Optimize What You Evaluate With: Search Result Diversification Based on Metric Optimization

Hai-Tao Yu

Information Intelligence Lab
Faculty of Library, Information and Media Science, University of Tsukuba
yuhaitao@slis.tsukuba.ac.jp

Abstract

Most of the existing methods for *search result diversification* (SRD) appeal to the *greedy strategy* for generating diversified results, which is formulated as a sequential process of selecting documents one-by-one, and the locally optimal choice is made at each round. Unfortunately, this strategy suffers from the following shortcomings: (1) Such a one-by-one selection process is rather time-consuming for both training and inference. (2) It works well on the premise that the preceding choices are optimal or close to the optimal solution. (3) The mismatch between the objective function used in training and the final evaluation measure used in testing has not been taken into account. We propose a novel framework through direct metric optimization for SRD (referred to as *MO4SRD*) based on the *score-and-sort strategy*. Specifically, we represent the diversity score of each document that determines its rank position based on a probability distribution. These distributions over scores naturally give rise to expectations over rank positions. Armed with this advantage, we can get the differentiable variants of the widely used diversity metrics. Thanks to this, we are able to directly optimize the evaluation measure used in testing. Moreover, we have devised a novel probabilistic neural scoring function. It jointly scores candidate documents by taking into account both cross-document interaction and permutation equivariance, which makes it possible to generate a diversified ranking via a simple sorting. The experimental results on benchmark collections show that the proposed method achieves significantly improved performance over the state-of-the-art results.

Introduction

To cope with ambiguous and/or underspecified queries, *search result diversification* (SRD) has been regarded as the key solution and has shown significantly increasing values in a wide range of domains, such as *web search* (Ma, Lyu, and King 2010; Liu et al. 2014; Liang 2019) and *recommender systems* (Liu et al. 2020b; Ding et al. 2021). According to *whether the subtopics (i.e., different information needs) underlying a query are given beforehand or not*, the task of SRD can be distinguished into *implicit SRD* and *explicit SRD*. The distinguishing characteristic of the implicit SRD is that the possible subtopics underlying a query are *unknown*. From a machine learning perspective, we can also classify SRD methods into unsupervised methods and supervised methods. The key difference is that unsupervised methods mainly rely on some heuristic criteria for generating the diversified ranking and no machine learning algorithms is used. Noteworthy, finding a group of subtopic strings that covers well all the possible information needs behind the query is a challenging task. In most realistic scenarios explicit subtopics are not available (Kim and Lee 2015). In this paper, we focus on how to perform implicit SRD in a supervised manner. The information retrieval community has experienced a flourishing development of SRD methods, such as the methods (Santos, Macdonald, and Ounis 2010; Dang and Croft 2012; Yu and Ren 2014; Dang and Croft 2013; Hu et al. 2015; Sarwar et al. 2020) for explicit SRD and the methods (Carbonell and Goldstein 1998; Sanner et al. 2011; Gollapudi and Sharma 2009; Zuccon et al. 2012; Raiber and Kurland 2013; Yu et al. 2018, 2017) for implicit SRD. Later on, due to the breakthrough successes of neural network based models, significant efforts (Zhu et al. 2014; Xu et al. 2020; Liu et al. 2020a; Yigit-Sert et al. 2020; Xia et al. 2015, 2017, 2016; Xu et al. 2017; Jiang et al. 2017; Feng et al. 2018) have been made in exploring how to deploy machine learning methods, especially neural network based models, to solve SRD problems.

Despite the successes achieved by the aforementioned studies, fundamental research questions remain open. First, most studies appeal to formulate diversified document ranking as a sequential process, where the locally optimal choice is made at each round. The key drawback is that the commonly used greedy strategy works well on the premise that the preceding choices are optimal or close to the optimal solution. However, in most cases, this strategy fails to guarantee the optimal solution. What is more, this one-by-one selection process is rather time-consuming with the increase of the number of candidate documents, which poses an additional challenge. Second, in order to overcome the aforementioned shortcoming, Qin, Dou, and Wen (2020) explored how to generate diversified ranking in a score-and-sort manner. Unfortunately, the mismatch between the objective function used in training and the final evaluation measure used in testing has not been taken into account. Third, the recent work by Yan et al. (2021) showed significantly improved performance by directly optimizing the smooth approximation of a specific diversity metric. How-

ever, a global tuning parameter is required when deriving the approximated rank of each document, which leads to impacted performance. Fourth, using a deterministic scoring function fails to capture *uncertainty*, which is an inherent part of the learning process of a model and the retrieval process, such as *parameter uncertainty* (different combinations of weights that explain the data equally well), *structural uncertainty* (which neural architecture to use for neural ranking), and *aleatoric uncertainty* (noisy data). In view of the fact that the SRD problem is bound to attract more attention in the era of big data, the aforesaid shortcomings motivate us to approach SRD in a novel way. In this paper, we propose a novel framework for SRD based on the *score-and-sort strategy*. The main contributions are listed as follows:

(1) We propose a novel probabilistic scoring function. On the one hand, we represent the diversity score of each document that determines its rank position based on a probability distribution (the controlling parameters can be learned automatically) rather than a deterministic value. These distributions over scores naturally give rise to expectations over rank positions. On the other hand, we jointly score candidate documents by taking into account both cross-document interaction and permutation equivariance, which makes it possible to generate a diversified ranking via a simple sorting.

(2) Thanks to the easy computation of expected rank of each document, we can derive differentiable reformulations of the widely used diversity metrics. These differentiable reformulations naturally give rise to better optimization objectives for SRD, which ensures the consistency between the objective function used in training and the final evaluation measure used in testing. Noteworthy, no tuning parameter is required within the differentiable reformulation.

