike}@google.com
Google Research
Mountain View, California

ABSTRACT

Existing work on search result diversification typically falls into the "next document" paradigm, that is, selecting the next document based on the ones already chosen. A sequential process of selecting documents one-by-one is naturally modeled in learning-based approaches. However, such a process makes the learning difficult because there are an exponential number of ranking lists to consider. Sampling is usually used to reduce the computational complexity but this makes the learning less effective. In this paper, we propose a soft version of the "next document" paradigm in which we associate each document with an approximate rank, and thus the subtopics covered prior to a document can also be estimated. We show that we can derive differentiable diversification-aware losses, which are smooth approximation of diversity metrics like α-NDCG, based on these estimates. We further propose to optimize the losses in the learning-to-rank setting using neural distributed representations of queries and documents. Experiments are conducted on the public benchmark TREC datasets. By comparing with an extensive list of baseline methods, we show that our Diversification-Aware LEarning-TO-Rank (DALETOR) approaches outperform them by a large margin, while being much simpler during learning and inference.

CCS CONCEPTS

• **Information systems** → **Information retrieval**;

KEYWORDS

search result diversification; diversification-aware loss; learning-to-rank

ACM Reference Format:
Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky. 2021. Diversification-Aware Learning to Rank using Distributed Representation. In *WWW '21: The Web Conference, April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3449831

1 INTRODUCTION

Search result diversification is critical for the utility of search engines due to the diverse information needs of users and the ambiguity of short queries. For example, a query about "Eiffel Tower" may seek for its history information or its visiting address. A list of results covering different subtopics is more desired in this case. Indeed, diversification has been a long-standing research topic in the Information Retrieval community, with the seminal work of Maximal Marginal Relevance (MMR) dating back to year 1998 [10].

Since users often do not examine all the returned results thoroughly but only look at a few top ones, the goal of search result diversification is to present relevant but diverse results at the top of a ranked list. These notions are taken into account by commonly used diversity evaluation metrics, including α-NDCG [13], ERR-IA [11] (Intent-Aware metrics [1]), and S-recall [45]. All of them consider both the ranks of relevant documents and how well the subtopics of a given query are covered by the top ranked documents. The contribution of a subtopic in a lower-ranked document is down-weighted if it has been covered well by top-ranked ones.

While traditional methods for diversification are mainly manually crafted [10, 14, 34], recent research in this area shifts to supervised learning methods [18, 26, 35, 40, 42, 44, 46] and shows superior performance with respect to the diversity evaluation metrics. However, different from the standard learning-to-rank setting [27], the design of learning approaches for diversification is non-trivial due to the inter-dependency among documents. Almost all of them fall into the so-called "next document" paradigm, that is, selecting the next document among the remaining ones to maximize an objective with respect to the ones already chosen. Such a paradigm is intuitively appealing and fits the diversification task naturally. However, the main challenge is that learning is inherently less effective because there is an exponentially large number of ranking lists to consider. Different methods are proposed to alleviate this problem. For example, R-LTR [46] and SVM-DIV [44] mainly focus on the ideal diversified ranking lists. Reinforcement learning (RL) based approaches [18, 42] try to maximize the *expected* rewards over sampled lists from a distribution. Recently proposed PAMM [40] and DVGAN [26] maximize the margin between sampled positive and negative lists for training and show better performance. However, the huge number of candidates poses challenges for high-quality sampling [26].

The main difficulty of the existing learning-based approaches lies in the *hard* setting in the "next document" paradigm – the next document is evaluated based on the materialized previously selected documents. The key idea of this paper is to use a *soft* version where we do not need to materialize ranking lists. Specifically, we compute a differentiable *approximate rank* and associate it with each document. Given such approximate ranks, we can estimate the subtopic coverage of the documents prior to each document, and then use these estimates for diversification. In particular, we show that we can translate a diversity evaluation metric (e.g., α-NDCG)

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449831

Table 1: An example illustrating how distributed representation helps optimize diversification-aware ranking losses. d_1 d_2 and d_3 are relevant to subtopic #1 in f_1. d_4 is relevant to subtopic #2 in f_2. The third dimension f_3 is a diversity-useful dimension and the last, f_4, is an independent dimension.

subtopics	#1	#2	-	
distributed dims	f_1	f_2	f_3	f_4
q	1	1	0	0
d_1	1	0	1	1
d_2	1	0	0	1
d_3	1	0	0	1
d_4	0	1	0	1

into a differentiable loss function based on these estimates. Such a loss function can be trained effectively by gradient descent in an end-to-end fashion.

With such a diversification-aware loss function, we formulate our problem in a learning-to-rank setting where a ranking function is learned to *score and sort* documents. To make the learning more effective, instead of using aggregated score features like BM25 or TF-IDF (commonly used in the traditional learning-to-rank setting [27]), we resort to the distributed representations of queries and documents, where subtopic-relevant dimensions and diversity-useful dimensions can be learned.

A simple example in Table 1 illustrates this. Suppose there are 2 subtopics encoded in a 4-dimension latent space. Without loss of generality, we define the coordinate with the first two axes f_1 and f_2 aligning with the 2 subtopic vectors. It is then easy to pick out the query and subtopic-relevant documents by looking at the alignments in these dimensions. Suppose we find 4 relevant documents: d_1, d_2, and d_3 match subtopic #1; d_4 matches subtopic #2. An ideal diversified ranking must have d_4 ranked at either the first or the second position, which can be achieved by assigning different scores to d_1, d_2, and d_3 in the "score-and-sort" setting. However, it is not feasible to distinguish d_1, d_2, and d_3 by just looking at the alignments with the subtopics. In such case, the remaining dimensions of distributed representations become useful in generating the diversified ranking. In this example, suppose the four documents distributed non-trivially along axis f_3, a diversification-aware loss function can facilitate learning such distributed representations and output a list $[d_1, d_4, d_2, d_3]$ by favoring a ranking function such as $f_1 + 1.5f_2 + f_3$, while a standard loss function may not be able to pick up f_3, as it has no effect on document relevance.

Note that once the ranking function has been learned, our method does not require the subtopics of a query to be given during *inference* and thus belongs to the *implicit* category. This is different from the *explicit* approaches (e.g., DVGAN [26] or DSSA [23]), which assume the subtopics to be available at inference time – not a realistic assumption for most search engines.

One obvious problem of the score-and-sort approach is the duplicate documents where two documents have the exact same feature representation and thus would always have the same scores. The benchmark dataset used in our experiments does have many documents covering the same subtopics but they are not duplicates. Without any pre-processing, our methods perform very well on this dataset, confirming that the diversification-aware loss can effectively leverage the difference among documents that are relevant to the same subtopics and yield a diversification-aware ranking function. In reality, this problem can be easily fixed by a pre-processing step where we make sure that all documents for a query have a minimum difference (e.g., using a near-deduplication technique [6]).

In summary, we make the following contributions in this paper:

- We propose a novel method that can translate a diversity evaluation metric to a differentiable diversification-aware loss.
- We show that such a loss function can be effective in learning with the distributed representation using deep neural networks that are efficient and easily extendable.
- We conduct experiments on a public benchmark dataset and show that our proposed method can significantly outperform strong recent baselines.

