Learning for Search Result Diversification

Yadong Zhu Yanyan Lan Jiafeng Guo Xueqi Cheng Shuzi Niu
Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
{zhuyadong, niushuzi}@software.ict.ac.cn
{lanyanyan, guojiafeng, cxq}@ict.ac.cn

ABSTRACT

Search result diversification has gained attention as a way to tackle the ambiguous or multi-faceted information needs of users. Most existing methods on this problem utilize a heuristic predefined ranking function, where limited features can be incorporated and extensive tuning is required for different settings. In this paper, we address search result diversification as a learning problem, and introduce a novel relational learning-to-rank approach to formulate the task. However, the definitions of ranking function and loss function for the diversification problem are challenging. In our work, we firstly show that diverse ranking is in general a sequential selection process from both empirical and theoretical aspects. On this basis, we define ranking function as the combination of relevance score and diversity score between the current document and those previously selected, and loss function as the likelihood loss of ground truth based on Plackett-Luce model, which can naturally model the sequential generation of a diverse ranking list. Stochastic gradient descent is then employed to conduct the unconstrained optimization, and the prediction of a diverse ranking list is provided by a sequential selection process based on the learned ranking function. The experimental results on the public TREC datasets demonstrate the effectiveness and robustness of our approach.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval – *Retrieval Models*

General Terms

Algorithms, Experimentation, Performance, Theory

Keywords

Diversity, Relational Learning-to-Rank, Sequential Selection, Plackett-Luce Model

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609634 .

1. INTRODUCTION

Most users leverage Web search engine as a predominant tool to fulfill their information needs. Users' information needs, typically described by keyword based queries, are often ambiguous or multi-faceted. On the one hand, for some ambiguous queries, there are multiple interpretations of the underlying needs (e.g., query "band" may refer to the rock band, frequency band or rubber band). On the other hand, queries even with clear definition might still be multi-faceted (e.g., "britney spears"), in the sense that there are many aspects of the information needs (e.g., news, videos, photos of britney spears). Therefore, search result diversification has attracted considerable attention as a means to tackle the above problem [1]. The key idea is to provide a diversified result list, in the hope that different users will find some results that can cover their information needs.

Different methods on search result diversification have been proposed in literature, which are mainly non-learning methods, and can be divided into two categories: implicit methods and explicit methods. Implicit methods [3] assume that similar documents cover similar aspects, and rely on inter-document similarity for selecting diverse documents. While explicit methods [29] directly model the aspects of user queries and select documents that cover different aspects for diversification. However, most existing methods utilize a heuristic predefined utility function, and thus limited features can be incorporated and extensive tuning is required for different retrieval settings.

In this paper, we address search result diversification as a learning problem where a ranking function is learned for diverse ranking. Different from traditional relevance ranking based on the assumption of independent document relevance [17], diverse ranking typically considers the relevance of a document in light of the other retrieved documents [29]. Therefore, we introduce a novel *Relational Learning-to-Rank* framework (R-LTR for short) to formulate the task of search result diversification. R-LTR considers the interrelationships between documents in the ranking process, besides the content information of individual documents used in traditional learning-to-rank framework. However, the definitions of ranking function and loss function for the diversification problem are challenging.

From the top-down user browsing behavior and the ubiquitous greedy approximation for diverse ranking, we find that search result diversification is in general a sequential ranking process. Therefore, we propose to define the ranking function and loss function in a sequential way: (1) The ranking function is defined as the combination of relevance

score and diversity score, where the relevance score only depends on the content of the document, and the diversity score depends on the relationship between the current document and those previously selected. We describe different ways to represent the diversity score. (2) The loss function is defined as the likelihood loss of ground truth based on Plackett-Luce model [18], which can naturally model the sequential generation of a diverse ranking list. On this basis, stochastic gradient descent is employed to conduct the unconstrained optimization, and the prediction of diverse ranking list is provided by a sequential selection process based on the learned ranking function.

To evaluate the effectiveness of the proposed approach, we conduct extensive experiments on the public TREC datasets. The experimental results show that our methods can significantly outperform the state-of-the-art diversification approaches, with Official Diversity Metrics (ODM for short) of TREC diversity task including $ERR\text{-}IA$[1, 6], $\alpha\text{-}NDCG$[11] and $NRBP$[12]. Furthermore, our methods also achieve best in the evaluations of traditional intent-aware measures such as Precision-IA [1] and Subtopic Recall [37]. In addition, we give some discussions on the robustness of our methods and the importance of the proposed diversity features. Finally, we also study the efficiency of our approach based on the analysis of running time.

The main contributions of this paper lie in:

1. the proposal of a novel R-LTR framework to formulate search result diversification as a learning problem, where both content information and relationship among documents are considered;

2. the new definitions of ranking function and loss function based on the foundation of sequential selection process for diverse ranking;

3. an empirical verification of the effectiveness of the proposed approach based on public datasets.

The rest of the paper is organized as follows. We first review some related work in Section 2. We then introduce the R-LTR framework in Section 3, and describe the specific definitions of ranking function and loss function, learning and prediction procedures in Section 4. Section 5 presents the experimental results. Section 6 concludes the paper.

2. RELATED WORK

Most existing diversification methods are non-learning methods, which can be mainly divided into two categories: implicit approaches and explicit approaches.

The implicit methods assume that similar documents cover similar aspects and model inter-document dependencies. For example, Maximal Marginal Relevance (MMR) method [3] proposes to iteratively select a candidate document with the highest similarity to the user query and the lowest similarity to the already selected documents, in order to promote novelty. In fact, most of the existing approaches are somehow inspired by the MMR method. Zhai et al. [37] select documents with high divergence from one language model to another based on the risk minimization consideration.

The explicit methods explicitly model aspects of a query and then select documents that cover different aspects. The aspects of a user query can be achieved with a taxonomy [1, 32], top retrieved documents [5], query reformulations [24, 29], or multiple external resources [15]. Overall, the explicit methods have shown better experimental performances comparing with implicit methods.

There are also some other methods which attempt to borrow theories from economical or political domains. The work in [26, 33] applies economical portfolio theory for search result ranking, which views search diversification as a means of risk minimization. The approach in [13] treats the problem of finding a diverse search result as finding a proportional representation for the document ranking, which is like a critical part of most electoral processes.

The authors of [2, 27] try to construct a dynamic ranked-retrieval model, while our paper focuses on the common static ranking scenario. There are also some on-line learning methods that try to learn retrieval models by exploiting users' online feedback [25, 31, 35, 30, 28]. These research work can tackle diversity problem to some extent, but they focus on an 'on-line' or 'coactive' scenario, which is different from our work (i.e. offline supervised learning scenario).