Related Work

In this section, we discuss related studies on SRD by classifying them into two groups. Due to space constraints, for an in-depth overview of SRD, we refer the reader to the work (Santos, Macdonald, and Ounis 2015). For the first group, *relevance* and *diversity* are quantified respectively. As a result, an explicit mechanism of balancing relevance and diversity becomes the core of methods of this kind. In general, relevance denotes to what extent a document provides useful information. Diversity denotes the marginal benefit of adding a document. These methods differ mainly in the following aspects: (1) how to represent diversity; (2) how to balance relevance and diversity and (3) how to generate the diversified ranking. For example, a typical instance is the *maximal marginal relevance* (MMR) model, which measures the diversity of a document based on the maximum similarity between this document and the previously selected documents. Later on, Guo and Sanner (2010) present a probabilistic latent view of MMR. Another line of studies build upon the cluster hypothesis (Rijsbergen 1979), which states that *closely associated documents tend to be relevant to the same requests*. Raiber and Kurland (2013) studied how to incorporate various types of cluster-related information based on Markov Random Fields. The methods (also referred to as *top-k retrieval* in (Zuccon et al. 2012; Gollapudi and Sharma 2009; Yu et al. 2017, 2018)) for implicit SRD perform a two-step process. The first step is to select an optimal subset of documents according to a specific objective function. At the second step, the selected documents are ordered in a particular way, e.g., in a decreasing order of relevance. The methods (Santos, Macdonald, and Ounis 2010; Dang and Croft 2012; Yu and Ren 2014; Dang and Croft 2013; Hu et al. 2015) for explicit SRD assume that the possible aspects representing different information needs of a query are given beforehand. For instance, the xQuAD framework (Santos, Macdonald, and Ounis 2010) downweights each subtopic based on the degree of its relevance to the already selected documents, thus the subtopics with less relevant documents will have a higher priority in the next round. Recently, many efforts (Zhu et al. 2014; Xu et al. 2020; Liu et al. 2020a; Yigit-Sert et al. 2020; Xia et al. 2015, 2017, 2016; Xu et al. 2017; Jiang et al. 2017; Feng et al. 2018) have been made to use machine learning technologies to train the diversification model. The advantages are straightforward. On one hand, it is easy to incorporate a large number of features into a specific diversification method. On the other hand, decades of work on machine learning can be leveraged to optimize the ranking functions. Compared with the unsupervised methods for either explicit SRD or implicit SRD, the diversification models (Radlinski, Kleinberg, and Joachims 2008; Xia et al. 2016; Yue and Joachims 2008; Xia et al. 2017; Jiang et al. 2017) based on machine learning technologies can achieve significantly improved performance.

For the second group, inspired by *bidirectional encoder representations from transformers* (BERT) and its variants, recent efforts (Yan et al. 2021; Qin, Dou, and Wen 2020) have been made to develop methods based on the score-and-sort strategy. In particular, the multi-head self-attention layer is a key component for incorporating cross-document interactions. Finally, the output is formulated as a univariate score, which determines a document's rank position. In this paper, we also use the score-and-sort strategy. Compared with Qin, Dou, and Wen (2020), our training objective is tightly related to the evaluation metric, which ensures the consistency between the objective function used in training and the final evaluation measure used in testing. In order to obtain the differentiable variant of the diversity metric, Yan et al. (2021) approximate the indicator function with a vanilla sigmoid as introduced by Qin, Liu, and Li (2010). Unfortunately, a predefined global tuning parameter is required and has to be fine-tuned, which limits the final performance. We note that Xia et al. (2015) and Xu et al. (2017) also explored how to directly optimize diversity metrics, but the final result is generated in a greedy manner.

Preliminaries

In this section, we first introduce the general SRD framework following the Cranfield paradigm. Then we review two widely used diversity evaluation metrics.

Cranfield Search Result Diversification

Let $\mathcal{Q}$ and $\mathcal{D}$ be the query space and the document space, respectively. We use $\Phi : \mathcal{Q} \cup \mathcal{D} \rightarrow \mathbb{R}^d$ to denote the mapping

function for generating vector representations for documents and queries, where d represents the dimension size. For a query, there are h subtopics, where a *subtopic* refers to a different search intent or information need. Given a query q, we have a list of candidate documents $D = (d_1, ..., d_m)$ and a corresponding list of ground-truth labels $Y = (\mathbf{y}_1, ..., \mathbf{y}_m)$, where $Y \in \mathcal{T} := \mathbb{R}_{\geq 0}^{h \times m}$ and $\mathcal{T}$ denotes the space of the ground-truth labels. The subscript i as in d_i or $\mathbf{y}_i$ denotes the i-th position in the list. Moreover, $\mathbf{y}_i \in \mathbb{R}_{\geq 0}^{h}$ is a column vector that denotes the relevance assessment of document d_i with respect to each subtopic. In other words, Y_{ki} denotes the relevance label of document d_i with respect to the k-th subtopic. Noteworthy, the number of subtopics and the number of documents may differ from query to query. In practice, we get independently and identically distributed (i.i.d) samples $\mathcal{S} = \{(q^j, D^j, Y^j)\}_{j=1}^{n}$ from an unknown joint distribution $P(\cdot, \cdot, \cdot)$ over $\mathcal{Q} \times \mathcal{D} \times \mathcal{T}$. The superscript j denotes the data that are associated with the same query, which is omitted if the context provides sufficient clarity. We use π to denote a diversified ranking on candidate documents $D = (d_1, ..., d_m)$, and $\pi(i)$ / $\pi(d_i)$ yields the *rank position* of the i-th document in the diversified ranking. An ideal ranking refers to the optimal ranking of documents that systematically accounts for redundancy and ambiguity, which maximizes the likelihood that an average user can find documents relevant to her specific need. We use f parameterized by $\theta \in \Theta$ to denote a general ranking function. Commonly we measure the quality of ranking documents for a query using f with a loss function $\mathcal{R}(f(q, D), Y)$. We would like to learn the optimal function over a hypothesis space $\mathcal{F}$ of ranking functions that can *minimize the expected risk* as:

$$\min_{f \in \mathcal{F}} \Re(f) = \min_{f \in \mathcal{F}} \int_{\mathcal{Q} \times \mathcal{D} \times \mathcal{T}} \mathcal{R}(f(q, D), Y) dP(q, D, Y) \tag{1}$$

Unfortunately, $\Re(f)$ is intractable to optimize directly and the joint distribution is unknown, so instead we appeal to the *empirical risk minimization* to approximate the expected risk, which is defined as follows:

$$\min_{f \in \mathcal{F}} \tilde{\Re}(f; \mathcal{S}) = \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^{n} \mathcal{R}(f(q^j, D^j), Y^j) \tag{2}$$

Given the above general framework, different formulations of the ranking function and the loss function yield different models.