The rest of the paper is organized as follows. We review related work in Section 2. Our diversification-aware loss is described in Section 3 and a neural network based learning approach is described in Section 4. In Section 5, we give a theoretical analysis about our diversification-aware loss and distributed representation. We present our experiments in Section 6, our discussion in Section 7 and conclude this paper in Section 8.

2 RELATED WORK

2.1 Search Result Diversification

Diversification approaches can be broadly classified into two approaches: *implicit* and *explicit*. Implicit approaches promote novel documents compared to other documents based on inter-document similarity and do not require subtopics given during the evaluation time. Explicit approaches promote documents that improve coverage over specified subtopics.

Most implicit approaches are inspired by the seminal work of Maximal Marginal Relevance (MMR) method [10], which iteratively picks relevant documents that are novel to the document set so far, based on user defined functions for inter-document similarity. Supervised machine learning methods for implicit diversification [40, 41, 44, 46] learn a scoring function that optimize diversification using remote proxies of evaluation metrics. SVM-DIV [44] utilizes structural SVMs for scoring, whereas R-LTR [46] proposes a relational learning to rank framework, to model document relations of an ideally diversified ranking. Based on R-LTR, PAMM [40] improves the scoring function by considering difference between positive and negative rankings. NTN-DIV [41] uses a neural tensor network to learn document similarity automatically instead of using handcrafted features or functions. Reinforcement learning methods have been proposed to improve the greedy nature of sequential document selection: MDP-DIV [42] uses Markov Decision Processed (MDP) to optimize expected reward over sampled lists; M^{2}DIV improves over MDP using Recurrent Neural Networks (RNNs) to model the document sequence and Monte Carlo Tree Search (MCTS). An alternate approach tries to directly optimize user's utility instead of any proxies of diversity. Determinantal Point Processes [39] and Multi-Armed Bandits [36] are used to generate diverse rankings to optimize click-through rate.

Explicit diversification approaches, e.g., PM-2 [14] or xQuAD [34] improve the coverage over subtopics relevant to a query based on the sub-queries given during evaluation. Most recent methods in this category employ various neural architectures. Diverse Search with Subtopic Attention (DSSA) [23] uses RNNs with attention mechanism for a sequential procedure to greedily select relevant documents which are diverse to current selected set. As a follow-up to DSSA, DVGAN [26] uses Generative Adversarial Networks to frame the diversification problem as a minimax game between a generator and a discriminator, where the generator models document similarity and the discriminator uses subtopic information, and DESA [32] adds self-attention encoder and decoder to replace RNNs to model interactions of all documents in the list.

While our proposed method shares some commonalities with the above methods in terms of neural architecture (e.g., similarly to DESA [32] we use self-attention), there are two crucial differences. First, we estimate latent subtopics from the distributed representation of the query, rather than requiring providing them explicitly. Second, our end-to-end learning-to-rank framework directly optimizes a smooth approximation of the diversity metrics.

2.2 Learning To Rank

Learning to rank has traditionally focused [27, 33] on relevance. A scoring function per query-document pair (referred to as *univariate* scoring [3]) is learned to minimize the loss using learning algorithms such as Gradient Boosted Decision Trees (GBDT) [24] and neural networks [20, 29]. For learning the univariate scoring function, Deep Structured Semantic Matching models (DSSM) [22] learn a low dimensional embedding for queries and documents and use a dot-product as the score. In the domain of neural network scoring functions, *multivariate* scoring functions which capture cross document interactions as listwise context have been proposed in such as Deep Listwise Context Model [2], Groupwise Scoring Functions [3], Document Interaction Network [30], and SetRank [28]. We also leverage deep listwise context via self-attention, but our goal is diversify results, but not just on improving relevance as in existing learning to rank work.

2.3 Approximation of Ranking Metrics

Neural networks are amenable for end-to-end learning to directly optimize ranking metrics [7, 8] by creating a differentiable approximation. ApproxNDCG [8] revisits the idea proposed in [31] of replacing indicator functions in rank computation with sigmoid functions for a differentiable surrogate. Sampling scores from Gumbel distribution to compute the expectation of the approximated metric (over induced permutations) is shown to gain additional robustness [7, 19]. Based on these, we introduce differentiable approximations of diversity evaluation metrics for the first time.

Another common technique to make differentiable approximations of ranking metrics is LambdaRank [9, 16, 38], which uses the metric delta when two documents are swapped in the loss. Compared to the approximation techniques above, it is nontrivial to apply LambdaRank to non-additive diversification metrics. A recent work by Yigit-Sert et al. [43] treats diversification as a fusion task of rankings obtained by sorting the documents with respect to individual subtopics. The LambdaRank is used as the standard LTR to obtain the ranking for each subtopic, but not to optimize diversification metrics directly.

3 DIVERSIFICATION-AWARE LOSSES

3.1 Diversity Evaluation Metrics

To introduce our differentiable losses that are close approximations of the actual diversity evaluation metrics, we first review two commonly used diversity evaluation metrics: α-NDCG and ERR-IA.

3.1.1 α-NDCG. Consider n documents associated with a query and each document may cover 0 to m subtopics, which is indicated by subtopic labels y_{il}: $y_{il} = 1$ if document i covers subtopic l and $y_{il} = 0$ otherwise. The α discounted cumulative gain (α-DCG) [13] is then defined as,

$$\alpha\text{-DCG} = \sum_{i=1}^{n} \sum_{l=1}^{m} \frac{y_{il}(1-\alpha)^{c_{li}}}{\log_2(1+r_i)}, \tag{1}$$

where α is a parameter between 0 and 1 quantifying the probability a reader got the information about a given subtopic from a relevant document, r_i is the rank of the document i, and c_{li} is the number of times the subtopic l being covered by documents prior to rank r_i:

$$c_{li} = \sum_{j:r_j \le r_i} y_{jl} \,. \tag{2}$$

We can normalize this measure into the range $[0, 1]$ by dividing the optimal α-DCG given the document list:

$$\alpha\text{-NDCG} = \frac{\alpha\text{-DCG}}{\alpha\text{-DCG}_{\text{opt}}}. \tag{3}$$

α-NDCG@k are metrics commonly used, which are obtained by summing over only the top k ranked documents in the list.

3.1.2 ERR-IA. For the same setting as above, the ERR-IA [12] is defined as

$$\text{ERR-IA} \equiv \sum_{i=1}^{n} \frac{1}{r_i} \sum_{l=1}^{m} \frac{1}{m} \left(\prod_{j:r_j<r_i} (1 - \frac{2^{y_{jl}}-1}{2^{y_l^{\max}}}) \right) \frac{2^{y_{il}}-1}{2^{y_l^{\max}}}. \tag{4}$$

For binary labels $y_{il} = 0$ or 1 and $y_l^{\max} = 1$, this definition can be easily rewritten in terms of rank r_i and subtopic coverage c_{li},

$$\text{ERR-IA} = \sum_{i=1}^{n} \frac{1}{r_i} \sum_{l=1}^{m} \frac{1}{m} \frac{y_{il}}{2^{c_{li}+1}} \tag{5}$$

In practice, ERR-IA is further normalized by a constant $\sum_{r=1}^{n} \frac{1}{r} 2^{-r}$. Similar to α-NDCG@k, ERR-IA@k are commonly used metrics with a summation over the top k documents for both ERR-IA and the normalization factor.

