Adapting Markov Decision Process for Search Result Diversification

Long Xia, Jun Xu*, Yanyan Lan, Jiafeng Guo, Wei Zeng, Xueqi Cheng
CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences
{xialong,zengwei}@software.ict.ac.cn,{junxu,lanyanyan,guojiafeng,cxq}@ict.ac.cn

ABSTRACT

In this paper we address the issue of learning diverse ranking models for search result diversification. Typical methods treat the problem of constructing a diverse ranking as a process of sequential document selection. At each ranking position, the document that can provide the largest amount of additional information to the users is selected, because the search users usually browse the documents in a top-down manner. Thus, to select an optimal document for a position, it is critical for a diverse ranking model to capture the utility of information the user have perceived from the preceding documents. Existing methods usually calculate the ranking scores (e.g., the marginal relevance) directly based on the query and the selected documents, with heuristic rules or handcrafted features. The utility the user perceived at each of the ranks, however, is not explicitly modeled. In this paper, we present a novel diverse ranking model on the basis of continuous state Markov decision process (MDP) in which the user perceived utility is modeled as a part of the MDP state. Our model, referred to as MDP-DIV, sequentially takes the actions of selecting one document according to current state, and then updates the state for the chosen of the next action. The transition of the states are modeled in a recurrent manner and the model parameters are learned with policy gradient. Experimental results based on the TREC benchmarks showed that MDP-DIV can significantly outperform the state-of-the-art baselines.

KEYWORDS

learning to rank; search result diversification; Markov decision process

ACM Reference format:
Long Xia, Jun Xu*, Yanyan Lan, Jiafeng Guo, Wei Zeng, Xueqi Cheng. 2017. Adapting Markov Decision Process
for Search Result Diversification. In *Proceedings of SIGIR17, August 7–11, 2017, Shinjuku, Tokyo, Japan, ,* 10 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080775

* Corresponding author: Jun Xu.

SIGIR17, August 7–11, 2017, Shinjuku, Tokyo, Japan
 ISBN 978-1-4503-5022-8/17/08…$15.00.
DOI: http://dx.doi.org/10.1145/3077136.3080775

1 INTRODUCTION

In many information retrieval tasks, one important goal involves providing search results that covers a wide range of topics for a search query, called search result diversification [1]. One of the key problems in search result diversification is ranking, specifically, how to develop a ranking model that can sort documents based on their relevance to the given query as well as the novelty of the information in the documents.

Typical approaches to search result diversification, including the heuristic approaches and the learning approaches, treat the process of constructing a diverse ranking as a problem of sequential document selection. At each ranking position, the additional amount of information (utility) a document can provide is estimated, on the basis of the user query and the documents ranked ahead. The document that can provide maximal additional utility is selected. The sequential document selection matches well with the user activity of browsing the search results: search users usually browse the search results in a top-down manner. Thus, to accurately select the document at each of the positions, it is critical for a diverse ranking algorithm to model the utility of information the users have already perceived from the preceding documents.

Several methods for diverse ranking have been developed and applied to document retrieval. Different criteria are adopted in these methods to estimate the new utility a candidate document can provide. For example, in the representative heuristic approach of maximal marginal relevance (MMR) [2], the marginal relevance, which is defined as a sum of the query-document relevance and the maximal document distance, is used as the utility. In xQuAD [20], another widely used diverse ranking model, the utility is defined so as to explicitly account for the relationship between documents retrieved for the original query and the possible aspects underlying this query, in the form of sub-queries. In recent years, machine learning methods have been proposed and applied to search result diversification [18, 24–27, 31]. Typical diverse learning models, including the relational learning to rank (R-LTR) [31] and its variations [24–26], define the utilities as the linear combinations of the relevance features and the novelty features.

All the existing methods on diverse ranking [2, 20, 31] are designed to estimate the utility of a candidate document directly based on the user query and the preceding documents, calculated either by the carefully designed heuristics (e.g., the scoring functions in MMR and xQuAD) or as a linear combination of the handcrafted relevance features and novelty features (e.g., the scoring function in R-LTR). The utility perceived by the users from the preceding documents, however, is not explicitly modeled and fully utilized.

In this paper we propose to formalize the construction of a diverse ranking as a process of sequential decision making, which can

be modeled with a continuous state Markov decision process (MDP). The new diverse ranking model, referred to as MDP-DIV, model the user perceived utility of information as a part of its MDP state. Specifically, in MDP-DIV, a document ranking with M positions is considered as a sequence of M discrete time steps where each time step corresponds to a ranking position. The ranking of documents, thus, is formalized as a sequence of M decisions and each action corresponds to selecting one document from the candidate set. At each time step, the agent receives the environment's state, which models the user's dynamic state on the perceived utility, starting from the first ranking position. Based on the received state, the agent chooses an action. One time step later, as a consequence of the action, the search users perceive some additional utility from the new selected document, and the system transit to a new state. The transition function, which maps old state and the selected document to a new state, is implemented in a recurrent manner. At each time step, the chosen of the action depends on a policy, which is a function maps from the current state to a probability distribution of selecting each actions.

Reinforcement learning is employed to train the model parameters. Given a set of labeled queries, at each time step, the agent can receive a numerical action-dependent reward which can be defined upon the diversity evaluation measures. The policy gradient algorithm of REINFORCE [22] is adopted to adjust the model parameters so that expected long-term discounted rewards in terms of the diversity evaluation measure is maximized. In the testing phase, the system fully trusts the learned policy. Given a query and the associated documents, the action with the maximal probability is selected at each ranking position.

Advantages of the proposed model include: 1) explicitly modeling the dynamic state on the user perceived utility of information in diverse ranking learning, which unifies the relevance and novelty and can be utilized as the criterion for selecting documents; 2) ability to conduct diverse ranking learning in an end-to-end manner, achieving a diverse ranking model with no need of handcrafting features; 3) ability to learn a ranking model towards to a diversity evaluation measure, via involving the measure in the training.

To evaluate the effectiveness of MDP-DIV, we conducted experiments on the basis of TREC benchmark datasets. The experimental results showed that MDP-DIV can significantly outperform the state-of-the-art diverse ranking approaches including the heuristic methods of MMR, xQuAD, and the learning methods of R-LTR, PAMM, and PAMM-NTN. We analyzed the results and showed that MDP-DIV improved the performances through 1) optimizing the diversity evaluation measures in training, 2) modeling the dynamic user state on the perceived utility, and 3) utilizing both the immediate rewards and the long-term returns in training phase.

2 RELATED WORK

2.1 Search result diversification

It is a common practice to formalize the construction of a diverse ranking list in search as a process of sequential document selection. This is based on the observation that in diverse ranking the additional utility a document can provide depends on not only the document itself but also the preceding documents. Different models designed different criteria for estimating the utility the search users can perceive from a candidate document. Following the idea, Carbonell and Goldstein [2] proposed the maximal marginal relevance criterion to guide the selection of the documents. At each iteration, the document with the highest marginal relevance score is selected, where the score is a linear combination of the query-document relevance and the maximum distance of the document to the documents in current result set, in other words, novelty. The marginal relevance score is then updated in the next iteration as the number of documents in the result set increases by one. Based on MMR, Guo and Sanner [7] proposed the probabilistic latent MMR model. xQuAD [19] directly models different aspects underlying the original query in the form of sub-queries, and estimates the utility as the relevance of the retrieved documents to each identified sub-query. PM-2 [5] treats the problem of finding a diverse search result as finding a proportional representation for the document ranking. Hu et al. [9] proposed a utility function that explicitly leverages the hierarchical intents of queries and selects the documents that maximize diversity in the hierarchical structure. Evaluation methods have also developed based on the intent hierarchies [23]. He et al. [8] proposed to combine the implicit and explicit topic representations for constructing better diverse rankings. Gollapudi and Sharma [6] proposed an axiomatic approach to result diversification.