Recently, some researchers have proposed to utilize machine learning techniques to solve the diversification problem. Yue et al. [36] propose to optimize subtopic coverage as the loss function, and formulate a discriminant function based on maximizing word coverage. However, their work only focuses on diversity, and discards the requirements of relevance. They claim that modeling both relevance and diversity simultaneously is a more challenging problem, which is exactly what we try to tackle in this paper. In this paper, we propose a novel R-LTR approach to conduct search result diversification, which is different from traditional approaches and shows promising experimental performance.

3. RELATIONAL LEARNING-TO-RANK

Traditional relevance ranking has been well formulated as a learning-to-rank (LTR for short) problem [17], where a ranking function is defined on the content of each individual document and learned toward some loss functions. However, in diverse ranking scenario, the overall relevance of a document ranking for a given query, should depend not only on the individual ranked documents, but also on how they related to each other [29]. Therefore, in this paper, we introduce a novel R-LTR framework to formulate the diverse ranking problem. The difference between LTR and R-LTR is that the latter considers both contents of individual document and relations between documents. In the following paper, we use superscript to denote the id of a query and subscript to denote the id of a document.

Formally, let $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_i$ denotes the d dimensional feature vector of a candidate document x_i for query q; Let $R \in \mathcal{R}^{n \times n \times l}$ denote a 3-way *tensor* representing relationships between the n documents, where R_{ijk} stands for the k-th feature of relation between documents x_i and x_j. Let $\mathbf{y}$ be a ground-truth of the query q, in the form of a vector of ranking scores or a ranking list. Supposing that $\mathbf{f}(X, R)$ is a ranking function, and the goal of R-LTR is to output the best ranking function from a function space $\mathcal{F}$.

In training procedure, given the labeled data with N queries as: $(X^{(1)}, R^{(1)}, \mathbf{y}^{(1)}), (X^{(2)}, R^{(2)}, \mathbf{y}^{(2)}), \cdots, (X^{(N)}, R^{(N)}, \mathbf{y}^{(N)})$. A loss function L is defined, and the learning process is conducted by minimizing the total loss with respect to the given

training data.

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}\in\mathcal{F}} \sum_{i=1}^{N} L(\mathbf{f}(X^{(i)}, R^{(i)}), \mathbf{y}^{(i)}). \quad (1)$$

In prediction, given $X^{(t)}$ and $R^{(t)}$ of n_t documents for query q_t, we output $\hat{\mathbf{y}}^{(t)}$ based on the learned ranking function $\hat{\mathbf{f}}(X^{(t)}, R^{(t)})$.

In fact, the proposed R-LTR framework is very general, in the sense that many traditional ranking problems are its special cases.

(1) It is obvious to see that the conventional LTR framework is a special case of R-LTR. Specifically, if we ignore the relation tensor R, then we get the same function as that in traditional LTR, i.e. $\mathbf{f}(X, R) = \mathbf{f}(X)$.

(2) The 'learning to rank relational objects' framework [22, 23] is also a special case of R-LTR. Specifically, if we restrict the relation *tensor* R to be a *matrix*, with R_{ij} representing the relation between document x_i and x_j, then we get the same function as that in the problem of learning to rank relational objects.

The above framework gives a formulation of ranking problems involving relationship. When solving the specific problem, one needs to define the corresponding ranking function and loss function according to the task.

4. SEARCH RESULT DIVERSIFICATION VIA R-LTR FRAMEWORK

As mentioned in the previous section, it is natural to formulate search result diversification under R-LTR framework. In this paper, we mainly focus on the diverse ranking scenario. To apply the above framework to this specific task, the most challenging problem is the definition of ranking function and loss function.

4.1 Motivation

In order to properly define the ranking function and loss function, we first look into the diverse ranking problem.

(1) Empirically, users usually browse the Web search results in a top-down manner, and perceive diverse information from each individual document based on what he/she have obtained in the preceding results [8].

(2) Theoretically, diverse ranking can be naturally stated as a bi-criterion optimization problem, and it is NP-hard [1, 4]. Therefore, in practice, most previous approaches on search result diversification are based on greedy approximation, which sequentially select a 'local-best' document from the remanent candidate set [29].

From both empirical and theoretical analysis above, we can see that it is better to view diverse ranking as a sequential selection process, in the sense that the ranking list is generated in a sequential order, with each individual document ranked according to its relevance to the query and the relation between all the documents ranked before it.

4.2 Definition of Ranking Function

As discussed above, diverse ranking is in general a sequential selection process, where each individual document is ranked according to its relevance to the query and the relation between all the documents ranked before it. The intuitive idea is illustrated in Figure 1, when ranking documents in $X\backslash S$ given the already ranked results S, both content-based relevance and diversity relation between this

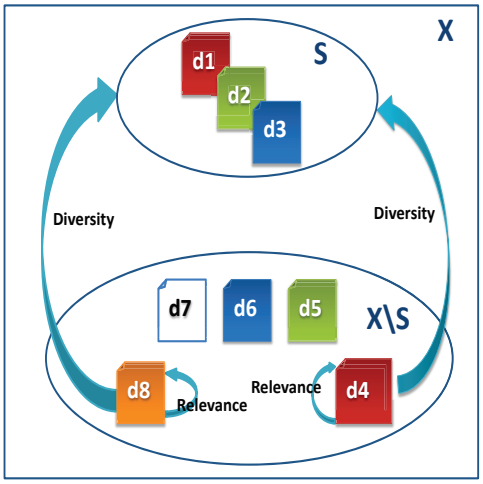

Figure 1: An illustration of the sequential way to define ranking function. All the rectangles represent candidate documents of a user query, and different colors represent different subtopics. The solid rectangle is relevant to the query, and the hollow rectangle is irrelevant to the query, and larger size means more relevance. X denotes all the candidate document collection. S denotes previously selected documents, and $X\backslash S$ denotes the remanent documents.

document and the previously selected documents in S should be considered. Noting that larger size of the rectangle means the document is more relevant to the query, and different colors represent different subtopics. Therefore, the document 8 may be more preferred than document 4 given S, since it is relevant to the query, and also provides different aspects additionally comparing with the selected set S.

Based on this ranking process, here we give the precise definition of ranking function. Given a query q, we assume that a set of documents have been selected, denoted as S, the scoring function on the candidate document in $X\backslash S$, is then defined as the combination of the relevance score and the diversity score between the current document and those previously selected, shown as follows.

$$f_S(x_i, R_i) = \omega_r^T \mathbf{x}_i + \omega_d^T h_S(R_i), \forall x_i \in X\backslash S, \quad (2)$$

where $\mathbf{x}_i$ denotes the relevance feature vector of the candidate document x_i, R_i stands for the *matrix* of relationships between document x_i and other selected documents, with each R_{ij} stands for the relationship vector between document x_i and x_j, represented by the *feature vector* of $(R_{ij1}, \cdots, R_{ijl})$, $x_j \in S$, and R_{ijk} stands for the k-th *relation feature* between documents x_i and x_j. $h_S(R_i)$ stands for the *relational function* on R_i, ω_r^T and ω_d^T stands for the corresponding relevance and diversity weight vector. When $S = \emptyset$, $f_S(x_i, R_i)$ is directly represented as $\omega_r^T \mathbf{x}_i$. Then the ranking function can be represented as the set of scoring function:

$$\mathbf{f}(X, R) = (f_{S_\emptyset}, f_{S_1}, \cdots, f_{S_{n-1}})$$

where S_i, denotes the previously selected document collection with i documents. From the above definition, we can see that if we do not consider diversity relation, our ranking

function reduce to $\mathbf{f}(X) = (f(x_1), \cdots, f(x_n))$, which is the traditional ranking function in learning-to-rank.