3.2 Differentiable Approximate Losses

In both α-DCG and ERR-IA metrics, r_i and c_{li} are ranking-dependent. These make them non-differentiable. The first contribution of this paper is a smooth approximation of the diversity metrics through *soft* versions of the rank r_i and subtopic coverage c_{li}. This is achieved by re-expressing them using the scores assigned to each document. In the following, we use α-DCG as an example. All the derivations can be applied to the ERR-IA metric without any difficulty.

Let s_i be the score for document i, r_i and c_{li} can then be formulated as:

$$r_i = 1 + \sum_j \mathbb{I}_{s_j > s_i},$$
$$c_{li} = \sum_j y_{jl} \mathbb{I}_{s_j > s_i}, \tag{6}$$

where $\mathbb{I}$ is the indicator function. The indicator function is not differentiable, but can be approximated by the sigmoid function

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x/T)}$$

where T is a positive smoothness parameter. Applying this approximation, to the explicit formulations of rank and subtopic coverage in Eq.(6), we get the differentiable smooth approximations to rank r_i and subtopic coverage c_{li} with a single parameter T.

$$R_i = 1 + \sum_{j \neq i} \text{sigmoid}\left(\frac{s_j - s_i}{T}\right) = \frac{1}{2} + \sum_j \frac{1}{1 + \exp\left(\frac{s_i - s_j}{T}\right)},$$
$$C_{li} = \sum_{j \neq i} y_{jl} \cdot \text{sigmoid}\left(\frac{s_j - s_i}{T}\right) = \sum_j \frac{y_{jl}}{1 + \exp\left(\frac{s_i - s_j}{T}\right)} - \frac{y_{il}}{2} \tag{7}$$

The approximations are strictly equal but not differentiable as $T \to 0$, and the larger is the parameter T, the more smooth are the differentiable approximations. These soft versions of ranks and subtopic coverage become the estimates to capture the *soft* version of *next document*.

By inserting these approximations to the α-DCG metric definition in Eq.(1), we then obtain a directly differentiable diversification-aware α-DCG loss,

$$\mathcal{L}_{\alpha\text{-DCG}}(\{s_i^q\}) = -\frac{1}{|Q|} \sum_{q \in Q} \sum_{i=1}^{n} \sum_{l=1}^{m} \frac{y_{il}^q (1-\alpha)^{C_{li}^q}}{\log_2(1 + R_i^q)}, \tag{8}$$

where we add back superscript q to define the loss over a set of queries Q. $\mathcal{L}_{\alpha\text{-NDCG}}$ can be similarly derived with a constant normalization weight for each query.

Finally, another useful variation to this diversity ranking loss is to add a stochastic treatment [7].

$$\mathcal{L}_{\text{Gumbel-}\alpha\text{-DCG}}(\{s_i^q\}) = \mathbb{E}_g[\mathcal{L}_{\alpha\text{-DCG}}(\{\beta(s_i^q + g_i)\})], \tag{9}$$

where parameter β is a noise-level parameter and Gumbel noise g_i is sampled from Gumbel distribution $g_i = -\log(-\log(U_i))$ with U_i uniformly distributed in $[0, 1]$.

4 NEURAL LEARNING

Now that we formulated the diversification-aware loss, we next describe how to leverage the neural networks to optimize this loss in the learning-to-rank setting.

4.1 Distributed Representation

Instead of the heuristic aggregated ranking features, our approach relies on distributed representations of the text-based query q and candidate document list $\{d_i\}$, which can be generated by a trainable neural encoder. Popular choices are BERT [15] or doc2vec [25]. Essentially, they encode the text-based queries and documents with various lengths into dense normalized vectors $\mathbf{e}_q \in \mathbb{R}^E$ and $\mathbf{e}_i \in \mathbb{R}^E$ of a fixed embedding dimension E, so that one can determine the similarity of different documents (and queries) in this latent space.

The latent space allows us to apply recent neural interaction methods to better capture the relationship between query-document pairs. In this work, we use a simple algorithm, latent cross [5], which can effectively generate high-order interaction features: for each pair of query and document representations $\mathbf{e}_q$ and $\mathbf{e}_i$, we define a query-document cross feature $\mathbf{c}_i \in \mathbb{R}^E$ by an element-wise multiplication,

$$\mathbf{c}_i = \mathbf{e}_i \circ \mathbf{e}_q. \tag{10}$$

The representation of each query can be thought as a mix of the subtopics and the element-wise product between a query and a document can then be easily de-mixed to compute the matching between subtopics and the document.

Note that the latent representations for query and document can be extracted from pre-trained models, or jointly tuned with the ranker end-to-end in our neural framework.

4.2 Listwise Context Embedding

Intuitively, information from the entire document list is helpful for the diversification task. Thus, we enrich the distributed representation of a document by considering representations of the entire list based on pairwise document similarity. Specifically, we incorporate into our framework the Document Interaction Network (DIN) proposed in [30], a similar idea also implemented in [32]. Note that while we use this to enhance our scoring function, the objective of this work is to improve diversification of ranking, whereas [30] focuses on improving ranking measures.

DIN generates an embedding of the candidate list, $\mathbf{a}_i$, for each document i, using the multi-head self-attention (MHSA) mechanism, introduced in Transformers [37]. DIN uses pairwise dot-product attention to capture document similarity between document i and every document in the list. For the multi-head self-attention mechanism, we concatenate the features for all documents in the list to input $D \in \mathbb{R}^{n \times k}$, where k is the feature dimension corresponding to one document. We project D into a query[1] matrix $Q = DW^Q$, a key matrix $K = DW^K$, and a value matrix $V = DW^V$ with trainable projection matrices W^Q, W^K, and $W^V \in \mathbb{R}^{k \times z}$, where z is the attention head size. Then a self-attention (SA) head computes the weighted sum of the transformed values V as,

$$\text{SA}(D) = \text{Softmax}(S(D))V, \tag{11}$$

where similarity matrix between Q and K is defined as $S(D) = \frac{QK^T}{\sqrt{z}}$. For each layer, the results from the H heads are concatenated to form the output of multi-head self-attention by

$$\text{MHSA}(D) = \text{concat}_{h \in [H]}[\text{SA}_h(D)]W_{\text{out}} + b_{\text{out}}, \tag{12}$$

where $W_{\text{out}} \in \mathbb{R}^{Hz \times z}$ and $b_{\text{out}} \in \mathbb{R}^{n \times z}$ are trainable parameters. To compute the listwise context embedding, we apply $L \geq 1$ layers of multi-head self-attention over the input documents D. Similar to Transformer [37], we also apply residual connections [21] and layer normalization [4] to each layer. We augment the features of the query-document scoring function with the listwise context embedding from the final output of the L-th self-attention layer. Since this embedding contains information from the whole candidate

[1]Please note that this query is different from the search query.