Machine learning techniques, which automatically learn the ranking models from the human labeled data, have been applied to construct diverse ranking models. Most of learning approaches still adopt sequential document selection as the basic framework, and the additional utility a candidate document can provide is usually modeled as a sum of the relevance score and the novelty score. For example, Zhu et al. [31], Xia et al. [24], and Xu et al. [26] employed a set of handcrafted relevance features and novelty features to calculate the relevance score and the novelty score, respectively. Both of the scores are defined as linear combinations of the features. Xia et al.[25] proposed to model the novelty score with the deep learning model of neural tensor networks. SVM-DIV [18] propose to construct a diverse ranking with the diversity criterion only. Structured output learning [11] and deep learning models [15] have also been employed to address the problem of learning diverse rankings.

Existing methods calculates the ranking scores directly based on the query and the selected documents, with the heuristic rules or the ranking features. Though it is a critical issue for constructing optimal diverse rankings, the dynamic utility the search user perceived from the preceding documents is still not explicitly modeled and fully utilized in current diverse ranking methods.

2.2 MDP for information retrieval

In this paper we employ MDP for constructing diverse ranking model, which has been widely used in variant IR applications. For example, in [13], a win-win search framework based on partially observed Markov decision process (POMDP) is proposed to model session search as a dual-agent stochastic game. In the model, the state of the search users are encoded as a four hidden decision making states. In [30], the log-based document re-ranking is also modeled as a POMDP to improve the re-ranking performances. MDP is also used for building recommender systems. For example, [21] designed an MDP-based recommendation model for taking

both the long-term effects of each recommendation and the expected value of each recommendation into account. Besides the MDP, researchers also employed the bandits model for constructing diverse ranking [18] and optimizing IR system [28].

Recent advances in deep learning makes it possible to incorporate deep learning methods with sequential decision making. In the literature of vision, Mnih et al. [16] attempts to implement attentional processing in a deep learning framework. Lu and Yang[12] proposes POMDP-Rec, a neural-optimized POMDP algorithm, for building a collaborative filtering recommender system.

Though MDP has been applied to various information retrieval tasks, applying it to learning to rank and search result diversification is hard. The difficulties lie in how to formalize diverse ranking under the MDP framework and how to convert the human labels to the supervision information that can be utilized by MDP. In this paper, we propose to formulate the diverse ranking learning as a problem of learning an MDP model.

3 MARKOV DECISION PROCESS

In the paper, we employ continuous state MDP[17, 22], a widely used sequential decision making model, for learning the diverse ranking. An MDP is composed by states, actions, rewards, policy, and transitions, and represented by a tuple $\langle S, A, T, R, \pi \rangle$:

States S is a set of states. For instance, in this paper we define the state as a tuple consisting of preceding document ranking, candidate documents, and the utility the user perceived from the preceding documents.

Actions A is a discrete set of actions that an agent can take. The actions available may depend on the state s, denoted as $A(s)$.

Transition T is the state transition function $s_{t+1} = T(s_t, a_t)$ which specifies a function which maps a state s_t into a new state s_{t+1} in response to the action selected a_t.

Reward $r = R(s, a)$ is the immediate reward, also known as reinforcement. It gives the immediate reward of taking action a at state s.

Policy $\pi(a|s)$ describes the behaviors of an agent, which is a probability distribution over the possible actions. π is usually optimized to decide how to move around in the state space to optimize the long term return.

The agent and environment interact at each of a sequence of discrete time steps, $t = 0, 1, 2, \cdots$. At each time step t the agent receives some representation of the environment's state, $s_t \in S$, and on that basis selects an action $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions available in state s_t. One time step later, in part as a consequence of its action, the agent receives a numerical reward, $r_{t+1} \in \mathbb{R}$ and finds itself in a new state $s_{t+1} = T(s_t, a_t)$. Figure 1 illustrates the agent-environment interaction in MDP.

4 MDP FORMULATION OF DIVERSE RANKING

In this paper, we employ the continuous state MDP to model the construction of the diverse ranking.

4.1 The basic model

Suppose we are given a query $\mathbf{q}$, which is associated with a set of retrieved documents $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\} \subseteq \mathcal{X}$, where both the query $\mathbf{q}$ and the documents $\mathbf{x}_i$ are represented as L-dimensional preliminary representations, i.e., the vectors learned by the doc2vec model [10], and $\mathcal{X}$ is the set of all possible documents. The goal of diverse ranking is to construct a model that can rank the documents so that the top ranked documents cover a wide range of subtopics for a search query.

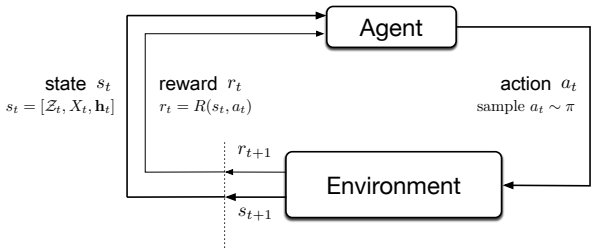

Figure 1: The agent-environment interaction in MDP.

Supervised learning approaches can be used to construct the model. Suppose we are given N labeled training queries $\{(q^{(n)}, X^{(n)}, J^{(n)})\}_{n=1}^{N}$, where $J^{(n)}$ denotes the human labels on the documents, in the form of a binary matrix. $J^{(n)}(i, j) = 1$ if document $\mathbf{x}_i^{(n)}$ contains the j-th subtopic of $q^{(n)}$ and 0 otherwise[1]. The learning of a diverse ranking model, thus, can be considered as the learning the parameters in an MDP model in which each time step corresponds to a ranking position. The states, actions, rewards, transitions, and policy of the MDP are set as:

States S: We design the state at time step t as a triple $s_t = [\mathcal{Z}_t, X_t, \mathbf{h}_t]$, where $\mathcal{Z}_t = \{\mathbf{x}_{(n)}\}_{n=1}^{t}$ is the sequence of t preceding documents, where $\mathbf{x}_{(n)}$ is the document ranked at position n. Note that we define $\mathcal{Z}_0 = \emptyset$ is a empty sequence; $X_t \in 2^{\mathcal{X}}$ is the set of candidate documents; $\mathbf{h}_t \in \mathbb{R}^K$ is a vector that encodes the user perceived utility from preceding documents in $\mathcal{Z}_t$, as well as the information need based on $\mathbf{q}$.

At the beginning ($t = 0$), the state is initialized as $s_0 = [\mathcal{Z}_0 = \emptyset, X_0 = X, \mathbf{h}_0]$: $\mathcal{Z}_0$ is initialized as an empty sequence $\emptyset$, the candidate set X_0 contains all of the M documents in X, and $\mathbf{h}_0$ is initialized as the user's initial information needs, implemented with a nonlinear transformation of the query:

$$\mathbf{h}_0 = \sigma(\mathbf{V}_q \mathbf{q}), \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^L$ is the preliminary representation of the user issued query, $\mathbf{V}_q \in \mathbb{R}^{K \times L}$ is the transformation matrix, and $\sigma(\mathbf{x})$ is the nonlinear sigmoid function applied to each of the entries:

$$\sigma(\mathbf{x}) = \sigma(\langle x_1, \cdots, x_K \rangle) = \left\langle \frac{1}{1 + e^{-x_1}}, \cdots, \frac{1}{1 + e^{-x_K}} \right\rangle.$$

Actions A: At each time step t, the $A(s_t)$ is the set of actions the agent can choose, each corresponds to selecting a document from X_t. That is, the action at the time step t, $a_t \in A(s_t)$ selects a document $\mathbf{x}_{m(a_t)} \in X_t$ for the ranking position $t + 1$, where $m(a_t)$ is the index of the document selected by a_t.

[1]Some datasets also use graded judgements. In this paper, we assume that all labels are binary.