4.2.1 Relational function $h_S(R_i)$

Please note that the relational function $h_S(R_i)$ denotes the way of representing the diversity relationship between the current document x_i and the previously selected documents in S. If we treat diversity relation as distance, $h_S(R_i)$ can be viewed as the distance of x_i to the set S. According to different definitions of the distance between an item and a set of items, $h_S(R_i)$ can be defined as the following three ways.

Minimal Distance. The distance between a document x_i and a set S is defined as the minimal distance of all the document pairs $(x_i, x_j), x_j \in S$.

$$h_S(R_i) = (\min_{x_j \in S} R_{ij1}, \cdots, \min_{x_j \in S} R_{ijl}).$$

Average Distance. The distance between a document x_i and a set S is defined as the average distance of all the document pairs $(x_i, x_j), x_j \in S$.

$$h_S(R_i) = (\frac{1}{|S|}\sum_{x_j \in S} R_{ij1}, \cdots, \frac{1}{|S|}\sum_{x_j \in S} R_{ijl}).$$

Maximal Distance. The distance between a document x_i and a set S is defined as the maximal distance of all the document pairs $(x_i, x_j), x_j \in S$.

$$h_S(R_i) = (\max_{x_j \in S} R_{ij1}, \cdots, \max_{x_j \in S} R_{ijl}).$$

4.2.2 Diversity Feature Vector R_{ij}

How to define discriminative features that can well capture diversity relation is critical for the success of R-LTR. In this work, we provides several representative features for the learning process, including semantic diversity features (i.e. subtopic diversity, text diversity, title diversity, anchor text diversity and ODP-based diversity) and structural diversity features (i.e. link-based diversity and url-based diversity).

Subtopic Diversity. Different documents may associate with different aspects of the given topic. We use Probabilistic Latent Semantic Analysis (PLSA) [16] to model implicit subtopics distribution of candidate objects, which is important for the diversification task as mentioned before. Therefore, we define the diversity feature based on implicit subtopics as follows.

$$R_{ij1} = \sqrt{\sum_{k=1}^{m}(p(z_k|x_i) - p(z_k|x_j))^2}$$

Text Diversity. Text dissimilarity is also meaningful for diversity. We propose to represent it as the cosine dissimilarity based on weighted term vector representations, and define the feature as follows.

$$R_{ij2} = 1 - \frac{\mathbf{d}_i \cdot \mathbf{d}_j}{\|\mathbf{d}_i\|\|\mathbf{d}_j\|},$$

where $\mathbf{d}_i, \mathbf{d}_j$ are the weighted document vectors based on $tf*idf$, and tf denotes the term frequencies, idf denotes inverse document frequencies. There also exists other computing ways such as the work in [14], which is based on sketching algorithm and Jaccard similarity.

Title Diversity. The way of computing title diversity feature is similar as that for text diversity feature, which is denoted as R_{ij3}.

Anchor Text Diversity. The anchor text can accurately describe the content of corresponding page and is important. This type of feature is computed similarly as text and title diversity features, denoted as R_{ij4}.

ODP-Based Diversity. The existing ODP taxonomy[1] offers a succinct encoding of distances between documents. Usually, the distance between documents on similar topics in the taxonomy is likely to be small. For two categories u and v, we define the categorical distance between them as following:

$$c\_dis(u, v) = 1 - \frac{|l(u, v)|}{\max\{|u|, |v|\}}$$

where $l(u, v)$ is the length of their longest common prefix. $|u|$ and $|v|$ is the length of category u and v. Then given two documents x_i and x_j and their category information sets $\mathcal{C}_i$ and $\mathcal{C}_j$ respectively, we define the ODP-based diversity feature as:

$$R_{ij5} = \frac{\sum_{u \in \mathcal{C}_i}\sum_{v \in \mathcal{C}_j} c\_dis(u, v)}{|\mathcal{C}_i| \cdot |\mathcal{C}_j|}$$

where $|\mathcal{C}_i|$ and $|\mathcal{C}_j|$ are the number of categories in corresponding category sets.

Link-Based Diversity. By constructing a web link graph, we can calculate the link similarity of any document pair based on direct inlink or outlink information. The link-based diversity feature is then defined as follows.

$$R_{ij6} = \begin{cases} 0 & \text{if } x_i \in inlink(x_j) \cup outlink(x_j), \\ 1 & \text{otherwise} \end{cases}$$

URL-Based Diversity. Given the url information of two documents, we can judge whether they belong to the same domain or the same site. The url-based diversity feature is then defined as follows.

$$R_{ij7} = \begin{cases} 0 & \text{if one url is another's } \textit{prefix} \\ 0.5 & \text{if they belong to the same } \textit{site} \text{ or } \textit{domain} \\ 1 & \text{otherwise} \end{cases}$$

Based on these diversity features, we can obtain the diversity feature vector $R_{ij} = (R_{ij1}, R_{ij2}, \cdots, R_{ij7})$. All the feature values are normalized to the range of [0,1]. Please note that there might be some other useful resources for the definition of diversity features, e.g., clickthrough logs, which will be further considered in our future work.

4.3 Definition of Loss Function

Motivated by the analysis that the process for diverse ranking is in general a sequential selection process, we propose to model the generation of a diverse ranking list in a sequential way, and define the loss function as the *likelihood loss of the generation probability*.

$$L(\mathbf{f}(X, R), \mathbf{y}) = -\log P(\mathbf{y}|X). \quad (3)$$

Intuitively, the generation probability of a ranking list can be viewed as a process to iteratively select the top ranked

[1] http://www.dmoz.org/

documents from the remaining documents. The precise definition is given as follows.

$$P(\mathbf{y}|X) = P(x_{y(1)}, x_{y(2)}, \cdots, x_{y(n)}|X) \quad (4)$$
$$= P(x_{y(1)}|X)P(x_{y(2)}|X\backslash S_1)\cdots P(x_{y(n-1)}|X\backslash S_{n-2}),$$

where $y(i)$ stands for the index of document which is ranked in position i in the ranking list $\mathbf{y}$, X denotes all the candidate documents, $S_i = \{x_{y(1)}, \cdots, x_{y(i)}\}$, denotes the previously selected document collection with i documents, $P(x_{y(1)}|X)$ stands for the probability that $x_{y(1)}$ is ranked first among the documents in X, and $P(x_{y(j)}|X\backslash S_{j-1})$ stands for the probability that document $x_{y(j)}$ is ranked first among the documents in $X\backslash S_{j-1}$.