Diversity Evaluation Metrics

Before reviewing the diversity metrics, we note that $I(Y_{ki})$ and $R(Y_{ki})$ are two functions that map a ground-truth relevance label to a numerical value or the probability of being relevant. $I(Y_{ki}) = 1$ if $Y_{ki} > 0$, otherwise $I(Y_{ki}) = 0$, it is used by α-$nDCG$. $R(Y_{ki})$ used by ERR and ERR-IA is defined as $\frac{2^{Y_{ki}}-1}{2^{Y^{max}}}$, where Y^{max} denotes the maximum ground-truth relevance value across the dataset.

$\boldsymbol{\alpha}$**-nDCG** (novelty-biased Discounted Cumulative Gain) (Clarke et al. 2008) extends the standard metric of nDCG (normalised Discounted Cumulative Gain) (Järvelin and Kekäläinen 2002) by rewarding newly retrieved subtopics and penalizing redundant subtopics. They assume binary relevance assessments, and use parameter α to reflect the possibility of assessor error. The gain value for document d_i is computed by summing over subtopics, namely, $G_i = \sum_{k=1}^{h} I(Y_{ki})(1-\alpha)^{c_{ki}}$, where $c_{ki} = \sum_{j:\pi(j)<\pi(i)} Y_{kj}$ denotes the number of times that the k-th subtopic has been covered by documents ranked above document d_i. The discounted cumulative gain is expressed as:

$$\alpha\text{-}DCG = \sum_{i=1}^{m} \frac{\sum_{k=1}^{h} I(Y_{ki})(1-\alpha)^{c_{ki}}}{log_2(\pi(i)+1)} \tag{3}$$

To compare the scores across various queries, α-DCG is commonly normalized, namely α-$nDCG = \frac{\alpha\text{-}DCG}{\alpha\text{-}DCG^*}$ where α-DCG^* denotes the maximum α-DCG value attained by the ideal ranking.

ERR-IA (Intent-Aware Expected Reciprocal Rank) is the intent-aware version of ERR (Expected Reciprocal Rank) by Chapelle et al. (2009). The underlying intuition of ERR-IA is to perform evaluation by applying a traditional metric to each subtopic independently and then combine the results based on the importance or probability of subtopics (denoted as $P(T_k|q)$), which is expressed as:

$$ERR-IA = \sum_{k=1}^{h} P(T_k|q) \sum_{i=1}^{m} \frac{1}{\pi(i)} R(Y_{ki}) \prod_{j:\pi(j)<\pi(i)} (1\text{-}R(Y_{kj})) \tag{4}$$

A closer look at the computation of α-$nDCG$ and ERR-IA reveals that they rely on the positions at which documents are ranked. Unfortunately, the rank information is commonly obtained via a traditional sorting algorithm (e.g., *Quicksort* (Hoare 1962)) or determined during the sequential selection process, which makes it hard to directly optimize the metrics. Taking the case of direct sorting for example, when we make small changes to the model parameters of a univariate scoring function, the output scores will typically change smoothly. In contrast, the ranks of documents will not change until the documents' scores exceed one another. Hence the metrics will make a discontinuous change. In other words, the metrics are non-smooth with respect to the model parameters, being everywhere either flat (with zero gradient) or discontinuous.

Proposed Method

In this section, we detail our proposed method named as MO4SRD. First, we describe the proposed probabilistic scoring function, including initial representations, major component layers and the scoring pipeline. We then show how to derive the differentiable reformulations of the widely used diversity metrics as the optimization objective.

Probabilistic Scoring Function

Our proposed probabilistic scoring function has two main characteristics. The first characteristic is that: different from most prior studies that treat *relevance* and *diversity* respectively, our scoring function uses an integrated univariate score (referred to as *diversity score* in the following) to determine the rank position of a document

in the diversified ranking, which paves the way for deploying the score-and-sort strategy. The second characteristic is that we view the diversity score of a document as a probabilistic value, which follows the Gaussian distribution. In a nutshell, given a query q and the list of candidate documents $D = (d_1, ..., d_m)$, the diversity score s_i of a specific document d_i follows the distribution $P(s_i|q, D) = \mathcal{N}(\mu(d_i|q, D), \sigma^2(d_i|q, D))$, where $\mu(d_i|q, D)$ and $\sigma^2(d_i|q, D)$ denote the mean and variance, respectively. Throughout this paper, we assume that the computation of diversity score, mean and variance for a document is conditioned on the query and other candidate documents. In the following, we will omit the condition of q, D if the context provides sufficient clarity. For instance, $P(s_i|q, D)$ is written as $P(s_i)$ for short.

In order to fulfill the aforementioned characteristics, our scoring function is designed to satisfy the following desiderata. First, in view of the fundamental principle for SRD (the more the information needs are satisfied or covered by the above ranked documents, the less the marginal benefit of a low-ranked document containing similar information provides), the scoring function must be able to characterize cross-document interaction among the candidate documents when generating the diversified result list. Second, given the requirement of coping with cross-document interaction among the candidate documents, a natural choice is to score the documents together. A desired property for such a scoring function is that the order of input documents should have no effect on the output, which is referred to as *permutation equivariance*. Third, the document-specific controlling parameters of a Gaussian distribution, such as mean and variance, should be automatically learned rather than using a globally predefined setting.

Next we elaborate on the major components including initial vector representation, multi-head self-attention layer and probabilistic regress layer. Finally, we describe the overall scoring pipeline.