Transition T: The transition function $T : S \times A \rightarrow S$ also consists of three parts, as shown in the following Equation (2):

$$\begin{aligned} s_{t+1} &= T(s_t, a_t) \\ &= T([\mathcal{Z}_t, X_t, \mathbf{h}_t], a_t) \\ &= \left[\mathcal{Z}_t \oplus \{\mathbf{x}_{m(a_t)}\}, X_t \setminus \{\mathbf{x}_{m(a_t)}\}, \sigma(\mathbf{V}\mathbf{x}_{m(a_t)} + \mathbf{W}\mathbf{h}_t)\right], \end{aligned} \tag{2}$$

where $\oplus$ concatenates the old sequence $\mathcal{Z}_t$ with $\mathbf{x}_{m(a_t)}$, $\mathbf{V} \in \mathbb{R}^{K \times L}$ is the document-state transformation matrix, and $\mathbf{W} \in \mathbb{R}^{K \times K}$ is the state-state transformation matrix. At each time step t, based on state s_t the system chooses an action a_t. Then, the system moves to time step $t+1$ and the system transits to a new state s_{t+1}: First, the system appends the selected document to the end of $\mathcal{Z}_t$, generating a new document sequence; Second, the selected document at step t is removed from the candidate set: $X_{t+1} = X_t \setminus \{\mathbf{x}_{m(a_t)}\}$. Thus, the number of actions the agent can choose at step $t+1$ is reduced by one. Third, the information from the user's last state and the selected document are combined together to form a new user state.

Note that in the initialization of $\mathbf{h}$, the parameter $\mathbf{V}_q$ is used for transforming the query to state. In the state transformation, another parameter $\mathbf{V}$ is used for transforming the selected document to state. The setting is based on the consideration that they have different goals: $\mathbf{V}_q$ is for transforming the query $\mathbf{q}$ which represents the information needs of the search users; $\mathbf{V}$ is for transforming the documents $\mathbf{x}$ which contain the utility that can be perceived by the users for fulfilling the information needs.

Also note that though the state transition function is implemented in a recurrent fashion, they have striking difference with recurrent neural networks (RNN): in MDP-DIV the input at time step t depends on the output (action) at the time step $t-1$.

Reward R: The reward can be considered as an evaluation of the quality of the selected document. In search result diversification, the diversity evaluation measures are used to evaluate the quality of a ranking. Most of these measures are calculated in a sequential manner. Thus, it is natural to define the reward function on the basis of the diversity evaluation measures. For example, based on the diversity evaluation measure of α-DCG, we can define the reward function as the promotion of α-DCG caused by choosing the action a_t:

$$R_{\alpha\text{-DCG}}(s_t, a_t) = \alpha\text{-DCG}[t+1] - \alpha\text{-DCG}[t],$$

where α-DCG$[t]$ is the discounted cumulative gain [4] at the t-th position, and the α-DCG value at the rank 0 is defined as zero: α-DCG$[0] = 0$. [2]

Similarly, on the basis of diversity evaluation measure of S-recall [29], we can also define another reward which is the promotion of S-recall by the action:

$$R_{\text{S-recall}}(s_t, a_t) = \text{S-recall}[t+1] - \text{S-recall}[t],$$

where S-recall$[t]$ is the S-recall value at the t-th position, and S-recall$[0] = 0$.

Since the training algorithm learns the model parameters under the supervision of the rewards, defining the rewards according to a diversity evaluation measure can guide the training process to achieve an optimal model in terms of that evaluation measure.

[2]The calculation of reward is based on the document sequence $\mathcal{Z}_t$ in s_t, the selected documents $\mathbf{x}_{m(a_t)}$, and the relevance labels of these documents. Here we assume that the state s_t also contains the document labels in the training phase.

Policy $\pi(a|s)$: The policy $\pi : A \times S \rightarrow [0, 1]$ defines the probability of selecting each action. Given current state $s_t = [\mathcal{Z}_t, X_t, \mathbf{h}_t]$ and a possible action a_t, the policy π is defined as a normalized soft-max function whose input is the bilinear product of the utility and the selected document:

$$\pi(a_t|[\mathcal{Z}_t, X_t, \mathbf{h}_t]) = \frac{\exp\left\{\mathbf{x}_{m(a_t)}^T \mathbf{U}\mathbf{h}_t\right\}}{Z}, \tag{3}$$

where $\mathbf{U} \in \mathbb{R}^{L \times K}$ is the parameter in the bilinear product and Z is the normalization factor:

$$Z = \sum_{a \in A(s_t)} \exp\left\{\mathbf{x}_{m(a)}^T \mathbf{U}\mathbf{h}_t\right\}.$$

Construction of a diverse ranking for the queries in the training data can be formalized as: given a user query $\mathbf{q}$, a set of M candidate documents X, and the corresponding human labels J, the system state is initialized as $s_0 = [\mathcal{Z}_0 = \emptyset, X_0 = X, \mathbf{h}_0 = \sigma(\mathbf{V}_q\mathbf{q})]$. Then, at each of the time steps $t = 0, \cdots, M-1$, the agent receives the state $s_t = [\mathcal{Z}_t, X_t, \mathbf{h}_t]$, chooses an action a_t which selects the document $\mathbf{x}_{m(a_t)}$ from the candidate set, and places it to the rank $t+1$. Moving to the next step $t+1$, the state becomes $s_{t+1} = [\mathcal{Z}_{t+1}, X_{t+1}, \mathbf{h}_{t+1}]$. On the basis of the human labels J for the query, the agent receives immediate reward $r_{t+1} = R([\mathcal{Z}_t, X_t, \mathbf{h}_t], a_t)$, which could be used as supervision for training the model parameters. The process is repeated until the candidate set becomes empty.

Note that in online ranking/testing phase, there is no reward available because the queries are unlabeled. To construct a diverse ranking, we fully trust the learned policy and choose the action with maximal probability at each time step.

Next, we will discuss the off-line training algorithm and online ranking algorithm.

4.2 Learning with policy gradient

The model has parameters $\Theta = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$ to learn. Inspired by the REINFORCE [22] algorithm of policy gradient, we devised a novel algorithm which can learn the parameters toward the diversity evaluation measure. The algorithm is referred as MDP-DIV and shown in Algorithm 1. The Algorithm 2 shows the procedure of sampling an episode for Algorithm 1.

The basic idea of Algorithm 1 is updating the parameters via Monte-Carlo stochastic gradient ascent. At each iteration, an episode (consisting a sequence of M states, actions, and rewards) is sampled according to current policy. Then, at each time step t of the sampled episode, the model parameters are adjusted according to the gradients of the parameters $\nabla_\Theta \log \pi(a_t|s_t; \Theta)$, scaled by the step size η, discount rate γ^t, and the long-term return G_t, where G_t is defined as the discounted sum of the rewards from position t:

$$G_t = \sum_{k=0}^{M-1-t} \gamma^k r_{t+k+1}, \tag{4}$$

where $M = |X|$ is the number documents in the candidate set. Note that if $\gamma = 1$, G_0 is exactly the evaluation measure calculated at the final rank of the document list, i.e., α-DCG@M or S-recall@M. Intuitively, the setting of G_t let the parameters move most in the directions so that the favor actions can yield the highest return.