4.3.1 Plackett-Luce based Probability $P(\mathbf{y}|X)$

The above sequential definition approach can be well captured by the Plackett-Luce Model [18]. Therefore, we propose to define $P(x_{y(1)}|X)$ and $P(x_{y(j)}|X\backslash S_{j-1})$ in a similar way, shown as follows, $j \geq 2$.

$$P(x_{y(1)}|X) = \frac{\exp\{f_\emptyset(x_{y(1)})\}}{\sum_{k=1}^{n}\exp\{f_\emptyset(x_{y(k)})\}}, \quad (5)$$

$$P(x_{y(j)}|X\backslash S_{j-1}) = \frac{\exp\{f_{S_{j-1}}(x_{y(j)}, R_{y(j)})\}}{\sum_{k=j}^{n}\exp\{f_{S_{k-1}}(x_{y(k)}, R_{y(k)})\}}. \quad (6)$$

Incorporating Eq.(5) and Eq.(6) into Eq.(4), the *generation probability* of a diverse ranking list is formulated as follows.

$$P(y|X) = \prod_{j=1}^{n}\frac{\exp\{f_{S_{j-1}}(x_{y(j)}, R_{y(j)})\}}{\sum_{k=j}^{n}\exp\{f_{S_{k-1}}(x_{y(k)}, R_{y(k)})\}}, \quad (7)$$

where $S_0 = \emptyset$, $f_\emptyset(x, R) = \omega_r^T\mathbf{x}$.

4.3.2 Relation to ListMLE in Learning-to-Rank

Incorporating Eq.(7) into the definition of the loss function Eq.(3), we can obtain the precise definition of the loss function as follows.

$$L(\mathbf{f}(X,R), \mathbf{y}) = -\sum_{j=1}^{n}\log\left\{\frac{\exp\{f_{S_{j-1}}(x_{y(j)}, R_{y(j)})\}}{\sum_{k=j}^{n}\exp\{f_{S_{k-1}}(x_{y(k)}, R_{y(k)})\}}\right\} \quad (8)$$

We can see that our loss function is similar to that in ListMLE [34], which is formulated as follows.

$$L(\mathbf{f}(X), y) = -\sum_{j=1}^{n}\log\left\{\frac{\exp\{f(x_{y(j)})\}}{\sum_{k=j}^{n}\exp\{f(x_{y(k)})\}}\right\},$$

where $f(x)$ is the score function in traditional learning-to-rank, i.e. $f(x) = \omega^T\mathbf{x}$.

Therefore, if we do not consider diversity relation in our framework, our loss function will reduce to the same form of that in ListMLE. That is to say, ListMLE is a special case of our loss function.

4.4 Learning and Prediction

Based on the definitions of ranking function and loss function, we present the learning and prediction process in this section. Specifically, we first describe how to construct the training data, and then introduce the optimization procedure. Finally, we show how to make predictions based on the learned ranking function.

Algorithm 1 Construction of Approximate Ideal Ranking List

Input:
$(q_i, X^{(i)}, \mathbf{T}_i, P(x_j^{(i)}|t)), t \in \mathbf{T}_i, x_j^{(i)} \in X^{(i)}$
Output: $\mathbf{y}^{(i)}$
1: Initialize $S_0 \leftarrow \emptyset, \mathbf{y}^{(i)} = (1, \cdots, n_i)$
2: **for** $k = 1, ..., n_i$ **do**
3: bestDoc $\leftarrow \text{argmax}_{x\in X^{(i)}\backslash S_{k-1}} ODM(S_{k-1} \cup x)$
4: $S_k \leftarrow S_{k-1} \cup$ bestDoc
5: $y^{(i)}(k) =$ the *index* of bestDoc
6: **end for**
7: **return** $\mathbf{y}^{(i)} = (y^{(i)}(1), \cdots, y^{(i)}(n_i))$.

Algorithm 2 Optimization Algorithm

Input: training data $\{(X^{(i)}, R^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$,
parameter: learning rate η, tolerance rate ϵ
Output: model vector: ω_r, ω_d
1: Initialize parameter value ω_r, ω_d
2: **repeat**
3: Shuffle the training data
4: **for** $i = 1, ..., N$ **do**
5: Compute gradient $\Delta{\omega_r}^{(i)}$ and $\Delta{\omega_d}^{(i)}$
6: Update model: $\omega_r = \omega_r - \eta \times \Delta{\omega_r}^{(i)}$,
$\omega_d = \omega_d - \eta \times \Delta{\omega_d}^{(i)}$
7: **end for**
8: Calculate likelihood loss on the training set
9: **until** the change of likelihood loss is below ϵ

4.4.1 Training Data

The labeled data in search result diversification such as TREC diversity task are usually provided in the form of $(q_i, X^{(i)}, \mathbf{T}_i, P(x_j^{(i)}|t)), t \in \mathbf{T}_i, x_j^{(i)} \in X^{(i)}$, where $X^{(i)}$ is a candidate document set of query q_i, $\mathbf{T}_i$ is the subtopics of query q_i, t is a specific subtopic in $\mathbf{T}_i$, and $P(x_j^{(i)}|t)$ describes the relevance of document $x_j^{(i)}$ to subtopic t. We can see that the above form of labeled data deviates the formulation of $\mathbf{y}^{(i)}$ in our R-LTR framework, which requires a *ranking list* of candidate documents. In order to apply R-LTR, we need to construct $\mathbf{y}^{(i)}$ from the provided form of labeled data.

We propose to construct an approximate ideal ranking list by maximizing the ODM measures (e.g., $ERR\text{-}IA$), and use the approximate ideal ranking list as the training ground-truth $\mathbf{y}^{(i)}$ for query q_i, as described in Algorithm 1.

According to the results in [20], if a submodular function is monotonic (i.e., $f(S) \leq f(T)$, whenever $S \subseteq T$) and normalized (i.e., $f(\emptyset) = 0$), greedily constructing gives an $(1 - 1/e)$-approximation to the optimal. Since any member of ODM is a submodular function, we can easily prove that Algorithm 1 is $(1 - 1/e)$-approximation to the optimal (We omit the proof here). And the quality of training ground-truth can be guaranteed.

