Modeling Document Novelty with Neural Tensor Network for Search Result Diversification

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

ABSTRACT

Search result diversification has attracted considerable attention as a means to tackle the ambiguous or multi-faceted information needs of users. One of the key problems in search result diversification is *novelty*, that is, how to measure the novelty of a candidate document with respect to other documents. In the heuristic approaches, the predefined document similarity functions are directly utilized for defining the novelty. In the learning approaches, the novelty is characterized based on a set of handcrafted features. Both the similarity functions and the features are difficult to manually design in real world due to the complexity of modeling the document novelty. In this paper, we propose to model the novelty of a document with a neural tensor network. Instead of manually defining the similarity functions or features, the new method automatically learns a nonlinear novelty function based on the preliminary representation of the candidate document and other documents. New diverse learning to rank models can be derived under the relational learning to rank framework. To determine the model parameters, loss functions are constructed and optimized with stochastic gradient descent. Extensive experiments on three public TREC datasets show that the new derived algorithms can significantly outperform the baselines, including the state-of-the-art relational learning to rank models.

Keywords

search result diversification; neural tensor network; relational learning to rank

1. INTRODUCTION

In web search, it has been widely observed that a large fraction of queries are ambiguous or multi-faceted. Search result diversification has been proposed as a way to tackle this problem and diverse ranking is one of the central problems. The goal of diverse ranking is 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. Thus, how to measure the novelty of a candidate document with respect to other documents becomes a key problem in the designing of the diverse ranking models.

Methods for search result diversification can be categorized into heuristic approaches and learning approaches. The heuristic approaches construct diverse rankings with heuristic rules [3, 8, 14, 24, 25, 26]. As a representative model, the maximal marginal relevance (MMR) [3] formulates the construction of a diverse ranking as a process of sequential document selection. At each iteration the document with the highest marginal relevance is selected. The marginal relevance consists of the relevance score and novelty score. The novelty score is calculated based on a predefined document similarity function. Thus, the selection of the document similarity function becomes a critical issue for MMR. Different choices of the similarity functions result in different ranking lists. Usually it is difficult to define an appropriate similarity function in a real application.

Recently, machine learning models have been proposed and applied to the task of search result diversification [17, 20, 23, 28, 32]. The basic idea is to automatically learn a diverse ranking model from the labeled training data. Relational learning to rank is one of the representative framework in this field. In relational learning to rank, the novelty of a document with respect to the previously selected documents is encoded as a set of handcrafted novelty features. Several algorithms have been developed under the framework and state-of-the-art performances have been achieved [28, 32]. However, it is still an unsolved problem to define a set of novelty features which can effectively capture the complex document relationship. Unlike the designing of relevance features in conventional learning to rank, it is much more difficult to extract novelty features for search result diversification. Currently, a very limited number of novelty features can be utilized when constructing a diverse ranking model. For example, in R-LTR [32] and PAMM [28], the novelty of a document is characterized with only seven novelty features. Most of the features are based on the cosine similarities of two documents represented with tf-idf vectors or topic vectors. Thus, it is very difficult, if not impossible, for users to handcraft an optimal set of novelty features for search result diversification.

To address above problems and inspired by the neural models for relation classification [27], we propose to model the document novelty for search result diversification using a neural tensor network (NTN). Unlike existing methods

*Corresponding author: Jun Xu.

SIGIR '16, July 17–21, 2016, Pisa, Italy.

DOI: http://dx.doi.org/10.1145/2911451.2911498

which manually define the document similarity functions or novelty features, the method automatically learns a nonlinear document novelty function from the training data. It first generates the novelty signals with a nonlinear tensor layer, through interacting the candidate document with other documents. Then, a max-pooling operation is applied to select the most effective novelty signals. Finally, the selected signals are combined linearly to form the final document novelty score.

New diverse ranking models, then, can be proposed under the relational learning to rank framework. The marginal relevance in relational learning to rank, which is used for selecting the best document at each step, is calculated as a sum of the query-document relevance score and document novelty score. Modeling the document novelty score with the proposed neural tensor network, we can achieve new diverse ranking models. On the basis of existing relational learning to rank algorithms of R-LTR and PAMM, two new loss functions are constructed and optimized, achieving two novel diverse ranking algorithms of R-LTR-NTN and PAMM-NTN.

To evaluate the effectiveness of the proposed algorithms, we conducted extensive experiments on three public TREC benchmark datasets. The experimental results showed that our proposed algorithms, including R-LTR-NTN and PAMM-NTN, can significantly outperform the state-of-the-art baselines including heuristic approaches of MMR, and learning approaches of SVM-DIV [29], R-LTR, and PAMM. Analysis showed that the proposed approaches achieved better results through learning better document dissimilarities in terms of distinguishing the documents with different subtopics. Thus, the proposed algorithms have the ability to improve the queries with high ambiguity.