Algorithm 1 MDP-DIV learning

Input: Labeled training set $D = \{(\mathbf{q}^{(n)}, X^{(n)}, J^{(n)})\}_{n=1}^{N}$, learning rate η, discount factor γ, and reward function R
Output: $\Theta = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$

Initialize $\Theta = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\} \leftarrow$ random values in $[-1, 1]$
repeat
  for all $(\mathbf{q}, X, J) \in D$ **do**
    $(s_0, a_0, r_1, \cdots, s_{M-1}, a_{M-1}, r_M) \leftarrow$ SampleEpisode$(\Theta, \mathbf{q}, X, J, R)$ {Algorithm (2), and $M = |X|$}
    for $t = 0$ **to** $M - 1$ **do**
      $G_t \leftarrow \sum_{k=0}^{M-1-t} \gamma^k r_{t+k+1}$ {Equation (4)}
      $\Theta \leftarrow \Theta + \eta \gamma^t G_t \nabla_\Theta \log \pi(a_t|s_t; \Theta)$ {Equation (5)}
    end for
  end for
until converge
return Θ

The gradient of MDP-DIV at time step t is $\nabla_\Theta \log \pi(a_t|s_t; \Theta)$, which the direction that most increase the probability of repeating the action a_t on future visits to state s_t, and is defined as

$$\nabla_\Theta \log \pi(a_t|s_t; \Theta) = \nabla_\Theta f(a_t|s_t) - \frac{\sum_{a \in A_t} \nabla_\Theta f(a|s_t) \exp\{f(a|s_t)\}}{\sum_{a \in A_t} \exp\{f(a|s_t)\}}, \tag{5}$$

where $f(a|s_t) = \mathbf{x}_{m(a)}^T (\mathbf{U}\mathbf{h}_t)$, and $\nabla_\Theta f(a|s_t) = \{\nabla_\mathbf{U} f(a|s_t), \nabla_{\mathbf{V}_q} f(a|s_t), \nabla_\mathbf{V} f(a|s_t), \nabla_\mathbf{W} f(a|s_t)\}$, where

$$\nabla_\mathbf{U} f(a|s_t) = \mathbf{x}_{m(a)} \mathbf{h}_t^T.$$

As for $\nabla_{\mathbf{V}_q} f(a|s_t)$, $\nabla_\mathbf{V} f(a|s_t)$, and $\nabla_\mathbf{W} f(a|s_t)$, they can be calculated in a similar way:

$$\nabla_{\mathbf{V}_q} f(a|s_t) = \left(\nabla_{\mathbf{V}_q} \mathbf{h}_t\right) \mathbf{U}^T \mathbf{x}_{m(a)},$$
$$\nabla_\mathbf{V} f(a|s_t) = \left(\nabla_\mathbf{V} \mathbf{h}_t\right) \mathbf{U}^T \mathbf{x}_{m(a)},$$
$$\nabla_\mathbf{W} f(a|s_t) = \left(\nabla_\mathbf{W} \mathbf{h}_t\right) \mathbf{U}^T \mathbf{x}_{m(a)},$$

where $\nabla_{\mathbf{V}_q} \mathbf{h}_t$, $\nabla_\mathbf{V} \mathbf{h}_t$, and $\nabla_\mathbf{W} \mathbf{h}_t$ can be calculated recursively:

$$\begin{aligned}\nabla_{\mathbf{V}_q} \mathbf{h}_t &= \nabla_{\mathbf{V}_q} \sigma(\mathbf{V}\mathbf{x}_{m(a_{t-1})} + \mathbf{W}\mathbf{h}_{t-1}) \\ &= \text{diag}(\mathbf{h}_t \circ (\mathbf{1} - \mathbf{h}_t)) \left(\nabla_{\mathbf{V}_q} (\mathbf{W}\mathbf{h}_{t-1})\right) \\ &= \text{diag}(\mathbf{h}_t \circ (\mathbf{1} - \mathbf{h}_t)) \left(\nabla_{\mathbf{V}_q} \mathbf{h}_{t-1}\right) \mathbf{W}^T,\end{aligned}$$

where $\mathbf{1}$ is an K-dimensional vector of ones, operator "$\circ$" denotes the element-wise vector product, operator "diag" generates an $K \times K$ diagonal matrix according to the input vector, and $\nabla_{\mathbf{V}_q} \mathbf{h}_{t-1}$ can be further unrolled in a similar way. At $t = 0$, $\nabla_{\mathbf{V}_q} \mathbf{h}_0$ is:

$$\nabla_{\mathbf{V}_q} \mathbf{h}_0 = \nabla_{\mathbf{V}_q} \sigma(\mathbf{V}_q \mathbf{q}) = \text{diag}(\mathbf{h}_0 \circ (\mathbf{1} - \mathbf{h}_0)) \mathbf{I}_{K,L,K,L} \mathbf{q},$$

where $\mathbf{I}_{K,L,K,L} \in \mathbb{R}^{K \times L \times K \times L}$ is an identity tensor.

$$\begin{aligned}\nabla_\mathbf{V} \mathbf{h}_t &= \nabla_\mathbf{V} \sigma(\mathbf{V}\mathbf{x}_{m(a_{t-1})} + \mathbf{W}\mathbf{h}_{t-1}) \\ &= \text{diag}(\mathbf{h}_t \circ (\mathbf{1} - \mathbf{h}_t)) \left(\nabla_\mathbf{V} (\mathbf{V}\mathbf{x}_{m(a_{t-1})} + \mathbf{W}\mathbf{h}_{t-1})\right) \\ &= \text{diag}(\mathbf{h}_t \circ (\mathbf{1} - \mathbf{h}_t)) \left(\mathbf{I}_{K,L,K,L} \mathbf{x}_{m(a_{t-1})} + (\nabla_\mathbf{V} \mathbf{h}_{t-1}) \mathbf{W}^T\right),\end{aligned}$$

Algorithm 2 SampleEpisode

Input: Parameters $\Theta = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$, $\mathbf{q}$, X, J, and R
Output: An episode

Initialize $s \leftarrow [\emptyset, X, \sigma(\mathbf{V}_q \mathbf{q})]$ {Equation (1)}
$M \leftarrow |X|$
$E = ()$ {empty episode}
for $t = 0$ **to** $M - 1$ **do**
  $A \leftarrow A(s)$ {Possible actions according to X in state s}
  for all $a \in A$ **do**
    $P(a) \leftarrow \pi(a|s; \Theta)$
  end for
  Sample an action $\hat{a} \in A$, according to P
  $r \leftarrow R(s, \hat{a})$ {Calculation on the basis of J}
  Append $(s, \hat{a}, r)$ to the tail of E
  $[\mathcal{Z}, X, \mathbf{h}] \leftarrow s$
  $s \leftarrow \left[\mathcal{Z} \oplus \{\mathbf{x}_{m(\hat{a})}\}, X \setminus \{\mathbf{x}_{m(\hat{a})}\}, \sigma(\mathbf{V}\mathbf{x}_{m(\hat{a})} + \mathbf{W}\mathbf{h})\right]$
end for
return $E = (s_0, a_0, r_1, \cdots, s_{M-1}, a_{M-1}, r_M)$

where $\nabla_\mathbf{V} \mathbf{h}_{t-1}$ can be unrolled in a similar way. At $t = 0$, $\nabla_\mathbf{V} \mathbf{h}_0$ is:

$$\nabla_\mathbf{V} \mathbf{h}_0 = \nabla_\mathbf{V} \sigma(\mathbf{V}_q \mathbf{q}) = \mathbf{0}_{K,L,1,K},$$

where $\mathbf{0}_{K,L,1,K} \in \mathbb{R}^{K \times L \times 1 \times K}$ is a tensor of zeros.

$$\begin{aligned}\nabla_\mathbf{W} \mathbf{h}_t &= \nabla_\mathbf{W} \sigma(\mathbf{V}\mathbf{x}_{m(a_{t-1})} + \mathbf{W}\mathbf{h}_{t-1}) \\ &= \text{diag}(\mathbf{h}_t \circ (\mathbf{1} - \mathbf{h}_t)) \left(\nabla_\mathbf{W} (\mathbf{V}\mathbf{x}_{m(a_{t-1})} + \mathbf{W}\mathbf{h}_{t-1})\right) \\ &= \text{diag}(\mathbf{h}_t \circ (\mathbf{1} - \mathbf{h}_t)) \left(\mathbf{I}_{K,K,K,K} \mathbf{h}_{t-1} + (\nabla_\mathbf{W} \mathbf{h}_{t-1}) \mathbf{W}^T\right),\end{aligned}$$

where $\mathbf{I}_{K,K,K,K} \in \mathbb{R}^{K \times K \times K \times K}$ is an identity tensor, and $\nabla_\mathbf{W} \mathbf{h}_{t-1}$ can be unrolled in a similar way. At $t = 0$, $\nabla_\mathbf{W} \mathbf{h}_0$ is:

$$\nabla_\mathbf{W} \mathbf{h}_0 = \nabla_\mathbf{W} \sigma(\mathbf{V}_q \mathbf{q}) = \mathbf{0}_{K,K,1,K},$$

where $\mathbf{0}_{K,K,1,K} \in \mathbb{R}^{K \times K \times 1 \times K}$ is a tensor of zeros.