4.4.2 Learning

Given the training data $\{(X^{(i)}, R^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$, the total loss is represented as follows.

$$-\sum_{i=1}^{N}\sum_{j=1}^{n_i}\log\left\{\frac{\exp\{\omega_r^T\mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\sum_{k=j}^{n_i}\exp\{\omega_r^T\mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}}\right\} \quad (9)$$

Algorithm 3 Ranking Prediction via Sequential Selection

Input: $X^{(t)}, R^{(t)}, \omega_r, \omega_d$
Output: $\mathbf{y}^{(t)}$
1: Initialize $S_0 \leftarrow \emptyset, \mathbf{y}^{(t)} = (1, \cdots, n_t)$
2: **for** $k = 1, ..., n_t$ **do**
3: bestDoc $\leftarrow \text{argmax}_{x \in X_t} f_{S_{k-1}}(x, R)$
4: $S_k \leftarrow S_{k-1} \cup$ bestDoc
5: $y^{(t)}(k) \leftarrow$ the *index* of bestDoc
6: **end for**
7: **return** $\mathbf{y}^{(t)} = (y^{(t)}(1), \cdots, y^{(t)}(n_t))$

For such a unconstrained optimization problem, we employ Stochastic Gradient Descent (SGD) to conduct optimization as shown in Algorithm 2. According to Eq.(9), the gradient at training sample $X^{(i)}$ is computed as follows.

$$\Delta\omega_r^{(i)} = \sum_{j=1}^{n_i} \left\{ \frac{\sum_{k=j}^{n_i} \mathbf{x}_{y(k)}^{(i)} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}}{\sum_{k=j}^{n_i} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}} - \frac{\mathbf{x}_{y(j)}^{(i)} \exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}} \right\},$$

$$\Delta\omega_d^{(i)} = \sum_{j=1}^{n_i} \left\{ \frac{\sum_{k=j}^{n_i} h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)}) \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}}{\sum_{k=j}^{n_i} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}} - \frac{h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)}) \exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}} \right\}.$$

4.4.3 Prediction

As the ranking function is defined sequentially, traditional prediction approach (i.e., calculating the ranking score of each independent document simultaneously and sorting them in descending order to obtain a ranking list) fails in our framework. According to the sequential selection essence of diverse ranking, we propose a sequential prediction process, as described in Algorithm 3. Specifically, in the first step, the most relevant document with maximal relevance score will be selected and ranked first. If the top k items have been selected, then the document in position $k+1$ should be with maximum f_{S_k}. At last, all the documents are ranked accordingly, and we obtain the final ranking list.

Assuming that the size of output ranking is K, the size of candidate set is n, then this type of sequential selection algorithm 3 will have time complexity of $O(n * K)$. Usually, the original value of n is large, therefore, an initial retrieval can be applied to provide a filtered candidate set with relatively small size (e.g., top 1000 or 3000 retrieved documents). With a small K, the prediction time is linear.

5. EXPERIMENTS

In this section, we evaluate the effectiveness of our approach empirically. We first introduce the experimental setup. We then compare our approach with baseline methods under different diversity evaluation measures. Furthermore, we analyze the performance robustness of different diversity methods and the importance of our proposed diversity features. Finally, we study the efficiency of our approach based on the analysis of running time.

5.1 Experimental Setup

Here we give some introductions on the experimental setup, including data collections, evaluation metrics, baseline models and detailed implementation.

5.1.1 Data Collections

Our evaluation was conducted in the context of the diversity tasks of the TREC2009 Web Track (WT2009), TREC2010 Web Track (WT2010), and TREC2011 Web Track (WT2011), which contain 50, 48 and 50 test queries (or topics), respectively. Each topic includes several subtopics identified by TREC assessors, with binary relevance judgements provided at the subtopic level[2]. All the experiments were carried out on the ClueWeb09 Category B data collection[3], which comprises a total of 50 million English Web documents.

5.1.2 Evaluation Metrics

The current official evaluation metrics of the diversity task include $ERR\text{-}IA$ [6], $\alpha\text{-}NDCG$ [11] and $NRBP$ [12]. They measure the diversity of a result list by explicitly rewarding novelty and penalizing redundancy observed at every rank. We also use traditional diversity measures for evaluation: Precision-IA [1] and Subtopic Recall [37].They measure the precision across all subtopics of the query and the ratio of the subtopics covered in the results, respectively. All the measures are computed over the top-k search results ($k = 20$). Moreover, the associated parameters α and β are all set to be 0.5, which is consistent with the default settings in official TREC evaluation program.

5.1.3 Baseline Models

To evaluate the performance of our approach, we compare our approach with the state-of-the-art approaches, which are introduced as follows.

QL. The standard Query-likelihood language model is used for the initial retrieval, which provides the top 1000 retrieved documents as a candidate set for all the diversification approaches. It is also used as a basic baseline method in our experiment.

MMR. MMR is a classical implicit diversity method in the diversity research. It employs a linear combination of relevance and diversity as the metric called "marginal relevance" [3]. MMR will iteratively select document with the largest "marginal relevance".

xQuAD. The explicit diversification approaches are popular in current research field, in which xQuAD is the most representative and used as a baseline model in our experiments [29].

PM-2. PM-2 is also a explicit method that proposes to optimize proportionality for search result diversification [13]. It has been proved to achieve promising performance in their work, and is also chosen as baseline in our experiment.

[2]For WT2011 task, assessors made graded judgements. While in the official TREC evaluation program, it mapped these graded judgements to binary judgements by treating $values > 0$ as relevant and $values \leq 0$ as not relevant.
[3]http://boston.lti.cs.cmu.edu/Data/clueweb09/

Table 1: Relevance Features for learning on ClueWeb09-B collection [21, 19].

Category	Feature Description	Total
Q-D	TF-IDF	5
Q-D	BM25	5
Q-D	QL.DIR	5
Q-D	MRF	10
D	PageRank	1
D	#Inlinks	1
D	#Outlinks	1

ListMLE. ListMLE is a plain learning-to-rank approach without diversification considerations, and is a representative listwise relevance approach in LTR field [17].

SVMDIV. SVMDIV is a representative supervised approach for search result diversification [36]. It proposes to optimize subtopic coverage by maximizing word coverage. It formulates the learning problem and derives a training method based on structural SVMs. However, SVMDIV only models diversity and discards the requirement of relevance. For fair performance comparison, we will firstly apply ListMLE to do the initial ranking to capture relevance, and then use SVMDIV to re-rank top-K retrieved documents to capture diversity.

The above three diversity baselines: MMR, xQuAD and PM-2, all require a prior relevance function to implement their diversification steps. In our experiment, we choose ListMLE as the relevance function to implement them, and denote them as: MMR$_{list}$, xQuAD$_{list}$ and PM-2$_{list}$, respectively.

According to the different ways in defining the relational function $h_S(R^i)$ in section 4.2.1, our R-LTR diversification approach has three variants, denoted as R-LTR$_{min}$, R-LTR$_{avg}$ and R-LTR$_{max}$, respectively.

5.1.4 Implementation

In our experiments, we use Indri toolkit (version 5.2)[4] as the retrieval platform. For the test query set on each dataset, we use a 5-fold cross validation with a ratio of 3:1:1, for training, validation and testing. The final test performance is reported as the average over all the folds.