Initial Vector Representation For implementing the mapping function Φ which is used to generate vector representations for documents and queries, there are a number of choices, such as using heuristic aggregated ranking features (e.g., BM25 and TF-IDF) and the pre-trained BERT model (Devlin et al. 2019). Following the prior studies (Yan et al. 2021; Qin, Dou, and Wen 2020), we appeal to the doc2vec model (Le and Mikolov 2014) in this paper, which also enables us to make a fair comparison. Specifically, the text-based queries and documents are mapped into dense normalized vectors $\mathbf{x}_q \in \mathbb{R}^e$ and $\mathbf{x}_i \in \mathbb{R}^e$ of a fixed embedding size e. Inspired by the work (Yan et al. 2021; Qin et al. 2021), in order to well capture the interaction between query-document pairs, we also use the algorithm of *latent cross* (Beutel et al. 2018) to generate query-document cross feature. Given the vector representations $\mathbf{x}_q$ and $\mathbf{x}_i$ for a pair of query and document, the query-document cross feature is defined as: $\mathbf{c}_i = \mathbf{x}_q \odot \mathbf{x}_i$, where $\odot$ denotes element-wise multiplication.

Multi-Head Self-Attention Layer Inspired by the recent studies (Yan et al. 2021; Qin, Dou, and Wen 2020; Pang et al. 2020; Qin et al. 2021; Pasumarthi et al. 2020), we incorporate multi-head self-attention layer into our scoring function. Thanks to this, the scoring function is able to cope with cross-document interaction and preserve permutation equivariance. Please refer to (Pasumarthi et al. 2020; Pang et al. 2020) for the theoretical demonstration. Given the input matrix $M \in \mathbb{R}^{m\times a}$ corresponding to m candidate documents (a denotes the dimension size), the attention layer in Transformer (Vaswani et al. 2017) is formulated based on three projection matrices: $W^Q \in \mathbb{R}^{a\times b}$, $W^K \in \mathbb{R}^{a\times b}$ and $W^V \in \mathbb{R}^{a\times a}$ (where b is the projection size). Then we project M into a query[1] matrix $Q = MW^Q$, a key matrix $K = MW^K$ and a value matrix $V = MW^V$, respectively. At a high level, attention is a pooled combination of values of V across documents, weighted by pairwise scaled dot product similarity matrix $A(M)$ between query matrix Q and key matrix K:

$$A(M) = \frac{QK^\intercal}{\sqrt{b}} \tag{5}$$

Using these weights, a self-attention layer computes a weighted sum of V as follows:

$$SA(M) = Softmax(A(M))V \tag{6}$$

The study by Transformer (Vaswani et al. 2017) have shown that using multiple heads, which attend on different parts of the input, can be beneficial. For U heads, the output of multiple self-attention layers per head are concatenated and projected via a linear transformation using matrices $W_{out} \in \mathbb{R}^{Ua\times a}$ and bias term $b_{out} \in \mathbb{R}^a$ in order to ensure the output $MHSA(M) \in \mathbb{R}^{m\times a}$:

$$MHSA(M) = concat_{u\in[U]}[SA_t(M)]W_{out} + b_{out} \tag{7}$$

Additionally, we also apply residual connections and layer normalization. They also preserve the property of permutation equivariance due to the element-wise operations.

Probabilistic Regression Layer Inspired by the model of MDN (mixture density network) (Bishop 1994), we formulate the target distribution with respect to a document as a GMM (Gaussian mixture model):

$$P(s_i|q, D) = PRL(\mathbf{v}_i) = \sum_{j=1}^{V} \rho_j \mathcal{N}(s_j|\mu_j(\mathbf{v}_i), \sigma_j^2(\mathbf{v}_i)) \tag{8}$$

where $\mathbf{v}_i$ denotes the input vector corresponding to document d_i, $\{\rho_j, \mu_j, \sigma_j^2\}$ is a set of parameters of a GMM, namely mixture probabilities, mixture means, and mixture variances, respectively. One important analytical property that makes the Gaussian distribution extremely tractable is its closure under linear combinations, namely the linear combination of independent random variables having a Gaussian distribution also has a Gaussian distribution. Thanks to this, the controlling parameters of each distribution for representing the diversity score of a document are directly parameterized by the network architecture itself.

[1]We note that the *query* here is different from the search query.

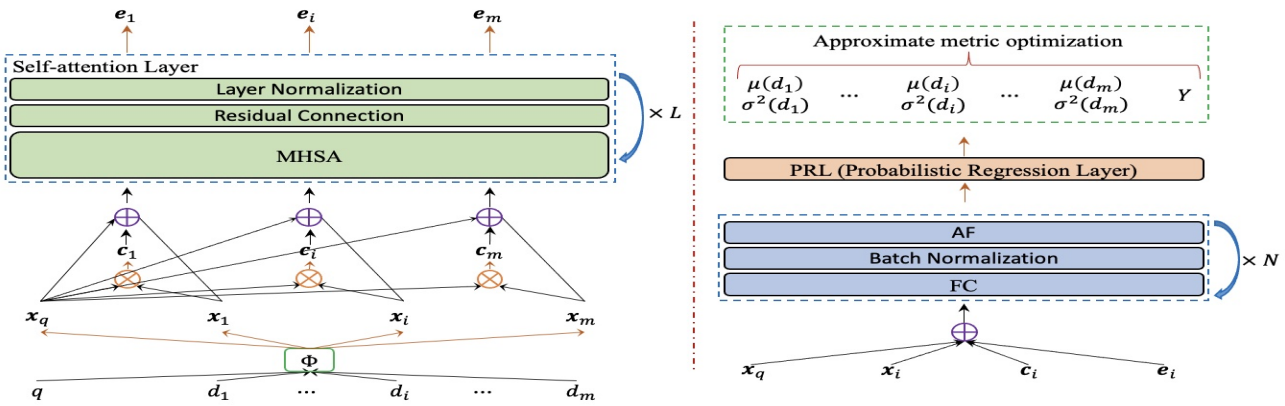

Figure 1: The proposed end-to-end framework for SRD.

Scoring Pipeline In Figure 1, we show the overall structure of the proposed scoring function. Given a query q and candidate documents D, we first get their vector representations based on the adopted mapping function Φ. As the input of stacked multi-head self-attention layers, we concatenate query vector $\mathbf{x}_q$, document vector $\mathbf{x}_i$ and query-document cross feature vector $\mathbf{c}_i$ to obtain the listwise contextual representation $\mathbf{e}_i$ for each document. Then, query vector $\mathbf{x}_q$, document vector $\mathbf{x}_i$, query-document cross feature vector $\mathbf{c}_i$ and listwise contextual representation $\mathbf{e}_i$ are concatenated and passed through a number of fully connected layers to compute the input vector of PRL. The output of PRL corresponds to the predicted diversity score for each document.