Contributions of the paper include: 1) We proposed to model the document novelty with a neural tensor network, which enables us to get rid of the manually defined similarity functions or handcrafted novelty features in search result diversification; 2) Based on the new document novelty model, two diverse ranking algorithms were derived under the framework of relational learning to rank; 3) The effectiveness of the proposed algorithms were verified based on public benchmark datasets.

The rest of the paper is organized as follows. After a summary of related work in Section 2, we present the neural tensor network model for measuring document novelty in Section 3. Section 4 presents the two derived diverse ranking algorithms under the relational learning to rank framework. Experimental results and discussions are given in Section 5. Section 6 concludes the paper and gives future directions.

2. RELATED WORK

This paper concerns about the ranking models for search result diversification. Existing methods can be categorized into heuristic approaches and learning approaches. One of the central problems in both of these two approaches is novelty, that is, how to model the novelty information of a document with respect to other documents.

2.1 Heuristic approaches

It is a common practice to use heuristic rules to construct a diverse ranking list in search. Usually, the rules are created based on the observation that in diverse ranking a document's novelty depends on not only the document itself but also the documents ranked in previous positions. Carbonell and Goldstein [3] proposed the maximal marginal relevance criterion to guide the design of diverse ranking models. The criterion is implemented with a process of iteratively selecting the documents from the candidate document set. 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 another word, novelty. The marginal relevance score is then updated in the next iteration as the number of documents in the result set increases by one. A number of methods have been developed under the criterion. PM-2 [8] treats the problem of finding a diverse search result as finding a proportional representation for the document ranking. xQuAD [26] directly models different aspects underlying the original query in the form of sub-queries, and estimates the relevance of the retrieved documents to each identified subquery. Hu et al. [14] proposed a diversification framework that explicitly leverages the hierarchical intents of queries and selects the documents that maximize diversity in the hierarchical structure. See also [2, 4, 10, 11, 12, 22]

All of these heuristic approaches rely on a predefined document similarity (or distance) function to measure the novelty of a document. Thus, the selection of the similarity function is critical for the ranking performances. Usually it is hard to design an optimal similarity function for a specific task. In this paper, we focus on the learning approaches to estimate the novelty scores of documents.

2.2 Learning approaches

Machine learning techniques have been applied to construct ranking models for search result diversification. In these approaches, the relevance features and novelty features are extracted for characterizing the relevance and novelty information of a document, respectively. The ranking score is usually a linear combination of these features and the parameters can be automatically estimated from the training data. Some promising results have been obtained. For example, Zhu et al. [32] proposed the relational learning to rank framework in which the diverse ranking is constructed with a process of sequential document selection. The training of a relational learning to rank model thus amounts to optimizing the object function based on the ground-truth rankings. With different definitions of the object functions and optimization techniques, different diverse ranking algorithms have been derived [28, 32]. Radlinski et al. [23] proposed online learning algorithms that directly learn a diverse ranking of documents based on users' clicking behaviors. More works please refer to [17, 20, 30].

Most learning approaches depend on a set of handcrafted novelty features to represent the novelty of a document. Construction of such features is usually difficult and time consuming in real applications. In real world, we have a very limited number of novelty features, which greatly limits the usability of these diverse ranking models. In this paper, we propose to automatically learn the novelty with a neural tensor network and enhance the usability of the diverse ranking algorithms.

3. MODELING DOCUMENT NOVELTY WITH NEURAL TENSOR NETWORK

Inspired by the neural models for relation classification, in

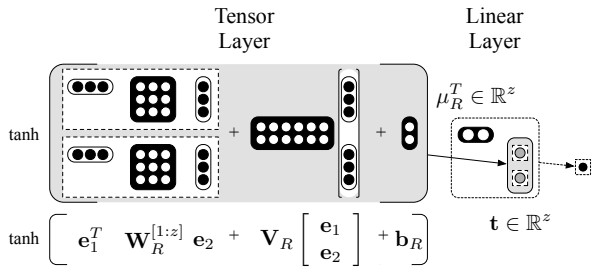

Figure 1: Visualization of the neural tensor network for relation classification. Each dashed box represents one slice of the tensor, in this case there are $z = 2$ slices.