For data preprocessing, we apply porter stemmer and stopwords removing for both indexing and query processing. We then extract features for each dataset as follows. For relevance, we use several standard features in LTR research [21], such as typical weighting models (e.g., TF-IDF, BM25, LM), and term dependency model [19, 38], as summarized in Table 1, where Q-D means that the feature is dependent on both query and document, and D means that the feature only depends on the document. For all the Q-D features, they are applied in five fields: body, anchor, title, URL and the whole document, resulting in 5 features in total, respectively. Additionally, the MRF feature has two types of values: ordered phrase and unordered phrase [19], so the total feature number is 10.

For three baseline models: MMR, xQuAD and PM-2, they all have a single parameter λ to tune. We perform a 5-fold cross validation to train λ through optimizing $ERR\text{-}IA$. Additionally, for xQuAD and PM-2, the official subtopics are used as a representation of taxonomy classes to simulate their best-case scenarios, and uniform probability for all subtopics is assumed as [29, 13].

For ListMLE and SVMDIV, we utilize the same training data generated by Algorithm 1 to train their model, and also conduct 5-fold cross validation. ListMLE adopts the relevance features summarized in Table 1. SVMDIV adopts the representative word level features with different importance criterion, as listed in their paper and released code [36]. As described in above subsection, SVMDIV will re-rank top-K retrieved documents returned by ListMLE. We test $K \in \{30, 50, 100\}$, and find it performs best at $K = 30$. Therefore, the following results of SVMDIV are achieved with $K = 30$.

For our approach, the learning rate η parameter in Algorithm 2 is chosen from 10^{-7} to 10^{-1}, and the best learning rate is obtained based on the performance of validation set.

5.2 Performance Comparison

5.2.1 Evaluation on Official Diversity Metrics

We now compare our approaches to the baseline methods on search result diversification. The results of performance comparison are shown in Table 2, 3, and 4. We also present the performance of top performing systems on Category-B reported by TREC [7, 10, 9], which are just taken as indicative references. The number in the parentheses are the relative improvements compared with the baseline method QL. Boldface indicates the highest scores among all runs.

From the results we an see that, our R-LTR outperform the plain LTR approach without diversification consideration, i.e. ListMLE, which can be viewed as a special case of our approach. Specifically, the relative improvement of R-LTR$_{min}$ over ListMLE is up to 41.87%, 49.71%, 29.17%, in terms of $ERR\text{-}IA$ on WT2009, WT2010, and WT2011, respectively. It indicates that our approach can tackle multi-criteria ranking problem effectively, with the consideration of both content-based information and diversity relationship among candidate objects.

Regarding the comparison among representative implicit and explicit diversification approaches, explicit methods (i.e. xQuAD and PM-2) show better performance than the implicit method (i.e. MMR) in terms of all the evaluation measures. MMR is the least effective due to its simple predefined "marginal relevance". The two explicit methods achieve comparable performance: PM-2$_{list}$ wins on WT2010 and WT2011, while xQuAD$_{list}$ wins on WT2009, but their overall performance differences are small.

Furthermore, our approach outperforms the state-of-the-art explicit methods in terms of all the evaluation measures. For example, with the evaluation of $ERR\text{-}IA$, the relative improvement of R-LTR$_{min}$ over the xQuAD$_{list}$ is up to 17.18%, 11.26%, 13.38%, on WT2009, WT2010, WT2011, respectively, and the relative improvement of R-LTR$_{min}$ over the PM-2$_{list}$ is up to 18.31%, 10.65%, 10.59% on WT2009, WT2010, WT2011, respectively. Although xQuAD$_{list}$ and PM-2$_{list}$ all utilize the official subtopics as explicit query aspects to simulate their best-case scenarios, their performances are still much lower than our learning-based approaches, which indicates that there might be certain gap between their heuristic predefined utility functions and the final evaluation measures.

Comparing with the learning-based diversification baseline method, our R-LTR approach also show better per-

[4]http://lemurproject.org/indri

Table 2: Performance comparison of all methods in official TREC diversity measures for WT2009.

Method	ERR-IA	α-NDCG	NRBP
QL	0.1637	0.2691	0.1382
ListMLE	0.1913 (+16.86%)	0.3074 (+14.23%)	0.1681 (+21.64%)
MMR_{list}	0.2022 (+23.52%)	0.3083 (+14.57%)	0.1715 (+24.09%)
xQuAD_{list}	0.2316 (+41.48%)	0.3437 (+27.72%)	0.1956 (+41.53%)
PM-2_{list}	0.2294 (+40.13%)	0.3369 (+25.20%)	0.1788 (+29.38%)
SVMDIV	0.2408 (+47.10%)	0.3526 (+31.03%)	0.2073 (+50.00%)
R-LTR_{min}	**0.2714** (+65.79%)	0.3915 (+45.48%)	**0.2339** (+69.25%)
R-LTR_{avg}	0.2671 (+63.16%)	**0.3964** (+47.31%)	0.2268 (+64.11%)
R-LTR_{max}	0.2683 (+63.90%)	0.3933 (+46.15%)	0.2281 (+65.05%)
TREC-Best	0.1922	0.3081	0.1617

Table 3: Performance comparison of all methods in official TREC diversity measures for WT2010.

Method	ERR-IA	α-NDCG	NRBP
QL	0.1980	0.3024	0.1549
ListMLE	0.2436 (+23.03%)	0.3755 (+24.17%)	0.1949 (+25.82%)
MMR_{list}	0.2735 (+38.13%)	0.4036 (+33.47%)	0.2252 (+45.38%)
xQuAD_{list}	0.3278 (+65.56%)	0.4445 (+46.99%)	0.2872 (+85.41%)
PM-2_{list}	0.3296 (+66.46%)	0.4478 (+48.08%)	0.2901 (+87.28%)
SVMDIV	0.3331 (+68.23%)	0.4593 (+51.88%)	0.2934 (+89.41%)
R-LTR_{min}	**0.3647** (+84.19%)	**0.4924** (+62.83%)	**0.3293** (+112.59%)
R-LTR_{avg}	0.3587 (+81.16%)	0.4781 (+58.10%)	0.3125 (+101.74%)
R-LTR_{max}	0.3639 (+83.79%)	0.4836 (+59.92%)	0.3218 (+107.74%)
TREC-Best	0.2981	0.4178	0.2616

formance than the SVMDIV approach. The relative improvement of R-LTR_{min} over the SVMDIV is up to 12.71%, 9.49%, 10.02%, in terms of *ERR-IA* on WT2009, WT2010, WT2011, respectively. SVMDIV simply uses weighted word coverage as a proxy for explicitly covering subtopics, while our R-LTR approach directly models the generation probability of the diverse ranking based on the sequential ranking formulation. Therefore, our R-LTR approach shows deeper understanding and better formulation of diverse ranking, and leads to better performance. We further conduct statistical tests on the results, which indicates that all these improvements are statistically significant ($p\text{-}value < 0.01$).