Approximation of Expected α-$nDCG$

With the probabilistic formulation of diversity score in place, the probability $P(d_i \triangleright d_j)$ that document d_i beats document d_j (i.e., d_i should be ranked above d_j) can be easily expressed as $P(S_i - S_j > 0)$, where S_i and S_j are the draws from $P(s_i)$ and $P(s_j)$, respectively. In other words, this probability is simply the integral of the difference of two Gaussian random variables, which is itself a Gaussian.

$$\begin{aligned} P_{ij} &= P(d_i \triangleright d_j) = P(S_i - S_j > 0) \\ &= \int_0^\infty \mathcal{N}(s|\mu(d_i) - \mu(d_j), \sigma^2(d_i) + \sigma^2(d_j))ds \\ &= \frac{1}{2}[1 + erf(\frac{\mu(d_i) - \mu(d_j)}{\sqrt{2(\sigma^2(d_i) + \sigma^2(d_j))}})] \end{aligned} \tag{9}$$

$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)dt$ is the Gauss error function.

Inspired by (Qin, Liu, and Li 2010; Yan et al. 2021), given the list of candidate documents $D = (d_1, ..., d_m)$, the rank position of the i-th document can be given as

$$\pi(i) = 1 + \sum_{j:j\neq i} \mathbb{I}\{d_j \triangleright d_i\} \tag{10}$$

$\mathbb{I}\{e\}$ is the indicator, which is one if the condition e is true and zero otherwise. Analogously, the number of times that the k-th subtopic has been covered by the above ranked documents can be given as:

$$c_{ki} = \sum_{j:j\neq i} I(Y_{kj})\mathbb{I}\{d_j \triangleright d_i\} \tag{11}$$

Given Equation-10 and Equation-11, we go one step further by taking the expectation, thus we have

$$\mathbb{E}[\pi(i)] = 1 + \sum_{j:j\neq i} P(d_j \triangleright d_i) \tag{12}$$

$$\mathbb{E}[c_{ki}] = \sum_{j:j\neq i} I(Y_{kj})P(d_j \triangleright d_i) \tag{13}$$

With the pairwise comparison (Equation-9) in place, the expected rank of a document and the expected number of times that the k-th subtopic has been covered by documents ranked above document d_i can be easily computed using Equation-12 and Equation-13, respectively.

By replacing the non-differentiable factors $\pi(i)$ and c_{ki} with their expectations $\mathbb{E}[\pi(i)]$ and $\mathbb{E}[c_{ki}]$, we can get the differentiable variants of α-DCG and ERR-IA as follows:

$$\widehat{\alpha\text{-}DCG} = \sum_{i=1}^{m} \frac{\sum_{k=1}^{h} I(Y_{ki})(1-\alpha)^{\mathbb{E}[c_{ki}]}}{log_2(\mathbb{E}[\pi(i)] + 1)} \tag{14}$$

$$\widehat{ERR-IA} = \frac{1}{h}\sum_{i=1}^{m} \frac{1}{\mathbb{E}[\pi(i)]} \sum_{k=1}^{h} R(Y_{ki}) \prod_{j:\mathbb{E}[\pi(j)]<\mathbb{E}[\pi(i)]} (1 - R(Y_{kj})) \tag{15}$$

By defining the negative metric score as the minimization objective, a novel loss function that is a tightly related to the evaluation metric can be obtained correspondingly. Taking α-DCG for example, because $\frac{1}{\log_2(1+t)}$ is convex for $t > 0$, we have $\frac{1}{\log_2(\mathbb{E}[\pi(i)]+1)} \leq \mathbb{E}[\frac{1}{\log_2(\pi(i)+1)}]$ through Jensen's inequality. Therefore, the above approximation of α-DCG is a lower bound of the expected α-DCG. A closer look at Equation-14 reveals that: this formulation closely resembles the idea of SoftRank (2008). The key difference is that Taylor et al. (2008) proposed an approximate way of computing the expected nDCG by introducing the so-called rank-binomial distribution, which suffers from a computational complexity of $\mathcal{O}(m^3)$. The proposed method has the same time complexity as Yan et al. (2021) and Qin, Liu, and Li (2010), which is of order $\mathcal{O}(m^2)$ for a single query at

training time. Thanks to the adopted *score-and-sort* strategy, MO4SRD allows $\mathcal{O}(m)$ inference complexity at test time, which can be done in parallel. On the contrary, the methods relying on greedy selection suffer from a high computational cost of $\mathcal{O}(mk)$ for diversifying the top-k results. We note that a diversification algorithm achieving the best performance can be useless if its high computational cost forbids its use in a real-world applications[2].

Experiments

In this section we report a series of experiments to evaluate the proposed method by comparing it to the state-of-the-art diversification approaches. First, we detail the test collections. Second, we describe the baseline methods and configurations[3]. Finally, we describe the experimental results.

Test Collections

Four standard test collections released in the diversity tasks of TREC Web Track from 2009 to 2012 are adopted for the experiments (50 queries per each year). Each query is structured as a set of representative subtopics. Queries numbered 95 and 100 in TREC 2010 are discarded due to the lack of judgment data, resulting in 198 queries being finally used. For each query, the candidate documents are annotated with a binary relevance label per subtopic by the TREC assessors. We report the results with different cutoff values 5, 10 and 20 to show the performance of each method at different positions, where α for α-$nDCG$ is set as 0.5 so as to keep consistent with the official TREC evaluation program. For a fair comparison with recent methods, the experiments are conducted based on the pre-processed dataset[4] following the practices in (Xia et al. 2017; Feng et al. 2018). Specifically, the ClueWeb09 Category B collection consisting of 50 million English web documents is used as the base. The doc2vec model is trained on all documents and the number of vector dimensions is set as 100. The initial vector representations of queries and documents can be obtained given the trained doc2vec model. Please refer to (Xia et al. 2017; Feng et al. 2018) for more details. We perform 5-fold cross validation experiments following the same subset split as (Feng et al. 2018). In each fold, three subsets are used as the training data, the remaining two subsets are used as the validation data and the testing data. We use the training data to learn the ranking model, use the validation data to select the hyper parameters, and use the testing data for evaluation. Finally, we report the performance based on the averaged evaluation scores across five folds with 300 epochs.