Compared with conventional REINFORCE algorithm, MDP-DIV is based on a modified MDP model in which the user state of perceived utility is initialized with query and modeled in a recurrent manner. Thus, in the training phase, MDP-DIV needs to estimate the policy function, as well as the functions for state initialization and state transition. In [16], similar idea was presented for extracting information from images. In this paper we adapt the model for the task of search result diversification.

4.3 Online ranking

In the online ranking, the ranking system receives a user query $\mathbf{q}$ and the associated documents $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$. Since there exists no human label for calculating the immediate rewards, the system fully relies on the learned policy π for generating the diverse ranking, as shown in Algorithm 3. After initializing with $\mathbf{q}$, the algorithm makes a sequence of greedy decisions: at each step the action with the maximal probability is chosen (line 5 of Alglrithm 3), and the action in return update the state for choosing the next action (line 7 and line 8 of Algorithm 3).

The time complexity of the online ranking algorithm is of $O\left(\min\{KL^2, LK^2\} \frac{M(2+M)}{4} + (M-1)(K^2 + KL)\right)$ per query. The first

Algorithm 3 MDP-DIV online ranking

Input: Parameters $\mathbf{\Theta} = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$, query $\mathbf{q}$, documents X
Output: Permutation of documents τ

```
Initialize s ← [∅, X, σ(V_q q)]{Equation (1)}
M ← |X|
for t = 0 to M − 1 do
    A ← A(s) {Possible actions according to X in state s}
    â ← arg max_{a∈A} π(a|s; Θ){Choosing most possible action}
    τ[t + 1] ← m(â){Document x_{m(â)} is ranked at t + 1}
    [Z, X, h] ← s
    s ← [Z ⊕ {x_{m(â)}}, X \ {x_{m(â)}}, σ(V x_{m(â)} + W h)]
end for
return τ
```

part corresponds to calculating the policy for all of the possible actions at each iteration and the second part corresponds to updating the state for the next iteration. The term $\min\{KL^2, LK^2\}$ is for calculating the matrix multiplication $\mathbf{x}_{m(a_t)}^T U \mathbf{h}_t$ in the policy with different ways. In most cases L is larger than K. Please note that the online ranking algorithm actually runs $M-1$ iterations for ranking M documents, because at the last iteration $A(s_{M-1})$ contains only one action. Usually, K and L are not very large, e.g, we set $K = 5$ and $L = 100$ in our experiments. Thus, the online ranking algorithm is efficient if the candidate set is not very large. In our experiments, on average it takes about 20 milliseconds for ranking about 200 documents, on a server with 24GB memory and two Intel Xeon E5410 2.33GHz Quad-Core processors. Note that in the analysis the time for document embedding is not taken into consideration as the document embeddings can be calculated offline.

4.4 Theoretical analysis

The learning phase of MDP-DIV tries to optimize general diversity evaluation measures with reinforcement learning. The measures can be α-DCG and S-recall, or any other measures that can be calculated at each of the ranking position. We explain why this is the case.

In the training, Monte-Carlo stochastic gradient ascent is used to conduct the optimization. Given a query $\mathbf{q}$ in the training set, we want to maximize the value V, which is the expected return of the query:

$$\max_{\mathbf{\Theta}} V(\mathbf{q}) = \mathbb{E}_\pi G_0,$$

where G_0 is the discounted sum of the rewards, starting from position 0, as defined in Equation (4). Please note G_0 is the diversity evaluation measure if $\gamma = 1$. Thus, maximizing $V(\mathbf{q})$ is actually maximizing the expected diversity evaluation measure for the query.

According to the policy gradient theorem presented in [22], chapter 13, the gradient of the performance metric with respect to the parameters $\mathbf{\Theta}$ on each query can be calculated as

$$\nabla_{\mathbf{\Theta}} V(\mathbf{\Theta}) = \mathbb{E}_{s\sim\rho, a\sim\pi} Q^\pi(s,a) \nabla_{\mathbf{\Theta}} \pi(a|s),$$

where ρ is the discounted state distribution given a query $\mathbf{q}$ and model parameters, which is defined as:

$$\rho(s|\mathbf{q};\mathbf{\Theta}) = \sum_{t=1}^{\infty} \gamma^{t-1} p(s_0 \to s, t|\mathbf{q}_n;\mathbf{\Theta}),$$

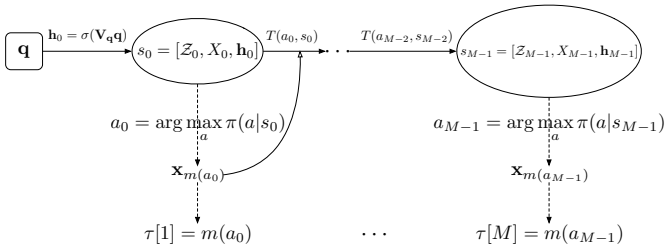

Figure 2: Online document ranking in MDP-DIV.

where $p(s_0 \to s, t|\mathbf{q}_n;\mathbf{\Theta})$ is the probability of transitioning from the initial state s_0 given the query $\mathbf{q}$ in t steps [22]. $Q^\pi(s,a)$ is the expected return starting from s, taking the action a and thereafter following the policy π:

$$Q^\pi(s,a) = \mathbb{E}_\pi[G_t|s_t = s, a_t = a].$$

Monte-Carlo method is used to estimate the gradient. Specifically, given a sampled episode $s_0, a_0, r_1, \cdots, s_{M-1}, a_{M-1}, r_M$ and a specific time step t, the gradient can be estimated as [22]

$$\begin{aligned}
\widehat{\nabla_{\mathbf{\Theta}} V(\mathbf{\Theta})} &\overset{\text{sample}}{=} \gamma^t \sum_{a\in A(s_t)} \nabla_{\mathbf{\Theta}} \pi(a|s_t) Q^\pi(s_t,a) \\
&= \gamma^t \sum_{a\in A(s_t)} \pi(a|s_t) \cdot \left(Q^\pi(s_t,a) \frac{\nabla_{\mathbf{\Theta}} \pi(a|s_t)}{\pi(a|s_t)} \right) \\
&\overset{\text{sample}}{=} \gamma^t Q^\pi(s_t,a_t) \frac{\nabla_{\mathbf{\Theta}} \pi(a_t|s_t)}{\pi(a_t|s_t)} \\
&\overset{\text{sample}}{=} \gamma^t G_t \nabla_{\mathbf{\Theta}} \log \pi(a_t|s_t).
\end{aligned}$$

The first $\overset{\text{sample}}{=}$ replaces s by its sample s_t, which is sampled according to ρ; the second $\overset{\text{sample}}{=}$ replaces a by its sample a_t, which is sampled according to π; and the third $\overset{\text{sample}}{=}$ replaces the thereafter decision process guided by π with the sampled episode. Note that $\mathbb{E}_\pi[G_t|s_t,a_t] = Q^\pi(s_t,a_t)$ and $\nabla_{\mathbf{\Theta}} \log \pi(a_t|s_t) = \frac{\nabla_{\mathbf{\Theta}} \pi(a_t|s_t)}{\pi(a_t|s_t)}$.