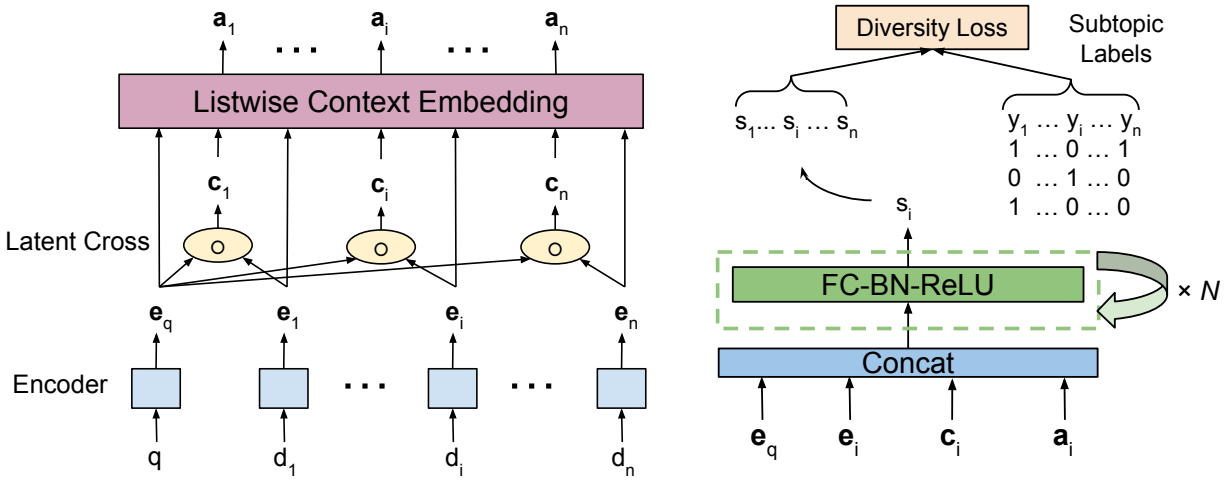

Figure 1: An illustration of the diversification-aware learning to rank (DALETOR) architecture. The inputs $\{\mathbf{e}_q, \mathbf{e}_i, \mathbf{c}_i, \mathbf{a}_i\}$ on the right are corresponding outputs from the left. FC stands for fully-connected layer, BN stands for batch normalization and ReLU stands for the nonlinear activation.

list, it serves as complementary features to distinguish documents covering the same subtopics.

4.3 Architecture

Figure 1 summarizes our end-to-end framework of Diversification-Aware LEarning TO Rank (DALETOR). From a text-based query and candidate document list, we first obtain their distributed representations, $\mathbf{e}_i$, with a trainable document encoder. In the case when the latent cross (in the blue box) is applied, we generate the query-document cross representations $\mathbf{c}_i$ by an element-wise multiplication of query and document representations. If DIN (in the purple box) is incorporated, we pass all representations from query $\mathbf{e}_q$, documents $\mathbf{e}_i$, and query-document cross $\mathbf{c}_i$ (when applicable) to a self-attention layer to get the listwise context representation $\mathbf{a}_i$ for each document. Finally, all generated representations, query $\mathbf{e}_q$, document $\mathbf{e}_i$, latent cross $\mathbf{c}_i$, and listwise context embedding $\mathbf{a}_i$, are concatenated and passed through several full connected layers to compute the final ranking score for each document associated with the query. The output scores are then fed into a diversification-aware loss during training and used for sorting during inference. For a univariate neural scorer $s(.)$, the scoring function for the full DALETOR architecture is as follows:

$$\begin{aligned} \mathbf{a}_i &= \mathrm{DIN}_i(\{\mathrm{concat}(\mathbf{e}_q, \mathbf{e}_j, \mathbf{c}_j)\}) \\ s_{\mathrm{DALETOR}}(q, \{d_i\}) &= \{s(\mathrm{concat}(\mathbf{e}_q, \mathbf{e}_i, \mathbf{c}_i, \mathbf{a}_i))\}. \end{aligned} \tag{13}$$

5 A THEORETICAL ANALYSIS

In this section, we elaborate in-depth how diversification loss and distributed representation work to output a diversified ranking list.

5.1 Relevance and Diversity in the Differential α-DCG Loss

As the α-DCG loss captures both the relevance through rank discount $1/\log_2(1+R)$ and the diversity through coverage discount $(1-\alpha)^C$, optimizing α-DCG loss trains the model to capture both sides of diversified ranking.

To show that, consider two documents 0 and 1 with close but different initial scores $s_0 \approx s_1$, while scores of all other documents are fairly far from these two. We want to ask how this difference $\delta s \equiv s_0 - s_1$ changes over the training course, especially, the sign of $\frac{d\delta s(t)}{dt}$ and the sign of the coefficient if $\frac{d\delta s(t)}{dt}$ is proportional to δs.

Scores tend to go downward along the gradient,

$$s_i(t+1) - s_i(t) \propto -\frac{\partial \mathcal{L}}{\partial s_i}.$$

As a result, the relative score changes as

$$\frac{d\delta s(t)}{dt} \propto \delta s(t+1) - \delta s(t) \propto \frac{\partial \mathcal{L}}{\partial s_1} - \frac{\partial \mathcal{L}}{\partial s_0}.$$

For small score difference $|\delta s| \ll T$, rewriting s_0 and s_1 in δs and $\bar{s} \equiv \frac{s_0+s_1}{2}$ as $s_0 = \bar{s} + \frac{\delta s}{2}$ and $s_1 = \bar{s} - \frac{\delta s}{2}$, we can expand around $\bar{s}$,

$$\begin{aligned} \frac{d\delta s(t)}{dt} \propto & \left(\frac{\partial \mathcal{L}}{\partial s_1} - \frac{\partial \mathcal{L}}{\partial s_0}\right)\Big|_{s_0, s_1 = \bar{s}} \\ & - \frac{\delta s}{2}\left(\frac{\partial^2 \mathcal{L}}{\partial s_0^2} + \frac{\partial^2 \mathcal{L}}{\partial s_1^2} - 2\frac{\partial^2 \mathcal{L}}{\partial s_0 \partial s_1}\right)\Big|_{s_0, s_1 = \bar{s}} + o(\delta s^2). \end{aligned} \tag{14}$$

So the gradients $\partial \mathcal{L}/\partial s$ and the hessians $\partial^2 \mathcal{L}/\partial s^2$ are important for this analysis. We compute the explicit expressions of gradients and hessians for α-DCG loss and softmax loss (not shown in the main text) and keep the dominant order in analysis below.