Baseline Methods and Configuration

In this work, the following baseline methods are compared: (1) **xQuAD** (Santos, Macdonald, and Ounis 2010) and **PM-2** (Dang and Croft 2012) are adopted to represent the typical unsupervised methods for SRD. (2) **DSSA** (Jiang et al. 2017) and **DVGAN** (Liu et al. 2020a) represent the supervised methods that incorporate explicit subtopic features. (3) **SVM-DIV** (Yue and Joachims 2008), **R-LTR** (Zhu et al. 2014), **PAMM** (Xia et al. 2015), **NTN-DIV** (Xia et al. 2016), **MDP-DIV** (Xia et al. 2017), **M^{2}DIV** (Feng et al. 2018), **PPG** (Xu et al. 2020) and **Graph4DIV** (Su et al. 2021) represent the supervised methods that do not use explicit subtopic features. (4) **DESA** (Qin, Dou, and Wen 2020) and **DALETOR** (Yan et al. 2021) represent the state-of-the-art supervised methods based on the score-and-sort strategy. Different from DALETOR, explicit subtopics are used by DESA. By following DALETOR (Yan et al. 2021), we choose the optimizer of Adagrad, and set the dimensions of fully connected layers before either probabilistic or deterministic regression as $[256, 128, 64]$. The other hyper-parameters are chosen via a grid search: number of attention heads $\in \{2, 4, 6\}$, number of self-attention layers $\in \{2, 4, 6\}$, learning rate $\in \{0.001, 0.01\}$, activation functions $\in \{ReLU, GELU\}$, global variance $\sigma \in \{0.1, 1, 10\}$ and number of Gaussian components $V \in \{1, 10\}$. Using a consistent network setting enables us to conduct a fair comparison between DALETOR and MO4SRD. Furthermore, we investigate the following variants of MO4SRD: MO4SRD(ERR-IA): using ERR-IA as the optimization objective. MO4SRD(α-DCG): using α-DCG as the optimization objective, MO4SRD($\sigma = b$): using a global variance setting for all documents. As far as we know, the effectiveness of optimizing ERR-IA has not been investigated by prior studies.

Experimental Evaluation

Noteworthy, the previous studies use different cutoff values to compute metric performance when comparing different models. For example, the models, like DSSA, DVGAN, DESA and Graph4DIV, mainly use a cutoff value of 20 (i.e., α-nDCG@20 and ERR-IA@20), while the methods, such as DALETOR and PPG, mainly use cutoff values of 5 and 10. In order to obtain an in-depth comparison of the typical methods for SRD, we use cutoff values of 5, 10 and 20[5]. Table 1 shows the overall performance of the involved approaches, respectively. The best result is indicated in bold, and the second-best result is underlined. † denotes that the results in terms of α-nDCG@5, α-nDCG@10, ERR-IA@5 and ERR-IA@10 are cited from Yan et al. (2021). $\star$ denotes that the results in terms of α-nDCG@20 and ERR-IA@20 are cited from Su et al. (2021).

From Table 1, we can observe that: (1) DALETOR and MO4SRD significantly outperforms the previous approaches for SRD, including the newly proposed models DVGAN (Liu et al. 2020a), PPG (Xu et al. 2020), Graph4DIV (Su et al. 2021) and DESA (Qin, Dou, and Wen 2020). The key reason is that DALETOR and MO4SRD directly use the evaluation metric as the optimization objective, which circumvents the mismatch between the objective function used in training and the final evaluation measure used in testing. (2) Due to the unavailability of the

[2]Here we compare time complexity by focusing on the way of generating the ranked list, the complexity of score estimation is not included since it varies due to the adopted network structures.

[3]Detailed implementation: https://github.com/wildltr/ptranking

[4]https://github.com/sweetalyssum/M2DIV

[5]The results are computed using the officially released script *ndeval* with the default settings.

Method	α-nDCG@5	α-nDCG@10	α-nDCG@20	ERR-IA@5	ERR-IA@10	ERR-IA@20
xQuAD(†,⋆)	0.3165	0.3941	0.413	0.2314	0.2890	0.317
PM-2(†,⋆)	0.3047	0.3730	0.411	0.2298	0.2814	0.306
DSSA(⋆)	-	-	0.456	-	-	0.356
DVGAN(⋆)	-	-	0.465	-	-	0.367
DESA(⋆)	-	-	0.464	-	-	0.363
Graph4DIV(⋆)	-	-	0.468	-	-	0.370
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	-
PPG	0.4799	0.5122	-	0.3727	0.3914	-
DALETOR(†)	**0.5009**	**0.5294**	-	**0.3942**	0.4119	-
DALETOR(reproduce)	0.4799	0.5084	0.5466	0.3789	0.3962	0.4067
MO4SRD(α-nDCG)	0.4738	0.5132	0.5509	0.3689	0.3905	0.4010
MO4SRD(ERR-IA)	0.4083	0.4606	0.5007	0.3129	0.3383	0.3495
MO4SRD(α-nDCG,$\sigma = 1.0$)	0.4930	0.5289$*$	**0.5656**$*$	0.3920	**0.4123**	**0.4225**
MO4SRD(ERR-IA,$\sigma = 1.0$)	0.4421	0.4798	0.5211	0.3483	0.3676	0.3790

Table 1: Performance comparison on TREC Web Track datasets, where $*$ indicates significant improvements over DALETOR(reproduce) with the Wilcoxon signed-rank test ($p < 0.05$) in terms of α-nDCG.