Among the R-LTR approaches, R-LTR_{min} obtains better performance than the other two variants especially on WT2010 and WT2011 data collection, although their performance difference is small. It indicates that when defining the diversity relation between a document and a set of documents, the minimal distance would be a better choice.

5.2.2 Evaluation on Traditional Diversity Metrics.

Additionally, we also evaluate all the methods under traditional diversity measures, i.e. Precision-IA and Subtopics Recall. The experimental results are shown in Figure 2 and 3. We can see that our approaches outperform all the baseline models on all the data collections in terms of both metrics, which is consistent with the evaluation results in Table 2, 3, and 4. It can further demonstrate the effectiveness of our approach on search result diversification from different aspects. When comparing the three variants of our R-LTR approaches, they all show similar performance and none obtains consistent better performance than the others under these two measures.

5.3 Robustness Analysis

In this section we analyze the robustness of these diversification methods, i.e., whether the performance improvement is consistent as compared with the basic relevance baseline QL. Specifically, we define the robustness as the Win/Loss ratio [36, 13] - the ratio of queries whose performance improves or hurts as compared with the original results from QL in terms of of *ERR-IA*.

Table 5: The robustness of the performance of all diversity methods in Win/Loss ratio

	WT2009	WT2010	WT2011	*Total*
ListMLE	20/18	27/16	26/11	73/45
MMR_{list}	22/15	29/13	29/10	80/38
xQuAD_{list}	28/11	31/12	31/12	90/35
PM-2_{list}	26/15	32/12	32/11	90/38
SVMDIV	30/12	32/11	32/11	94/34
R-LTR_{min}	**34/9**	**35/10**	**35/9**	**104/28**
R-LTR_{avg}	33/9	34/11	34/10	101/30
R-LTR_{max}	33/10	35/10	34/10	102/30

From results in Table 5, we find that our R-LTR methods achieve best as compared with all the baseline methods, with the total Win/Loss ratio around 3.49. Among the three variants of R-LTR methods, R-LTR_{min} performs better than the others, with the Win/Loss ratio as 3.71.

Based on the robustness results, we can see that the performance of our R-LTR approach is more stable than all the baseline methods. It demonstrates that the overall performance gains of our approach not only come from some small subset of queries. In other words, the result diversification for different queries could be well addressed under our approach.

5.4 Feature Importance Analysis

In this subsection, we analyze the relative importance of the proposed diversity features. Table 6 shows an ordered list of diversity features used in our R-LTR_{min} model according to the learned weights (average on three datasets). From the results, we can see that the subtopic diversity R_{ij1}(topic) is with the maximal weight, which is in accordance with

Table 4: Performance comparison of all methods in official TREC diversity measures for WT2011.

Method	ERR-IA	α-NDCG	NRBP
QL	0.3520	0.4531	0.3123
ListMLE	0.4172 (+18.52%)	0.5169 (+14.08%)	0.3887 (+24.46%)
MMR_{list}	0.4284 (+21.70%)	0.5302 (+17.02%)	0.3913 (+25.30%)
xQuAD_{list}	0.4753 (+35.03%)	0.5645 (+24.59%)	0.4274 (+36.86%)
PM-2_{list}	0.4873 (+38.44%)	0.5786 (+27.70%)	0.4318 (+38.26%)
SVMDIV	0.4898 (+39.15%)	0.5910 (+30.43%)	0.4475 (+43.29%)
R-LTR_{min}	**0.5389** (+53.10%)	**0.6297** (+38.98%)	**0.4982** (+59.53%)
R-LTR_{avg}	0.5276 (+49.89%)	0.6219 (+37.25%)	0.4724 (+51.26%)
R-LTR_{max}	0.5285 (+50.14%)	0.6223 (+37.34%)	0.4741 (+51.81%)
TREC-Best	0.4380	0.5220	0.4070

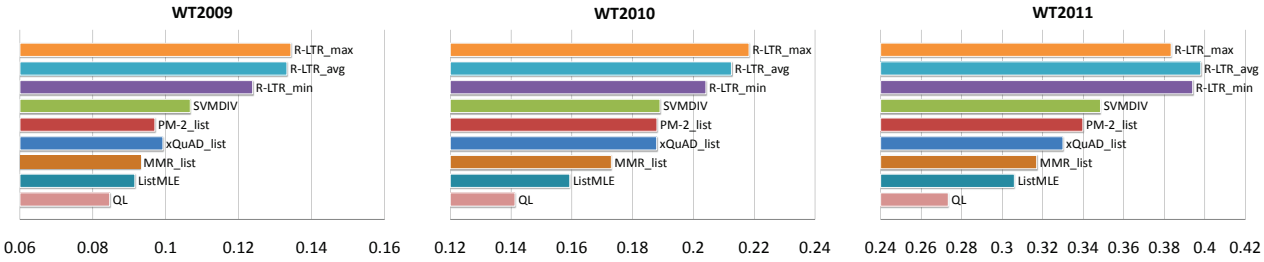

Figure 2: Performance comparison of all methods in Precision-IA for WT2009, WT2010, WT2011.

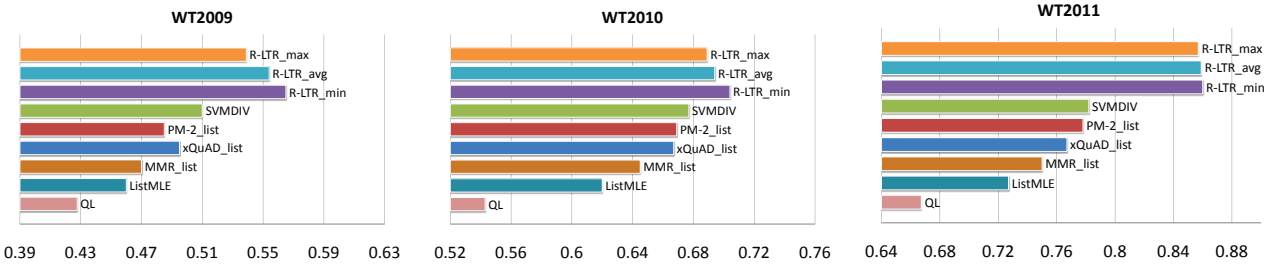

Figure 3: Performance comparison of all methods in Subtopic Recall for WT2009, WT2010, WT2011.