We can see that the updating rule in Algorithm 1 exactly follows the estimated gradients presented above. Thus, we can conclude MDP-DIV tries to optimize general diversity evaluation measures with Monte-Carlo stochastic gradient ascent when $\gamma = 1$.

4.5 Advantages

MDP-DIV provides an elegant approach to modeling user's dynamic state on the perceived utility during the browsing of the diverse ranking results. More importantly, it is a method that can be justified from the theoretical viewpoint, as discussed above. In addition, MDP-DIV has several other advantages when compared with the existing diverse ranking learning methods such as SVM-DIV, R-LTR and PAMM etc.

First, MDP-DIV can conduct an end-to-end learning of the diverse ranking model, which achieves a model with no need of handcrafting relevance features and novelty features. The inputs to the ranking model are the preliminary representations of the queries and the documents, e.g., the distributed representations learned by the doc2vec model. In contrast, all existing diverse ranking learning methods heavily depend on the handcrafted relevance

features and/or novelty features. It has been widely observed that high quality features are critical for constructing diverse ranking, while designing the features, especially designing the novelty features, is very difficult in real applications [25]. MDP-DIV solves the issue via learning a ranking model that needs only the preliminary representations of the queries and the documents.

Second, MDP-DIV utilizes both the immediate rewards and the long-term returns as the supervision information during its training. Specifically, given an episode, the parameters are updated after receiving each of the immediate rewards (line 5-8 of Algorithm 1). Meanwhile, the updating rule also utilizes the long-term return G_t, which accumulates all of the future rewards (line 6-7 of Algorithm 1), to re-scale the step size. In contrast, existing methods that directly optimize evaluation measures are only based on a evaluation measure calculated at a fixed position [24, 26] on the basis of whole ranking. Our empirical analysis in Section 5.3.2 also showed that training with both the rewards and the returns can achieve better ranking accuracies.

Third, MDP-DIV makes use of a unified criterion, the additional utility a search user can perceive, for selecting documents at each iteration. In contrast, the criterion adopted by most existing methods, e.g., the marginal relevance, consists of two individual factors: the relevance and the novelty. Heuristic diverse ranking model xQuAD tried to replace these two factors with "the relevance to the underlying sub-queries", which has shown to be more reasonable and effective. In this paper, we also showed that under the MDP framework, the document selection criterion can be unified to "the perceived utility", which has shown to be simple in concept and be powerful in the real applications.

5 EXPERIMENTS

We conducted experiments to test the performances of MDP-DIV using a combination of four TREC benchmark datasets: TREC 2009 Web Track (WT2009), TREC 2010 Web Track (WT2010), TREC 2011 Web Track (WT2011), and TREC 2012 Web Track (WT2012).

5.1 Experimental settings

The training of MDP-DIV model need lots of labeled queries because it has a large number of parameters. In experiments, for effective training of the model parameters, we combined four TREC datasets, achieving a new dataset with 200 queries, and in total about 45,000 labeled documents. Each query includes several subtopics identified by the TREC assessors. The document relevance labels are made at the subtopic level and the labels are binary[3].

All the experiments were carried out on the ClueWeb09 Category B data collection[4], which is comprised of 50 million English web documents. Porter stemming, tokenization, and stop-words removal (using the INQUERY list) were applied to the documents as preprocessing. For each query, the initial ranking is generated by Query-likelihood language model[14]. We conducted 5-fold cross-validation experiments. We randomly split the queries into five even subsets. At each fold, three subsets were used for training, one was used for validation, and one was used for testing. The results reported were the average over the five trials.

[3]WT2011 has graded judgements. In this paper we treat them as binary.
[4]http://boston.lti.cs.cmu.edu/data/clueweb09

The TREC official evaluation metrics for the diversity task were used in the experiments, including the ERR-IA [3] and α-NDCG [4]. They measure the diversity of a result list by explicitly rewarding diversity and penalizing redundancy observed at every rank. Following the default settings in official TREC evaluation program, the parameter α in these evaluation measures are set to 0.5. We also used traditional diversity measures of subtopic recall (denoted as "S-recall") [29]. All of the measures are computed over the top-k search results ($k = 5$ and $k = 10$).

We compared MDP-DIV with several state-of-the-arts baselines in search result diversification, including the heuristic methods:

MMR [2]: a heuristic approach in which the document is selected according to maximal marginal relevance.

xQuAD [19]: a representative approach which explicitly models different aspects underlying the original query in the form of sub-queries.

PM-2 [5]: a method of optimizing proportionality for search result diversification.

We also compared MDP-DIV with the learning methods:

SVM-DIV [27]: a learning approach which utilizes structural SVMs to optimize the subtopic coverage.

R-LTR [31]: a state-of-the-art learning approach developed in the relational learning to rank framework. Following the practice in [31], we used the results of R-LTR$_{\text{min}}$ in which the relation function was defined as the minimal distance of the candidate document to the selected documents

PAMM [24]: another learning algorithm under R-LTR framework. PAMM directly optimizes diversity evaluation measure using structured Perceptron. Following the practice in [24], we configured the PAMM algorithm to directly optimize α-NDCG@10 in our experiments, and set the number of sampled positive rankings per query $\tau^+ = 5$ and the number of sampled negative rankings per query $\tau^- = 20$.

NTN-DIV: a learning approach which automatically learns novelty features based on neural tensor networks. Following the practice in [25], we configured the learning of NTN-DIV algorithm to directly optimize α-NDCG@10 and the number of tensor slices is 7.

MDP-DIV and the baseline of NTN-DIV need preliminary representations of the queries and the documents as their inputs. In the experiments, we used the query vector and document vector generated by the doc2vec [10] to represent the document. Doc2vec model was trained on all of the documents in Web Track dataset and the number of vector dimensions were set to 100. For training the model, we used the distributed bag of words (DBOW) model[5]. The learning rate is set to 0.025 and the window size is set to 8.

The MDP-DIV also has some parameters. The reward function in MDP-DIV was set as either $R_{\alpha\text{-DCG}}$ or $R_{\text{S-recall}}$, denoted as MDP-DIV(α-DCG) and MDP-DIV(S-recall), respectively. In all of the experiments, the learning rate η is tuned on the basis of the validation set. We set the discounting parameter $\gamma = 1$, which means that the return is the undiscounted sum of the future rewards, which makes the long term return in Equation (4) becomes $G_t = \sum_{k=0}^{M-1-t} r_{t+k+1}$. It makes the training algorithm optimizes the diversity evaluation measure of α-DCG and S-recall.

[5]http://radimrehurek.com/gensim/tutorial.html

5.2 Experimental results

Table 1 reports the performances of our approach and all of the baseline methods in terms of the six diversity performance metrics, including α-NDCG@5, α-NDCG@10, S-recall@5, S-recall@10, ERR-IA@5, and ERR-IA@10. Boldface indicates the highest score among all runs. From the results we can see that, in terms of the six diversity evaluation metrics, both MDP-DIV(α-DCG) and MDP-DIV(S-recall) outperformed all of the baseline methods, including the heuristic method of MMR, xQuAD, PM-2 and learning methods of R-LTR, PAMM(α-NDCG), and NTN-DIV(α-NDCG). We conducted significance testing (t-test) on the improvements of our approaches over the best baseline NTN-DIV(α-NDCG). The results indicate that the improvements are significant (p-value < 0.05), in terms of all of the evaluation measures.

Comparing the results of the MDP-DIV(α-DCG) and MDP-DIV(S-recall), we can see that MDP-DIV(α-DCG) trained with α-DCG (setting α-DCG as reward function) performed better in terms of α-NDCG@5 and α-NDCG@10. Similarly, MDP-DIV(S-recall) trained with S-recall (setting S-recall as reward function) performed better in terms of S-recall@5 and S-recall@10. The results indicate that MDP-DIV can indeed enhance diverse ranking performance in terms of a measure by using the measure as reward function in training[6]. The result agrees well with the theoretical analysis shown in Section 4.4.