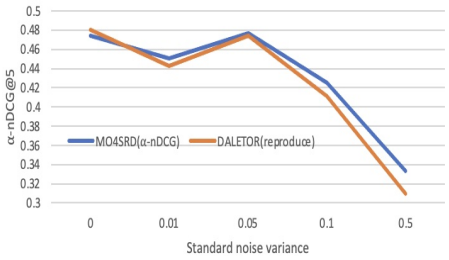

Figure 2: The impact of Gaussian noise on performance.

source code for DALETOR, we re-implemented it (denoted as *DALETOR(reproduce)*) on our own so as to make a fair comparison between DALETOR and MO4SRD such that they use the same network components except the regression part and the loss function. Unfortunately, we failed to achieve the same level of performance as reported in Yan et al. (2021). Compared with the re-implemented version, we can see that MO4SRD with learnable mean and variance can achieve competitive performance as DALETOR when we use α-DCG as the optimization objective. Moreover, when we use a uniform variance setting (i.e., $\sigma = 1.0$), MO4SRD significantly outperforms DALETOR in terms of α-nDCG. (3) A closer look the performance of MO4SRD's different variants reveals that using different metrics as the optimization objectives significantly affects the performance. Our results show that using α-DCG as the optimization objective is a better choice. The most possible reason is due to the underlying differences in rewarding aspect coverage and penalizing redundancy.

To show the importance of coping with the inherent uncertainty when performing SRD, Figure 2 illustrates the impact of adding Gaussian noise on performance. In particular, we add Gaussian noise to the embedding vectors of queries and documents before feeding them into each model, namely $\mathbf{x} = \mathbf{x} + \mathcal{N}(0, \sigma^2\mathbf{I})$. From Table 1 and Figure 2, we can observe that: though DALETOR(reproduce) performs slightly better than MO4SRD(α-nDCG) with no noise, its performance is significantly impacted when we increase the Gaussian noise. On the contrary, MO4SRD(α-nDCG) is less impacted and shows better performance than DALETOR(reproduce).

Conclusions

We proposed a novel method MO4SRD for SRD based on probabilistic regression. On one hand, MO4SRD represents the diversity score of each document using a probability distribution, which enables us to cope with the inherent uncertainty during the learning process of a model and the retrieval process. Further, this probabilistic formulation enables us to get the differentiable variants of the widely used diversity metrics and directly using them as the optimization objective. On the other hand, MO4SRD jointly scores candidate documents by taking into account both cross-document interaction and permutation equivariance, which makes it possible to generate a diversified ranking via a simple sorting. Compared to the state-of-the-art methods, MO4SRD achieves significantly improved performance. Our work also opens up many interesting future research directions. First, we have only demonstrated the effectiveness of probabilistic regression for SRD. It should be very interesting to evaluate how calibrated the proposed probabilistic regression is. Second, we plan to investigate the generalization ability of the proposed method by adapting it to other research topics, such as document summarization and paraphrase generation, where diversification plays an important role.

Acknowledgements

We sincerely thank all the anonymous reviewers for their helpful comments. This work was supported by JSPS KAKENHI Grant Number 19H04215.

References

Beutel, A.; Covington, P.; Jain, S.; Xu, C.; Li, J.; Gatto, V.; and Chi, E. H. 2018. Latent cross: Making use of context in recurrent recommender systems. *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 46–54.

Bishop, C. M. 1994. Mixture Density Networks. Technical report.

Carbonell, J.; and Goldstein, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st SIGIR*, 335–336.

Chapelle, O.; Metlzer, D.; Zhang, Y.; and Grinspan, P. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th CIKM*, 621–630.

Clarke, C. L. A.; Kolla, M.; Cormack, G. V.; Vechtomova, O.; Ashkan, A.; Büttcher, S.; and MacKinnon, I. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st SIGIR*, 659–666.

Dang, V.; and Croft, W. B. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th SIGIR*, 65–74.

Dang, V.; and Croft, W. B. 2013. Term level search result diversification. In *Proceedings of the 36th SIGIR*, 603–612.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, 4171–4186.

Ding, Q.; Liu, Y.; Miao, C.; Cheng, F.; and Tang, H. 2021. A Hybrid Bandit Framework for Diversified Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4036–4044.

Feng, Y.; Xu, J.; Lan, Y.; Guo, J.; Zeng, W.; and Cheng, X. 2018. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. In *Proceedings of SIGIR*, 125–134.

Gollapudi, S.; and Sharma, A. 2009. An Axiomatic Approach for Result Diversification. In *Proceedings of the 18th WWW*, 381–390.

Guo, S.; and Sanner, S. 2010. Probabilistic latent maximal marginal relevance. In *Proceedings of the 33rd SIGIR*, 833–834.

Hoare, C. A. R. 1962. Quicksort. *The Computer Journal*, 5(1): 10–16.

Hu, S.; Dou, Z.; Wang, X.; Sakai, T.; and Wen, J.-R. 2015. Search Result Diversification Based on Hierarchical Intents. In *Proceedings of the 24th CIKM*, 63–72.

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

Jiang, Z.; Wen, J.-R.; Dou, Z.; Zhao, W. X.; Nie, J.-Y.; and Yue, M. 2017. Learning to Diversify Search Results via Subtopic Attention. In *Proceedings of the 40th SIGIR*, 545–554.

Kim, S.-J.; and Lee, J.-H. 2015. Subtopic mining using simple patterns and hierarchical structure of subtopic candidates from web documents. *Information Processing \& Management*, 51(6): 773–785.

Le, Q.; and Mikolov, T. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st ICML*, 1188–1196.

Liang, S. 2019. Collaborative, Dynamic and Diversified User Profiling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4269–4276.

Liu, J.; Dou, Z.; Wang, X.; Lu, S.; and Wen, J.-R. 2020a. DVGAN: A Minimax Game for Search Result Diversification Combining Explicit and Implicit Features. In *Proceedings of SIGIR*, 479–488.

Liu, X.; Bouchoucha, A.; Sordoni, A.; and Nie, J.-Y. 2014. Compact Aspect Embedding for Diversified Query Expansions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 115–121.