To show the α-DCG loss promotes the relevant documents, we consider two documents: document 0 is not relevant to any of the subtopics $y_{0l} = 0$, while document 1 covers some subtopic, $\exists l$ s.t. $y_{1l} = 1$. Without loss of generality, assume document 1 covers subtopic $l = 1$. As scores of document 0 and 1 are close to each other but far from the others $|s_0 - s_1| \ll T$ and $|s_{0,1} - s_j| \gg T$ for $\forall j \neq 0, 1$, we have $R_0 \approx R_1$, $C_{l0} \approx C_{l1}$, and the derivative of

the sigmoid function $\text{sigmoid}'\left(\frac{s_j - s_{0,1}}{T}\right) \approx 0$ for $\forall j \neq 0, 1$. So the relative score change between 0 and 1 will be dominated by the gradient $\frac{\partial \mathcal{L}}{\partial s_1}$ term in Eq.(14) as

$$\frac{d\delta s}{dt} \propto -\frac{1}{T}\frac{y_{11}}{1+\bar{R}}\frac{(1-\alpha)^{\bar{C}}\ln 2}{(\ln(1+\bar{R}))^2} < 0.$$

The ranker learns to rank the relevant document 1 higher relative to document 0.

To demonstrate the loss promotes the diversity through learning features differentiating documents with the same labels, we now suppose the two documents cover the same subtopic of the query, say topic 1, $y_{01} = y_{11} = 1$ and 0 for $l \neq 1$. Then it is easy to show the gradient terms in Eq.(14) vanish and the rate of relative score becomes dominated by the hessian terms. Ignoring the higher order contributions of $1/\bar{R}$ from the derivatives of rank, we have

$$\frac{d\delta s(t)}{dt} \propto \left\{ \frac{\left(\ln\frac{1}{1-\alpha}\right)}{T^2}\frac{(1-\alpha)^{\bar{C}_t}}{\log_2(1+\bar{R})} + o\left(\frac{1}{1+\bar{R}}\right)\right\}\delta s,$$

where the coefficient of the δs term on the right hand side is positive. So indicated by the α-DCG loss, the score difference of the two documents with the same label tends to diverge over the training. In contrast, in other typical relevance ranking losses, softmax loss for example, one can show that the same coefficient in front of δs is negative so that the scorer trained with the softmax loss always tends to score the documents with the same label similarly.

5.2 Distributed Representation Facilitates Diversity

Distributed representation boosts diversity learning through the following three aspects: (i) Neural networks can learn the representations of subtopics in the latent space; (ii) The alignment between a subtopic and a candidate document can be learned from a transformation of the latent cross between the corresponding query and document; (iii) A score function that discriminates the documents aligning with the same subtopics can be learned from the query and subtopic independent dimensions with nontrivial distributions of candidate documents.

Consider distributed representations in a latent space, the number of dimensions E is much greater than the number of subtopics m. All subtopics and query can be represented as vectors in the space: say $\mathbf{e}_l^q$ as the vector representation of subtopic l of query q. It can be decomposed into a component aligning with the query and a part that is perpendicular, $\mathbf{e}_l^q = \lambda_l \mathbf{e}_q + \mathbf{e}_l$, with the latter as the additional information in subtopics that clarifies different modalities of the query. Proposition (i) becomes to show whether a neural network is able to infer this subtopic component $\mathbf{e}_l$ from the training data query $\mathbf{e}_q$ and document representations $\{\mathbf{e}_i\}$.

In general, the subtopic vector can be further decomposed into a query-specific component and a general component,

$$\mathbf{e}_l = \mathbf{f}_l(\mathbf{e}_q) + \mathbf{e}_l^0. \tag{15}$$

Apparently, the general modality $\mathbf{e}_l^0$ can be learned from the training documents and encoded as constant biases in the neural network. While how to encode the query dependent part is less obvious: linear transform functions will fail to generate features that are perpendicular to their variable. Fortunately, the neural networks are nonlinear in nature and are thus capable to learn nonlinear functions $\mathbf{f}_l$ for subtopic representations. A simple example of such nonlinear functions is the projection operation used in Table 1: $\mathbf{f}_l(\mathbf{e}_q) = \mathcal{P}_l \cdot \mathbf{e}_q$. In the coordinate system with axes aligned with subtopic vectors as in the example in Table 1, the projection matrix $\mathcal{P}_l$ equals to 1 at diagonal indices l and 0 otherwise, and is linearly transformed from such a diagonal matrix in general.

In addition, such projection operations also allow the neural networks to demonstrate (ii): learn to retrieve the alignment between a document and a subtopic $\mathbf{e}_l \cdot \mathbf{e}_i$ from the query document latent cross $\mathbf{c}_i$:

$$\mathbf{e}_l \cdot \mathbf{e}_i = (\mathbf{f}_l(\mathbf{e}_q) + \mathbf{e}^0) \cdot \mathbf{e}_i = \mathbf{e}_q \cdot \mathcal{P}_l \cdot \mathbf{e}_i + \mathbf{e}^0 \cdot \mathbf{e}_i = \sum_j f_{l,j}(\mathbf{c}_i) + \mathbf{e}^0 \cdot \mathbf{e}_i,$$

which can be used as a feature in determining the relevance between the document and subtopic. In general, there exist nonlinear transformations in the representation space for neural networks to learn to encode implicitly subtopics and alignments between subtopic and documents – useful features in the diversification task.

Finally, to Proposition (iii), in the latent space, as $E \gg m$, there are many query and subtopic independent, in other words, relevance-neutral directions where the candidate documents are spreading over. The nontrivial distributions of the document lists in these dimensions are useless for relevance ranking but can be utilized to diversify the output list. For example, giving higher scores to documents with less neighbors along these dimensions will end up with a more diverse subtopic-coverage in top ranks. See the example in Table 1. Our α-DCG loss naturally leads the neural network to learn such a scoring rule by exploiting the nontrivial distributions in the query-independent dimensions.

Suppose n_l documents equally relevant to subtopic l are distributed over a subtopic independent dimension $\mathbf{e}_{\text{div}}$ as $\rho_l(x)$ with $x_i \equiv \mathbf{e}_{\text{div}} \cdot \mathbf{e}_i$ a measure on how a document i aligns with this dimension. Then the score function training dynamics of a document of alignment x can be obtained by integrating Eq. 14 over the distribution $\rho_l(x)$,

$$\frac{ds(x)}{dt} - \frac{d\bar{s}}{dt} \propto \frac{1}{n_l}\int d\xi \rho_l(\xi) a[s(x) - s(\xi)],$$

where $\bar{s} = \frac{1}{n_l}\int d\xi \rho_l(\xi) s(\xi)$ is the average score of these n_l documents, a is the positive coefficient computed form the hessian terms of α-DCG loss. Knowing that a is finite for $|s(\xi) - s(x)| \ll T$ and decays rapidly to zero when $|s(z) - s(x)| \gtrsim T$ due to the derivative of the sigmoid funtion, we can expand functions of ξ around x and get approximately,

$$\frac{ds(x)}{dt} - \frac{d\bar{s}}{dt} \propto -\frac{\bar{a}T^2}{n_l}\frac{\rho_l'(x)}{s'(x)}, \tag{16}$$

where $\rho_l'(x)$ and $s'(x)$ are the gradients of distribution and score function on alignment x. At the high score end $s(x) > \bar{s}$, when distribution decays with x, $\rho_l'(x) < 0$, score converges to increase with x, $s'(x) > 0$, and vice versa when $\rho_l'(x) > 0$. As a result, scorer learns to give high scores to documents at low document density ρ_l along the subtopic independent dimensions.