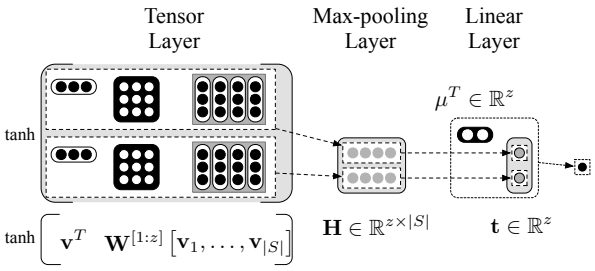

Figure 2: Visualization of the neural tensor network for modeling document novelty ($z = 2$).

this paper we propose to use neural tensor network to model the novelty of a document w.r.t. a set of other documents.

3.1 Neural tensor network

In deep learning literature, neural tensor networks (NTN) is originally proposed to reason the relationship between two entities in knowledge graph [27]. Given two entities $(\mathbf{e}_1, \mathbf{e}_2)$ represented with $l_\mathbf{e}$ dimensional features, the goal of NTN is to predict whether they have a certain relationship R. Specifically, NTN computes a score of how likely it is that these two entities are in certain relationship R by the following function:

$$g(\mathbf{e}_1, R, \mathbf{e}_2) = \mu_R^T \tanh\left(\mathbf{e}_1^T \mathbf{W}_R^{[1:z]} \mathbf{e}_2 + \mathbf{V}_R \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix} + \mathbf{b}_R\right),$$

where $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^{l_\mathbf{e}}$ are the vector representations of two entities, $\mathbf{W}_R^{[1:z]} \in \mathbb{R}^{l_\mathbf{e} \times l_\mathbf{e} \times z}$ is a tensor and the bilinear tensor product $\mathbf{e}_1^T \mathbf{W}_R^{[1:z]} \mathbf{e}_2$ results in a vector $\mathbf{h} \in \mathbb{R}^z$, where each entry of $\mathbf{h}$ is computed by one slice i $(i = 1, \cdots, z)$ of the tensor: $h_i = \mathbf{e}_1^T \mathbf{W}_R^{[i]} \mathbf{e}_2$. The other parameters for relation R are the standard form of a neural network: $\mathbf{V}_R \in \mathbb{R}^{z \times 2l_\mathbf{e}}$, $\mu_R \in \mathbb{R}^z$, and $\mathbf{b}_R \in \mathbb{R}^z$. Figure 1 illustrates the neural tensor network with two slices for entity relationship reasoning.

3.2 Modeling document novelty with neural tensor network

Intuitively, the neural tensor networks model the relationships between two entities with a bilinear tensor product. The idea can be naturally extended to model the novelty relation of a document with respect to other documents for search result diversification. That is, we can represent the novelty information of a candidate document as a bilinear tensor product of the document and other documents, as shown in Figure 2.

More specifically, suppose that we are given a set of M documents $X = \{d_j\}_{j=1}^M$, where each document d_j can be characterized with its preliminary representation $\mathbf{v}_j \in \mathbb{R}^{l_\mathbf{v}}$, e.g., the topic distribution [9, 13] of d_j or the document vector generated with a doc2vec [15] model. Given a candidate document $d \in X$ with its preliminary presentation $\mathbf{v}$, and a set of documents $S \subseteq X \setminus \{d\}$ with their preliminary representations $\{\mathbf{v}_1, \cdots, \mathbf{v}_{|S|}\}$, the novelty score of d with respect to the documents in S can be defined as a neural tensor network with z hidden slices:

$$g_n(\mathbf{v}, S) = \mu^T \max\left\{\tanh\left(\mathbf{v}^T \mathbf{W}^{[1:z]} \begin{bmatrix} \mathbf{v}_1, \ldots, \mathbf{v}_{|S|} \end{bmatrix}\right)\right\},$$

where each column in matrix $\begin{bmatrix} \mathbf{v}_1, \ldots, \mathbf{v}_{|S|} \end{bmatrix} \in \mathbb{R}^{l_\mathbf{v} \times |S|}$ stands for the preliminary representation vector of the corresponding document in S, $\mathbf{W}^{[1:z]} \in \mathbb{R}^{l_\mathbf{v} \times l_\mathbf{v} \times z}$ is a tensor, and $\mu \in \mathbb{R}^z$ the weights correspond to the slices of the tensor. As shown in Figure 2, the neural tensor network consists of a tensor layer, a max-pooling layer, and a linear layer.

Tensor Layer: The tensor layer takes the preliminary representations of the documents as inputs. The interactions between the document d and documents in S are represented as a bilinear product followed by a nonlinear operation:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^T \\ \vdots \\ \mathbf{h}_z^T \end{bmatrix} = \begin{bmatrix} \tanh\left(\mathbf{v}^T \mathbf{W}^{[1]} \begin{bmatrix} \mathbf{v}_1, \ldots, \mathbf{v}_{|S|} \end{bmatrix}\right) \\ \vdots \\ \tanh\left(\mathbf{v}^T \mathbf{W}^{[z]} \begin{bmatrix} \mathbf{v}_1, \ldots, \mathbf{v}_{|S|} \end{bmatrix}\right) \end{bmatrix}, \quad (1)$$

where $\mathbf{h}_i \in \mathbb{R}^{|S|}$ is computed by one slice of the tensor.