5.3 Discussion

We conducted experiments to show the reasons that MDP-DIV outperformed the baselines, using the results of MDP-DIV(α-DCG) on one trial of the cross validation as examples.

5.3.1 Effects of modeling user perceived utility. We analyzed how the user state on the perceived utility effects the selection of documents in MDP-DIV. Specifically, based on the trained MDP-DIV model, we tracked the online ranking process for query number 93 "ambiguous", which contains five subtopics. Figure 3 shows the details of the first three document selection steps, including the transition of the user dynamic state $\mathbf{h}_i$, the ranking score $f(a_t|s_t) = \mathbf{x}^T_{m(a_t)}\mathbf{U}\mathbf{h}_t$ for each of the actions[7], and the constructed document ranking. Due to the space limitation, we only showed the five top ranked documents $d_1, \cdots, d_5$, corresponding to the documents of enwp03-28-04544, en0007-80-16124, en0094-80-42411, en0006-08-03878, and en0010-24-38000 in the Clueweb09 collection, respectively. The subtopics covered by each of the documents are shown in the square brackets.

From Figure 3, we can see that $\mathbf{h}_t$ was updated after choosing each action, indicating the changes of the user state after perceiving the utility provided by the selected document. At step 0, the selected document d_2 covered subtopics 3 and 5. At step 1, as the consequence of the action the user state was updated, and the ranking score of d_4 (with the covered subtopic 5) was suppressed from 0.46 to 0.35, while the ranking scores of the other three documents (d_1, d_3, and d_5, with uncovered subtopics) were promoted. The results indicate that the user state $\mathbf{h}_1$ captured the utility provided by d_2, which made the ranking model focusing on documents that can provide the largest amount of new information. As the result, d_5, which contains two new subtopics 1 and 4, was selected at step 1. Similarly, at step 2 state vector $\mathbf{h}_2$ captured the utility provided by d_2 and d_5 and making the model to select d_1, which contains a new subtopic 2. In contrast, the ranking scores for d_3 and d_4, whose subtopics had been covered by the preceding documents, were suppressed. The phenomenon clearly indicates MDP-DIV can effectively capture the user perceived utility of information in its state, and utilize it for generating diverse rankings.

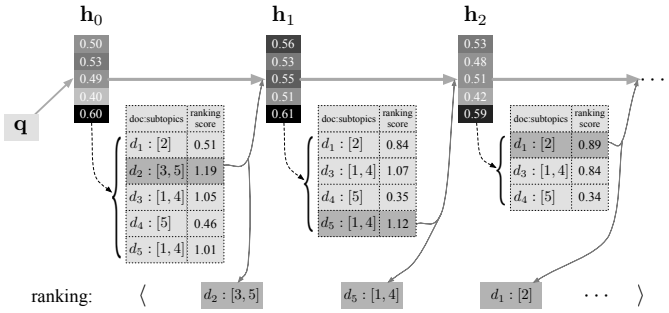

Figure 3: The online ranking process for query number 93.

5.3.2 Effects of using immediate rewards in training. One advantage of MDP-DIV is that it has the ability of utilizing the immediate rewards as the supervision in training, which makes the training more effective and efficient. We tried to verify the effectiveness and efficiency of using the immediate rewards in the training phase. Specifically, we modified the training Algorithm 1 so that the model parameters were updated only at the end of an episode (i.e., setting the iteration variable t in the line 5 of Algorithm 1 starts from M). In this way, the modified algorithm only utilizes the long term return of the whole episode for training, denoted as "MDP-DIV(ReturnOnly)". Figure 4 shows the performance curves of MDP-DIV(α-DCG) and MDP-DIV(ReturnOnly) trained with α-DCG, on the test data of one trail in the cross validation. The performances of other baseline methods on the same cross validation trail are also shown in the figure.

From the results, we can see that MDP-DIV(α-DCG) outperformed the MDP-DIV(ReturnOnly) in terms of both convergency rate and the converged performances. The result indicates that utilizing the immediate rewards in MDP-DIV(α-DCG) leads to an effective and efficient training algorithm. Note that in contrast, most existing learning approaches to diverse ranking, including R-LTR, PAMM, and NTN-DIV, can only utilize the accumulated information on the whole ranking as supervision in their training phase. For example, R-LTR uses the likelihood of the whole document rankings, and PAMM uses the predefined evaluation measure calculated based on the whole ranking. The experimental results showed one reason why MDP-DIV(α-DCG) can outperform these baselines.

We also noticed that the converged MDP-DIV(ReturnOnly) model still outperformed the baseline methods including SVM-DIV, R-LTR, PAMM, and NTN-DIV, indicating that modeling the user's dynamic state on the perceived utility with MDP is effective.

[6]Here we consider α-NDCG and α-DCG as "one" measure as the only difference between them is the normalization factor.

[7]In the online ranking, the selection of actions can be implemented as directly based on the ranking scores instead of based on the probabilities $\pi(a_t|s_t)$.

Table 1: Performance comparison of all methods on TREC web track datasets.

Method	α-NDCG@5	α-NDCG@10	S-recall@5	S-recall@10	ERR-IA@5	ERR-IA@10
MMR	0.2753	0.2979	0.4388	0.5151	0.2005	0.2309
xQuAD	0.3165	0.3941	0.4933	0.6043	0.2314	0.2890
PM-2	0.3047	0.3730	0.4910	0.6012	0.2298	0.2814
SVM-DIV	0.3030	0.3699	0.5122	0.6230	0.2268	0.2726
R-LTR	0.3498	0.4132	0.5397	0.6511	0.2521	0.3011
PAMM(α-NDCG)	0.3712	0.4327	0.5561	0.6612	0.2619	0.3029
NTN-DIV(α-NDCG)	0.3962	0.4577	0.5817	0.6872	0.2773	0.3285
MDP-DIV(S-recall)	0.4156	0.4734	**0.6123**	**0.7155**	0.2963	0.3477
MDP-DIV(α-DCG)	**0.4189**	**0.4762**	0.6102	0.7117	**0.2988**	**0.3494**

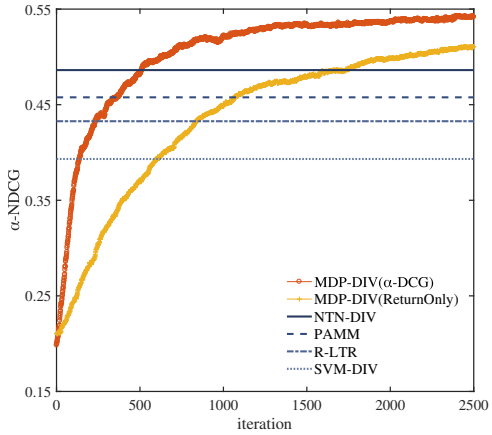

Figure 4: The performance curves on the test data for MDP-DIV(α-DCG), and the modified MDP-DIV(α-DCG) in which the training only involves the long-term returns. The performances of other baselines are shown as horizontal lines.

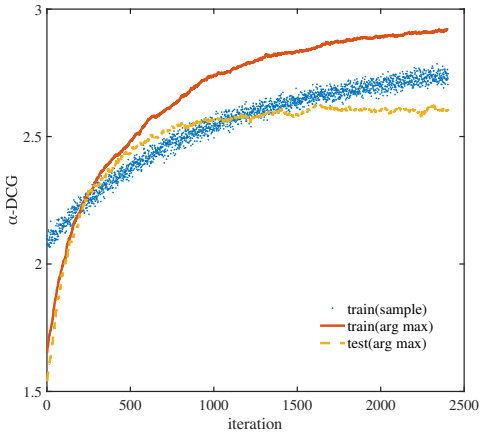

Figure 5: The performance curves in terms of α-DCG on the training data ("train(arg max)") and the test data ("test(arg max)"). The average performances of the sampled rankings over all training queries are also shown ("train(sample)").