6 EXPERIMENTS

To verify the effectiveness of DALETOR, we experiment on the diversity task benchmarks of TREC 2009 – 2012 Web Track datasets[2], which are derived from ClueWeb09 and commonly tested by existing diversification models. The combined dataset of the four TREC datasets includes 198 queries in total (out of 200, 2 queries with no subtopic judgment are dropped). Each query covers 3 to 8 subtopics and the corresponding candidate documents are labeled in binary at the subtopic level, identified by the TREC assessors. We focus on the TREC official diversity evaluation metrics of α-NDCG@k [13] and ERR-IA [11]@k with $k = 5$ and 10, where the parameter α is set to 0.5 as the default settings in official TREC evaluation program.

6.1 Settings

For a fair comparison with recent methods, we specifically work on a public, pre-processed version of the dataset[3] by Feng et al. [18], which is based on the official TREC judgements on the ClueWeb09 Category B data collection. There are 42,245 (40,537 unique) labeled candidate documents associated with the 198 queries. Among them, about one third (13,279) documents contain at least one subtopic. The query and document representation vectors were generated by doc2vec and the dimension of vector representations E was set to 100. Please refer to [18] for more details of the dataset.

We conduct 5-fold cross-validation experiments on the combined dataset with the same subset split as in [18]. At each fold, three subsets were used for training, one was used for hyper-parameter tuning, and one was used for testing. The results reported are the average over the five trials on the testing set in each fold.

Through cross-validation, we choose the following optimizer and model configurations: the optimizer is "Adagrad" [17] with learning rate $\eta = 0.01$. The univariate neural scorer contains three hidden layers with dimensions equal to 256, 128, and 64, followed by a one-dimension dense output layer to compute scores. When applicable, the listwise context embedding is composed of $L = 2$ self-attention layers with $H = 2$ attention heads in each layer with head size $z = 256$. Finally, the smoothness parameter of the α-DCG loss and its variants is set to $T = 0.1$.

We compare with the following baseline methods, including several recent state-of-the-art ones: **MMR** [10]: a heuristic approach with the documents selected sequentially according to maximal marginal relevance; **xQuAD** [34]: a representative method which models subtopics of the original query with sub-queries; **PM-2** [14]: a heuristic method of optimizing proportionality for search result diversification; **SVM-DIV** [44]: a learning approach which utilizes structural SVMs to optimize subtopic coverage; **R-LTR** [46]: a learning approach developed in the relational learning to rank framework; **PAMM** [40]: a learning approach that optimizes α-NDCG using structured Perceptron;

All methods above are taking classical aggregated features as input, while methods below are taking the distributed representations as input. **NTN-DIV** [41]: a learning approach which learns novelty features based on neural tensor networks, PAMM-NTN in specific to directly optimize α-NDCG@10; **MDP-DIV** [42]: a state-of-the-art reinforcement learning approach which uses a Markov Decision Process (MDP) to model the diverse ranking process; **M^{2}DIV** [18]: a state-of-the-art reinforcement learning approach which incorporates Monte Carlo Tree Search (MCTS) to enhance the MDP. **M^{2}DIV-LC**: The variant of M^{2}DIV, with latent cross features fed in as an input to the LSTM model[4].

We investigate the following models in our DALETOR framework: **DNN (softmax)**: a deep univariate scoring model with no latent cross or listwise context embedding, trained with the listwise softmax loss using number of covered subtopics as labels, which serves as a baseline that is not diversification-aware. **DNN (R-LTR)**: a deep univariate scoring model with no latent cross or listwise context embedding, trained with the ListMLE loss using scores from a greedy solution optimizing α-NDCG as labels, which serves as a diversification-aware baseline. **DNN (α-DCG)** and **DNN-LC (α-DCG)**: a deep univariate scoring model without listwise context embedding trained with the α-DCG loss, without and with latent cross. **DIN (α-DCG)** and **DIN-LC (α-DCG)**: a document interaction network model with listwise context trained with the α-DCG loss, without and with latent cross. We also test the other option to incorporate the listwise context with groupwise scoring functions (GSF) [3] and report the best among group sizes tuned over 4, 16, and 64. The results of **GSF (α-DCG)** and **GSF-LC (α-DCG)** are without and with latent cross respectively.

Table 2: Performance comparison with baselines. '*' indicates statistically significant improvement over M^{2}DIV. The best results are bolded.

Method	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
MMR	0.2753	0.2979	0.2005	0.2309
xQuAD	0.3165	0.3941	0.2314	0.2890
PM-2	0.3047	0.3730	0.2298	0.2814
SVM-DIV	0.3030	0.3699	0.2268	0.2726
R-LTR	0.3498	0.4132	0.2521	0.3011
PAMM	0.3712	0.4327	0.2619	0.3029
NTN-DIV	0.3962	0.4577	0.2773	0.3285
MDP-DIV	0.4189	0.4762	0.2988	0.3494
M^{2}DIV	0.4429	0.4839	0.3445	0.3658
DNN(softmax)	0.4280	0.4676	0.3293	0.3496
DNN(R-LTR)	0.4149	0.4517	0.3265	0.3454
DNN-LC(α-DCG)	0.4968*	**0.5322***	0.3868*	0.4068*
DIN-LC(α-DCG)	**0.5009***	0.5294*	**0.3942***	**0.4119***

Table 3: Benefits of the α-DCG loss, by comparing with R-LTR loss. '*' indicates statistically significant improvement over DNN(R-LTR). '†' indicates statistically significant improvement over DNN-LC(R-LTR).

Method	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
DNN(R-LTR)	0.4149	0.4517	0.3265	0.3454
DNN(α-DCG)	0.4614*	0.5005*	0.3633	0.3838*
DNN-LC(R-LTR)	0.4451	0.4842	0.3483	0.3690
DNN-LC(α-DCG)	**0.4968**†	**0.5322**†	**0.3868**†	**0.4068**†

[2] https://plg.uwaterloo.ca/ trecweb/
[3] https://github.com/sweetalyssum/M2DIV/

[4] We reproduced M^{2}DIV and M^{2}DIV-LC of the MCTS methods, but recalled their public results on other baselines in [18].

Table 4: Performance of the latent cross features and the groupwise scoring functions. '*' indicates statistically significant improvement over DNN(α-DCG). '†' indicates statistically significant improvement over DIN(α-DCG).