Table 6: Order list of diversity features with corresponding weight value.

feature	weight
R_{ij1}(topic)	3.71635
R_{ij3}(title)	1.53026
R_{ij4}(anchor)	1.34293
R_{ij2}(text)	0.98912
R_{ij5}(ODP)	0.52627
R_{ij6}(Link)	0.04683
R_{ij7}(URL)	0.01514

our intuition that diversity mainly lies in the rich *semantic information*. Meanwhile, the title and anchor text diversity R_{ij3}(title) and R_{ij4}(anchor) also work well, since these fields typically provide a precise summary of the content of the document. Finally, The Link and URL based diversity R_{ij6}(Link) and R_{ij7}(URL) seem to be the least important features, which may be due to the sparsity of such types of features in the data.

As a learning-based method, our model is flexible to incorporate different types of features for capturing both the relevance and diversity. Therefore, it would be interesting to explore other useful features under our R-LTR framework to further improve the performance of diverse ranking. We will investigate this issue in future.

5.5 Running Time Analysis

We further study the efficiency of our approach and the baseline models. All of the diversity methods associate with a sequential selection process, which is time-consuming due to the consideration of the dependency relations of document pairs. While as discussed before, this type of algorithms all have time complexity of $O(n * K)$, With a small K, the prediction time is linear.

All the learning-based methods (i.e. ListMLE, SVMDIV and R-LTR) need additional offline training time due to the supervised learning process. We compare the average training time of different learning-based methods, and the result is shown as following (unit: hour):

$$\text{ListMLE } (\sim 1.5h) \prec \text{SVMDIV } (\sim 2h) \prec \text{R-LTR } (\sim 3h)$$

We can observe that our approach takes longer but comparable offline training time among different learning-based methods. Besides, in our experiments, we also found that the three variants of our R-LTR approach are with nearly the same training time. We will attempt to optimize our

code to provide much faster training speed via parallelization technique in the following work.

6. CONCLUSIONS

In this paper, we propose to solve the search result diversification problem within a novel R-LTR framework. However, the specific definitions of ranking function and loss function are challenging. Motivated by the top-down user browsing behavior and the ubiquitous greedy approximation for diverse ranking, we firstly define the ranking function as the combination of relevance score and diversity score between the current item and those previously selected. Then the loss function is defined as the likelihood loss of ground truth based on Plackett-Luce model, which can naturally model the sequential generation of a diverse ranking list. On this basis, we utilize stochastic gradient descent to conduct the unconstrained optimization. The prediction of a diverse ranking list is then provided by iteratively maximizing the learned ranking function. Finally the experimental results on public TREC data collections demonstrate the effectiveness and robustness of our approach.

The proposed R-LTR framework is quite general that can be used in other applications, such as pseudo relevance feedback and topic distillation. Therefore, it would be interesting to apply our R-LTR framework in different applications in our future work.

7. ACKNOWLEDGEMENTS

This research work was funded by the 973 Program of China under Grants No.2012CB316303 and No.2013CB329602, 863 program of China under Grants No.2012AA011003, National Natural Science Foundation of China under Grant No.61232010 and No.61203298, and National Key Technology R&D Program under Grants No.2012BAH39B02 and No.2012BAH46B04.

8. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the 2th ACM WSDM*, pages 5–14, 2009.

[2] C. Brandt, T. Joachims, Y. Yue, and J. Bank. Dynamic ranked retrieval. In *Proceedings of the 4th ACM WSDM*, pages 247–256, 2011.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st ACM SIGIR*, pages 335–336, 1998.

[4] B. Carterette. An analysis of np-completeness in novelty and diversity ranking. In *Proceedings of the 2nd ICTIR*, 2009.

[5] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM CIKM*, pages 1287–1296, 2009.

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

[7] C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *TREC*, 2009.

[8] C. L. Clarke, N. Craswell, I. Soboroff, and A. Ashkan. A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the 4th ACM WSDM*, pages 75–84, 2011.

[9] C. L. Clarke, N. Craswell, I. Soboroff, and E. M.Voorhees. Overview of the trec 2011 web track. In *TREC*, 2011.

[10] C. L. Clarke, N. Craswell, I. Soboroff, and G. V.Cormack. Overview of the trec 2010 web track. In *TREC*, 2010.

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

[12] C. L. Clarke, M. Kolla, and O. Vechtomova. An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of the 2nd ICTIR*, pages 188–199, 2009.

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

[14] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th WWW*, pages 381–390, 2009.

[15] J. He, V. Hollink, and A. de Vries. Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th ACM SIGIR*, pages 851–860, 2012.

[16] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd ACM SIGIR*, pages 50–57, 1999.

[17] T.-Y. Liu. *Learning to Rank for Information Retrieval*. Springer, 2011.

[18] J. I. Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall, 1995.

[19] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th ACM SIGIR*, pages 472–479, 2005.

[20] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions–i. *Mathematical Programming*, 14(1):265–294, 1978.

[21] T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.*, pages 346–374, 2010.

[22] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, and H. Li. Global ranking using continuous conditional random fields. In *Proceedings of the 22th NIPS, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1281–1288, 2008.

[23] T. Qin, T.-Y. Liu, X.-D. Zhang, D.-S. Wang, W.-Y. Xiong, and H. Li. Learning to rank relational objects and its application to web search. In *Proceedings of the 17th WWW*, pages 407–416, 2008.

[24] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th ACM SIGIR*, 2006.

[25] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th ICML*, pages 784–791, 2008.

[26] D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. In *Proceedings of the 19th WWW*, pages 781–790, 2010.

[27] K. Raman, T. Joachims, and P. Shivaswamy. Structured learning of two-level dynamic rankings. In *Proceedings of the 20th ACM CIKM*, pages 291–296, 2011.

[28] K. Raman, P. Shivaswamy, and T. Joachims. Online learning to diversify from implicit feedback. In *Proceedings of the 18th ACM SIGKDD*, pages 705–713, 2012.

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

[30] P. Shivaswamy and T. Joachims. Online structured prediction via coactive learning. In *ICML'12*, 2012.

[31] A. Slivkins, F. Radlinski, and S. Gollapudi. Learning optimally diverse rankings over large document collections. In *Proceedings of the 27th ICML*, pages 983–990, 2010.

[32] S. Vargas, P. Castells, and D. Vallet. Explicit relevance models in intent-oriented information retrieval diversification. In *Proceedings of the 35th ACM SIGIR*, pages 75–84, 2012.

[33] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd ACM SIGIR*, pages 115–122, 2009.

[34] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th ICML*, pages 1192–1199, 2008.

[35] Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *NIPS*, pages 2483–2491, 2011.

[36] Y. Yue and T. Joachims. Predicting diverse subsets using structural svms. In *Proceedings of the 25th ICML*, pages 1224–1231, 2008.

[37] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proc. of the 26th ACM SIGIR*, pages 10–17, 2003.

[38] Y. Zhu, Y. Xue, J. Guo, Y. Lan, X. Cheng, and X. Yu. Exploring and exploiting proximity statistic for information retrieval model. In *Proceedings of the 8th Asia Information Retrieval Societies Conference*, volume 7675 of *Lecture Notes in Computer Science*, pages 1–13, 2012.