5.3.3 Analysis of convergence and online ranking criterion. We conducted experiments to test whether the ranking accuracy in terms of the evaluation measure can be continuously improved, as the training of MDP-DIV goes on.

Specifically, we tested the MDP-DIV(α-DCG) models generated at each of the training iteration in one trail of the cross validation. The performances in terms of α-DCG at the last position of the whole document ranking is reported. For each model, the average performances over all of the training queries (or the testing queries) are reported. Figure 5 shows the performance curves on the training data (solid red line and denoted as "train(arg max)") and on the test data (dashed yellow line and denoted as "test(arg max)"). For these two curves, the document rankings for the queries are generated by the online ranking Algorithm 3. The document rankings can also be generated through sampling during the training, via the episode sampling Algorithm 2. The average performances of the sampled rankings for all the training queries are also shown in the figure (blue dots and denoted as "train(sample)").

From the results shown in Figure 5, we can see that on both of the training set and test set, the ranking accuracies of MDP-DIV(α-DCG) steadily improves, as the training goes on. The experimental results also showed that the ranking accuracies of the sampled rankings (by Algorithm 2) has an obvious trend of steadily improving with some random noise, as the training goes on.

Comparing the sampled rankings ("train(sample)") and the ranking generated by the online ranking algorithm ("train(arg max)"), we can see that at the beginning of the training phase, the sampled rankings can achieve better α-DCG values than the rankings generated by the online ranking algorithm, on the basis of the training queries. As the training went on and after about 200 iterations, the online ranking algorithm outperformed the sampling method, and the trend remains to the end of the training. The phenomenon was repeated in other experiments. We analyzed the reasons. The online ranking algorithm (Algorithm 3) fully trusts the learned ranking model when generating the document ranking, i.e., $\hat{a} \leftarrow \arg\max_{a \in A} \pi(a|s; \boldsymbol{\Theta})$. In contrast, the sampled rankings are generated according to the same ranking model while with some randomness. At the beginning of the training phase, the model parameters are far from their optimal values. In many cases, fully trusting the policy leads to bad decisions and generating rankings with low performances. The sampling method, in contrast, may make better decisions due to the random natural of sampling. As

the training goes on, the model parameters gradually converge to nearly optimal values. Fully trusting the learned policy has the advantages of achieving stable and (nearly) optimal decisions in most cases. The sampling method, however, hurts from unstable results due to the random noise. The results clearly indicate that, fully trusting the learned model (as that of in Algorithm 3) in the online ranking phase is a good criterion, given the model is well trained.

6 CONCLUSION AND FUTURE WORK

In this paper we have proposed a novel approach to learning diverse ranking model for search result diversification, referred to as MDP-DIV. In contrast to existing methods, MDP-DIV explicitly models the dynamic utility the search users perceived during the browsing of the ranking result. The dynamic utility is modeled with a continuous state MDP and the model parameters are estimated with reinforcement learning. MDP-DIV offers several advantages: no need for handcrafting ranking features, optimizing diversity evaluation measures in training, utilizing both immediate rewards and long-term returns as supervision, and high accuracy in ranking. Experimental results based on the TREC benchmark datasets show that MDP-DIV can significantly outperform the baseline methods.

7 ACKNOWLEDGEMENTS

The work was funded by the National Key R&D Program of China under Grant No. 2016QY02D0405, 973 Program of China under Grant No. 2014CB340401 and 2012CB316303, the National Natural Science Foundation of China (NSFC) under Grants No. 61232010, 61472401, 61433014, 61425016, and 61203298, the Key Research Program of the CAS under Grant No. KGZD-EW-T03-2, and the Youth Innovation Promotion Association CAS under Grants No. 20144310 and 2016102.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying Search Results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*. ACM, New York, NY, USA, 5–14.

[2] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*. ACM, New York, NY, USA, 335–336.

[3] Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. 2011. Intent-based Diversification of Web Search Results: Metrics and Algorithms. *Inf. Retr.* 14, 6 (Dec. 2011), 572–592. DOI: http://dx.doi.org/10.1007/s10791-011-9167-7

[4] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and Diversity in Information Retrieval Evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 659–666.

[5] Van Dang and W. Bruce Croft. 2012. Diversity by Proportionality: An Election-based Approach to Search Result Diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 65–74.

[6] Sreenivas Gollapudi and Aneesh Sharma. 2009. An Axiomatic Approach for Result Diversification. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. 381–390.

[7] Shengbo Guo and Scott Sanner. 2010. Probabilistic Latent Maximal Marginal Relevance. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. ACM, New York, NY, USA, 833–834.

[8] Jiyin He, Vera Hollink, and Arjen de Vries. 2012. Combining Implicit and Explicit Topic Representations for Result Diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 851–860.

[9] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search Result Diversification Based on Hierarchical Intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, New York, NY, USA, 63–72.

[10] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. 1188–1196. http://jmlr.org/proceedings/papers/v32/le14.html

[11] Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. 2009. Enhancing Diversity, Coverage and Balance for Summarization Through Structure Learning. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, New York, NY, USA, 71–80.

[12] Zhongqi Lu and Qiang Yang. 2016. Partially Observable Markov Decision Process for Recommender Systems. *CoRR* abs/1608.07793 (2016).

[13] Jiyun Luo, Sicong Zhang, and Hui Yang. 2014. Win-win Search: Dual-agent Stochastic Game in Session Search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*. ACM, New York, NY, USA, 587–596.

[14] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, NY, USA.

[15] Lilyana Mihalkova and Raymond Mooney. 2009. Learning to Disambiguate Search Queries from Short Sessions. In *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, Vol. 5782. Springer.

[16] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent Models of Visual Attention. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*. MIT Press, Cambridge, MA, USA, 2204–2212.

[17] Martin L. Puterman. 2008. *Markov Decision Processes*. John Wiley & Sons, Inc.

[18] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning Diverse Rankings with Multi-armed Bandits. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*. ACM, New York, NY, USA, 784–791.

[19] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting Query Reformulations for Web Search Result Diversification. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. 881–890.

[20] Rodrygo L. T. Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. 2010. *Explicit Search Result Diversification through Sub-queries*. Springer Berlin Heidelberg, Berlin, Heidelberg, 87–99.

[21] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1265–1295.

[22] Richard S. Sutton and Andrew G. Barto. 2016. *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[23] Xiaojie Wang, Zhicheng Dou, Tetsuya Sakai, and Ji-Rong Wen. 2016. Evaluating Search Result Diversity Using Intent Hierarchies. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 415–424.

[24] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. 113–122.

[25] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2016. Modeling Document Novelty with Neural Tensor Network for Search Result Diversification. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. 395–404.

[26] Jun Xu, Long Xia, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2017. Directly Optimize Diversity Evaluation Measures: A New Approach to Search Result Diversification. *ACM Trans. Intell. Syst. Technol.* 8, 3, Article 41 (Jan. 2017), 26 pages.

[27] Yisong Yue and Thorsten Joachims. 2008. Predicting Diverse Subsets Using Structural SVMs. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*. ACM, New York, NY, USA, 1224–1231.

[28] Yisong Yue and Thorsten Joachims. 2009. Interactively Optimizing Information Retrieval Systems As a Dueling Bandits Problem. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 1201–1208.

[29] Cheng Xiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR '03)*. 10–17.

[30] Sicong Zhang, Jiyun Luo, and Hui Yang. 2014. A POMDP Model for Content-free Document Re-ranking. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*. ACM, New York, NY, USA, 1139–1142.

[31] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for Search Result Diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14)*. ACM, New York, NY, USA, 293–302.