Method	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
M^{2}DIV	0.4429	0.4839	0.3445	0.3658
M^{2}DIV-LC	0.4551	0.4971	0.3509	0.3735
DNN(α-DCG)	0.4614	0.5005	0.3633	0.3838
DNN-LC(α-DCG)	0.4968*	**0.5322***	0.3868	0.4068
DIN(α-DCG)	0.4615	0.5041	0.3582	0.3808
DIN-LC(α-DCG)	**0.5009**†	0.5294	**0.3942**†	**0.4119**
GSF(α-DCG)	0.4568	0.5023	0.3569	0.3802
GSF-LC(α-DCG)	0.4865	0.5219	0.3786	0.4003

6.2 Experimental results

The performance of different methods is reported in Tables 2, 3, and 4, where boldface indicates the highest scores among all methods in each metric. All statistical significance tests are under the double tailed t-test with p-value< 0.01 indicated by superscripts '$*$' and '†'. We can make several main observations from the results in these tables:

- Our complete model, DIN-LC(α-DCG), outperforms all the baseline models by a large margin, as shown in Table 2. The improvements are statistically significant in terms of both α-NDCG and ERR-IA metrics when compared with the state-of-the-art implicit method M^{2}DIV.
- The benefits of our diversification-aware losses are shown in Table 3. We compare our α-DCG loss and the R-LTR loss under both DNN and DNN-LC in this table. We can see that DNN(α-DCG) improves upon the baseline DNN(R-LTR) by more than 10%, so as DNN-LC(α-DCG) against the baseline DNN-LC(R-LTR).
- Our methods can effectively leverage the distributed representation. On one hand, using distributed representations of query and documents as input features directly improves the results by comparing DNN(R-LTR) and R-LTR, seen in Table 2. On the other hand, incorporating the latent cross features based on the distributed representation adds up another 8% increase in terms of α-NDCG and ERR-IA metrics consistently over DNN and DIN as in Table 4.
- Worth noting in Table 4, though the latent cross features also slightly help M^{2}DIV, they appear to bring much larger gain within our DALETOR framework.
- While slightly better, DIN-LC(α-DCG) does not show statistical significance when compared with DNN-LC(α-DCG). On the TREC dataset, the distributions of the candidate lists are consistent over the training and testing sets, so DNN itself is sufficient to encode information needed for the diversification task. In Sec. 6.2.3 we will show that the context-aware DIN models are more robust to the training-testing skew with perturbed testing sets.

6.2.1 Variants of α-DCG loss. We investigate different variants of α-DCG loss to show its extendability and robustness in terms of hyper-parameters. All experiments in this section are built on the DNN-LC base. We report α-DCG loss with different smoothness parameters T and Gumbel α-DCG loss ($\beta = 10$) defined in Eq.(9). The results are summarized in Table 5.

Table 5: Performance comparison of variants of α-DCG loss

Method	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
α-DCG (T=0.1)	0.4968	0.5322	**0.3868**	**0.4068**
α-DCG (T=1.0)	0.4811	0.5184	0.3703	0.3912
α-DCG (T=0.01)	0.4715	0.4978	0.3633	0.3799
Gumbel-α-DCG	**0.4970**	**0.5339**	0.3855	0.4066

From variants of approximation smoothness T, we find that the performance is robust in general and still outperforms existing methods. However, there exists an optimal smoothness with T around 0.1. Either too smooth $T = 1.0$ or too sharp $T = 0.01$ of the approximation makes the learning less efficient. This is intuitive since very sharp approximation of the metric (i.e., $T \to 0$) makes the gradient of the loss almost zero everywhere so that the learning becomes impossible. On the other end, very smooth approximation (i.e. $T \to \infty$) makes the loss deviate too far from the metric.

We also find some marginal but statistically insignificant improvement with stochastic treatment of the α-DCG loss. It shows our framework is easily extendable, and such modifications might be more significant in other datasets.

Table 6: Performance comparison of variants of self attention layers

(L, H, z)	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
(1, 1, 256)	0.4724	0.5182	0.3679	0.3915
(1, 2, 256)	0.4761	0.5139	0.3706	0.3908
(1, 3, 256)	0.4893	0.5224	0.3801	0.3993
(1, 4, 256)	0.4895	0.5252	0.3810	0.4010
(2, 1, 256)	0.4918	0.5299	0.3842	0.4052
(2, 2, 256)	0.5009	0.5294	0.3942	0.4119
(2, 3, 256)	0.4902	0.5224	0.3800	0.3991
(2, 4, 256)	**0.5066**	**0.5344**	**0.3950**	**0.4122**
(2, 2, 128)	0.4931	0.5295	0.3842	0.4048
(2, 2, 64)	0.4908	0.5245	0.3796	0.3993

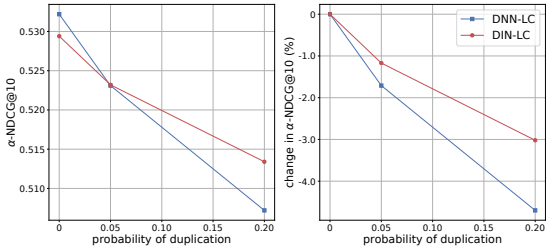

Figure 2: Model performance on perturbed dataset. Left: α-NDCG@10 metric. Right: Percentage change in α-NDCG@10 metric.

6.2.2 Variants of self-attention layers. We study the variants of listwise context embedding by tuning the self-attention layer hyperparameters in terms of number of attention layers L, number of attention heads in each layer H, and the size of each attention head z. In Table 6, we have performances of DIN models with number of layers $L = 1, 2$ and number of heads $H = 1, 2, 3, 4$ given head size $z = 256$ and DIN models with head size $z = 64, 128$, and 256 given $L = H = 2$.

In Table 6, we find that the performance of DIN model with $L = 2$ is consistently better than the DIN model with $L = 1$. However, when the model gets larger with $L > 2$, it becomes more difficult to train self-attention layers with the relatively small TREC dataset. The effect of the number of heads is less obvious. The performance tends to increase with the number of heads, but is not very consistent. The trend is more consistent with increasing head size: DIN models with larger head size perform better. But this increase becomes marginal when the head size is comparable to the embedding dimension of the representations.

6.2.3 Effects of listwise context. As we have seen in Table 2, when comparing DIN-LC with DNN-LC, we do not have statistically significant difference in performance by incorporating the listwise context embedding on the TREC dataset. The GSF results reported in Table 4 also show similar observations, despite some minor deficiency of GSF in utilizing the latent cross features. This result is likely due to the fact that DNN is already sufficient to capture the training distribution, which is consistent with the testing distribution, for the diversification task. However, if there is a noticeable difference between training and testing sets, DIN may better model the distribution change captured by the listwise context. In this section, we explore the robustness of the listwise context embedding by artificially introducing duplication into the testing set. Note that the goal of this section is to rigorously show the robustness of DIN. In practice, it is likely that the training and testing distributions are similar. Also, document near-deduplication techniques are recommended when there is a concern of duplicated documents.

We randomly duplicate fraction p of positive candidate documents by n times with n uniformly distributed in $[0, 20]$. We then apply the trained models on these perturbed testing sets.

As shown in Fig. 2, increasing the probability of duplication p, the testing set deviates from the original distribution of the training set and both DNN and DIN model performances reduce. But the reduction of DIN performance is clearly slower than the DNN model. In terms of percentage change in α-NDCG@10, DNN model performance is reduced by 4.7% and DIN model is reduced by 3.0% when about 20% of the positive documents are duplicated.

7 DISCUSSION

Here we discuss the differences between the traditional "next-document" diversification methods and the proposed "score-and-sort" setting with a "soft" version of "next document" to give an intuition on how it is able to promotes diversification. The key difference is that in the "score-and-sort" setting, the scores are predicted simultaneously rather than sequentially as in the "next-document" setting. We try to answer the following two successive questions: how can a scorer properly score the documents without knowing the context? And even if the scorer knows the scoring context, how can it differentiate documents with the same relevance to certain subtopics?