Liu, Y.; Xiao, Y.; Wu, Q.; Miao, C.; Zhang, J.; Zhao, B.; and Tang, H. 2020b. Diversified Interactive Recommendation with Implicit Feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4932–4939.

Ma, H.; Lyu, M. R.; and King, I. 2010. Diversifying Query Suggestion Results. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1399–1404.

Pang, L.; Xu, J.; Ai, Q.; Lan, Y.; Cheng, X.; and Wen, J. 2020. SetRank: Learning a Permutation-Invariant Ranking Model for Information Retrieval. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 499–508.

Pasumarthi, R. K.; Zhuang, H.; Wang, X.; Bendersky, M.; and Najork, M. 2020. Permutation Equivariant Document Interaction Network for Neural Learning to Rank. *Proceedings of the 2020 ACM SIGIR International Conference on Theory of Information Retrieval*, 145–148.

Qin, T.; Liu, T.-Y.; and Li, H. 2010. A general approximation framework for direct optimization of information retrieval measures. *Journal of Information Retrieval*, 13(4): 375–397.

Qin, X.; Dou, Z.; and Wen, J. R. 2020. Diversifying Search Results using Self-Attention Network. In *International Conference on Information and Knowledge Management, Proceedings*, 1265–1274.

Qin, Z.; Yan, L.; Zhuang, H.; Tay, Y.; Pasumarthi, R. K.; Wang, X.; Bendersky, M.; and Najork, M. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees? In *Proceedings of ICLR*.

Radlinski, F.; Kleinberg, R.; and Joachims, T. 2008. Learning Diverse Rankings with Multi-armed Bandits. In *Proceedings of the 25th ICML*, 784–791.

Raiber, F.; and Kurland, O. 2013. Ranking document clusters using markov random fields. In *Proceedings of the 36th SIGIR*, 333–342.

Rijsbergen, C. J. V. 1979. *Information Retrieval*. 2nd edition.

Sanner, S.; Guo, S.; Graepel, T.; Kharazmi, S.; and Karimi, S. 2011. Diverse retrieval via greedy optimization of expected 1-call@k in a latent subtopic relevance model. In *Proceedings of the 20th CIKM*, 1977–1980.

Santos, R. L. T.; Macdonald, C.; and Ounis, I. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th WWW*, 881–890.

Santos, R. L. T.; Macdonald, C.; and Ounis, I. 2015. Search Result Diversification. *Foundations and Trends in Information Retrieval*, 9(1): 1–90.

Sarwar, S. M.; Addanki, R.; Montazeralghaem, A.; Pal, S.; and Allan, J. 2020. Search Result Diversification with Guarantee of Topic Proportionality. In *Proceedings of the 2020 ACM SIGIR International Conference on Theory of Information Retrieval*, 53–60.

Su, Z.; Dou, Z.; Zhu, Y.; Qin, X.; and Wen, J.-r. 2021. Modeling Intent Graph for Search Result Diversification. In *Proceedings of the 44th SIGIR*. Association for Computing Machinery.

Taylor, M.; Guiver, J.; Robertson, S.; and Minka, T. 2008. SoftRank: Optimizing Non-smooth Rank Metrics. In *Proceedings of the 1st WSDM*, 77–86.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, U.; and Polosukhin, I. 2017. Attention Is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Xia, L.; Xu, J.; Lan, Y.; Guo, J.; and Cheng, X. 2015. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. In *Proceedings of the 38th SIGIR*, 113–122.

Xia, L.; Xu, J.; Lan, Y.; Guo, J.; and Cheng, X. 2016. Modeling Document Novelty with Neural Tensor Network for Search Result Diversification. In *Proceedings of the 39th SIGIR*, 395–404.

Xia, L.; Xu, J.; Lan, Y.; Guo, J.; Zeng, W.; and Cheng, X. 2017. Adapting Markov Decision Process for Search Result Diversification. In *Proceedings of the 40th SIGIR*, 535–544.

Xu, J.; Wei, Z.; Xia, L.; Lan, Y.; Yin, D.; Cheng, X.; and Wen, J.-R. 2020. Reinforcement Learning to Rank with Pairwise Policy Gradient. In *Proceedings of SIGIR*, 509–518.

Xu, J.; Xia, L.; Lan, Y.; Guo, J.; and Cheng, X. 2017. Directly optimize diversity evaluation measures: A new approach to search result diversification. *ACM Transactions on Intelligent Systems and Technology*, 8(3).

Yan, L.; Qin, Z.; Kumar Pasumarthi, R.; Wang, X.; and Bendersky, M. 2021. Diversification-Aware Learning to Rank using Distributed Representation. In *Proceedings of the Web Conference 2021*, 127–136.

Yigit-Sert, S.; Altingovde, I. S.; Macdonald, C.; Ounis, I.; and Ulusoy, Ö. 2020. Supervised approaches for explicit search result diversification. *Information Processing and Management*, 57(6): 102356.

Yu, H.-T.; Jatowt, A.; Blanco, R.; Joho, H.; Jose, J.; Chen, L.; and Yuan, F. 2017. A Concise Integer Linear Programming Formulation for Implicit Search Result Diversification. In *Proceedings of the 10th WSDM*, 191–200.

Yu, H.-T.; Jatowt, A.; Blanco, R.; Joho, H.; Jose, J.; Chen, L.; and Yuan, F. 2018. Revisiting the cluster-based paradigm for implicit search result diversification. *Information Processing and Management*, 54(4): 507–528.

Yu, H.-T.; and Ren, F. 2014. Search Result Diversification via Filling up Multiple Knapsacks. In *Proceedings of the 23rd CIKM*, 609–618.

Yue, Y.; and Joachims, T. 2008. Predicting Diverse Subsets Using Structural SVMs. In *Proceedings of the 25th ICML*, 1224–1231.

Zhu, Y.; Lan, Y.; Guo, J.; Cheng, X.; and Niu, S. 2014. Learning for Search Result Diversification. In *Proceedings of the 37th SIGIR*, 293–302.

Zuccon, G.; Azzopardi, L.; Zhang, D.; and Wang, J. 2012. Top-k retrieval using facility location analysis. In *Proceedings of the 34th ECIR*, 305–316.