Table 7: Candidate documents with the same subtopic label of Query 78.

	Subtopic label	Score(softmax)	Score(α-DCG)
d_1	[0 0 0 0 0 1 0 0]	0.905	21.418
d_2	[0 0 0 0 0 1 0 0]	0.906	14.785
d_3	[0 0 0 0 0 1 0 0]	0.902	14.292

To the first question, the naive solution is to take the context information as part of input to the scorer, which is what we did by incorporating listwise context using DIN. But we found in Table 2, the performance of DIN models is not significantly better than that of simple univariate models. This indicates that our models were able to learn subtopic distributions over the documents and reasonably diversify the results based on the listise diversification-aware losses.

The second question is special to the "score-and-sort" setting: in the "next-document" setting, we can output a diversified ranking as long as the relevance to each subtopics are inferred, but in the "score-and-sort" setting, we have to consult to features that are "perpendicular" to the relevance measure encoded in the distributed representations as shown in the Section 5. The intuition we got from the relevance-neutral features is that as long as we can score the documents with the same subtopic relevance differently, we will be able to rank the top ones in a diverse way. This is exactly we found from the scorers trained with diversification-aware loss: Table 7 shows several candidate document examples that have the same subtopic label and they are scored by a scorer trained with softmax loss – Score(softmax) and a scorer trained with α-DCG loss – Score(α-DCG). When the model is trained with softmax loss, which is not diversification-aware, it tends to score documents with the same label similarly. However, when the model is trained with the diversification-aware α-DCG loss, it can capture their differences in the latent space and score one document much higher than others to improve diversification.

Another advantage of the "score-and-sort" framework is it allows $O(n)$ inference complexity, which can be done in parallel and significantly reduces serving latency in real-world applications.

8 CONCLUSION

In this work, we introduced a new perspective for learning-based search result diversification. To enable the diversification-aware learning, we introduced a differentiable loss from a close approximation of a commonly used diversity metric, α-DCG in particular. With this differentiable loss, we could then train a neural network end-to-end from distributed representations of queries and documents. We further leveraged the latent cross of the distributed representations and the listwise context, resulting in a deep ranker that performs significantly better than state-of-the-art methods on the TREC dataset in terms of various diversity evaluation metrics. We further elaborated that how our model learns subtopics implicitly from distributed representations, how the approximate α-DCG loss promotes diversity by learning the distribution of the candidate list from subtopic-independent features.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying Search Results. In *International Conference on Web Search and Data Mining (WSDM)*. 5–14.

[2] Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 135–144.

[3] Qingyao Ai, Xuanhui Wang, Sebastian Bruch, Nadav Golbandi, Mike Bendersky, and Marc Najork. 2019. Learning Groupwise Multivariate Scoring Functions Using Deep Neural Networks. In *ICTIR*. 85–92.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[5] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *International Conference on Web Search and Data Mining (WSDM)*. 46–54.

[6] Andrei Z. Broder. 2000. Identifying and Filtering Near-Duplicate Documents. In *Annual Symposium on Combinatorial Pattern Matching*. 1–10.

[7] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. 2020. A Stochastic Treatment of Learning to Rank Scoring Functions. In *International Conference on Web Search and Data Mining (WSDM)*. 61–69.

[8] Sebastian Nima Bruch, Masrour Zoghi, Mike Bendersky, and Marc Najork. 2019. Revisiting Approximate Metric Optimization in the Age of Deep Neural Networks. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 1241–1244.

[9] Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to Rank with Nonsmooth Cost Functions. In *NeurIPS*. 193–200.

[10] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 335–336.

[11] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In *ACM Conference on Information and Knowledge Management (CIKM)*. 621–630.

[12] Olivier Chapelle, Donald Metzler, Ya Zhang, and Pierre Grinspan. 2009. Expected Reciprocal Rank for Graded Relevance. In *ACM Conference on Information and Knowledge Management (CIKM)*. 621–630.

[13] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 659–666.

[14] Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 65–74.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 4171–4186.

[16] Pinar Donmez, Krysta M Svore, and Christopher JC Burges. 2009. On the local optimality of lambdarank. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 460–467.

[17] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12 (July 2011), 2121–2159.

[18] Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2018. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 125–134.

[19] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. 2019. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850* (2019).

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

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[22] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *ACM Conference on Information and Knowledge Management (CIKM)*. 2333–2338.

[23] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. 2017. Learning to Diversify Search Results via Subtopic Attention. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 545–554.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Neural Information Processing Systems (NeurIPS)*. 3146–3154.

[25] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *International Conference on Machine Learning (ICML)*. 1188–1196.

[26] Jiongnan Liu, Zhicheng Dou, Xiaojie Wang, Shuqi Lu, and Ji-Rong Wen. 2020. DVGAN: A Minimax Game for Search Result Diversification Combining Explicit and Implicit Features. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 479–488.

[27] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[28] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2019. SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval. *arXiv:1912.05891* (2019).

[29] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. 2019. TF-Ranking: Scalable tensorflow library for learning-to-rank. In *ACM SIGKDD International Conference on KnowledgeDiscovery and Data Mining (KDD)*. 2970–2978.

[30] Rama Kumar Pasumarthi, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2020. Permutation Equivariant Document Interaction Network for Neural Learning to Rank. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 145–148.

[31] Tao Qin, Tie-Yan Liu, and Hang Li. 2010. A general approximation framework for direct optimization of information retrieval measures. *Information Retrieval* 13, 4 (2010), 375–397.

[32] Xubo Qin, Zhicheng Dou, and Ji-Rong Wen. 2020. Diversifying Search Results using Self-Attention Network. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*.

[33] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?. In *Proceedings of the The Ninth International Conference on Learning Representations (ICLR)*.

[34] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *The Web Conference (WWW)*. 881–890.

[35] Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. 2010. Learning Optimally Diverse Rankings over Large Document Collections. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10)*. 983–990.

[36] Alex Slivkins, Filip Radlinski, and Sreenivas Gollapudi. 2010. Learning optimally diverse rankings over large document collections. (2010).

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[38] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. 2018. The LambdaLoss Framework for Ranking Metric Optimization. In *ACM Conference on Information and Knowledge Management (CIKM)*. 1313–1322.

[39] Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, Sagar Jain, Ed H Chi, and Jennifer Gillenwater. 2018. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2165–2173.

[40] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 113–122.

[41] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2016. Modeling document novelty with neural tensor network for search result diversification. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 395–404.

[42] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2017. Adapting Markov Decision Process for Search Result Diversification. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 535–544.

[43] Sevgi Yigit-Sert, Ismail Sengor Altingovde, Craig Macdonald, Iadh Ounis, and Özgür Ulusoy. 2020. Supervised approaches for explicit search result diversification. *Information Processing & Management* 57, 6 (2020), 102356.

[44] Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural SVMs. In *International conference on Machine learning (ICML)*. 1224–1231.

[45] ChengXiang Zhai, William W. Cohen, and John D. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 10–17.

[46] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. 293–302.