Compared with the original neural tensor network in Section 3.1, the tensor in Equation (1) models the relationship between one document and *multiple documents* simultaneously. Thus, the output of Equation (1) is a $z \times |S|$ matrix rather than a z-dimensional vector. Also, since the number of documents in S varies in different document selection iterations, the term $\mathbf{V}_R \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}$ in the original tensor neural network is ignored. Moreover, in ranking we cares about the order of the documents rather than the ranking scores. Thus, the bias term $\mathbf{b}_R$ is also ignored.

Max-pooling Layer: In the max-pooling layer, the matrix outputted by the tensor layer is mapped to a z-dimensional vector with the max operation:

$$\mathbf{t} = \left[\max(\mathbf{h}_1^T), \cdots, \max(\mathbf{h}_z^T)\right]^T. \quad (2)$$

Intuitively, the pooling layer aggregates individual novelty signal learned at each tensor layer $\mathbf{h}_i^T$. Max-pooling extracts the most significant signals among them. Thus, vector $\mathbf{t}$ can be considered as a the z-dimensional novelty features and each dimension is defined by one slice of the tensor.

Linear Layer: Finally, the novelty score of the document is calculated as a linear combination of the novelty signals outputted by the max-pooling layer: $\mu^T \mathbf{t}$, where μ is an z-dimensional parameter vector.

4. DIVERSE RANKING ALGORITHMS BASED ON NEURAL TENSOR NETWORK

New diverse ranking algorithms can be derived based on the proposed neural tensor network for modeling document novelty. In this paper, we propose two algorithms under the framework of relational learning to rank.

Algorithm 1 Ranking via maximizing marginal relevance

Input: documents X and novelty features R
Output: ranking of documents Y
1: $S_0 \leftarrow$ empty set
2: **for** $r = 1, \cdots, M$ **do**
3: $\quad Y(r) \leftarrow \arg\max_{j:\mathbf{x}_j \in X \setminus S_{r-1}} f(\mathbf{x}_j, R_j, S_{r-1})$
4: $\quad S_r \leftarrow S_{r-1} \cup \{\mathbf{x}_{Y(r)}\}$
5: **end for**
6: **return** Y

4.1 Relational learning to rank

The relational learning to rank framework [32] formalizes the ranking of documents as a process of sequential document selection and defines the marginal relevance as linear combination of the relevance score and the novelty score. Formally, let $X = \{d_1, \cdots, d_M\}$ denotes the set of documents retrieved by a query q. For each query-document pair (q, d_i), relevance feature vector $\mathbf{x}_i \in \mathbb{R}^{l_\mathbf{x}}$ is extracted. Let $R \in \mathbb{R}^{M \times M \times K}$ denotes a 3-way tensor representing relationships between the documents, where R_{ijk} stands for the k-th feature of relationship between documents d_i and d_j. Assuming that a set of documents S have been selected in the previous iterations, the marginal relevance of the i-th candidate document with respect to S, denoted as $f(\mathbf{x}_i, R_i, S)$, is then defined as the combination of the relevance score and the novelty score:

$$f(\mathbf{x}_i, R_i, S) = \omega_r^T \mathbf{x}_i + \omega_n^T h_S(R_i), \forall \mathbf{x}_i \in X \setminus S, \qquad (3)$$

where $\omega_r^T \mathbf{x}_i$ stands for the relevance score and ω_r is the relevance weight vector, $\omega_n^T h_S(R_i)$ stands for the novelty score of the document with respect to S and ω_n is the diversity weight vector, R_i stands for the *matrix* of relationships between document $\mathbf{x}_i$ and other documents, and $h_S(R_i)$ stands for the aggregation function on R_i which aggregates the matrix R_i into a novelty feature vector. Usually, h_S can be one of the operations of max, min, or average.

According to the maximal marginal relevance criterion, sequential document selection process can be used to create a diverse ranking, as shown in Algorithm 1. The algorithm initializes S_0 as an empty set, and then iteratively selects the documents from the candidate set. At iteration r $(r = 1, 2, \cdots, M)$, the document with the maximal marginal relevance score $f(\mathbf{x}_j, R_j, S_{r-1})$ is selected and ranked at position r. At the same time, the selected document is inserted into S_{r-1}.

Given a set of training instances which consist of queries, documents, and their relevance labels, the model parameters can be learned from the training data. The process amounts to optimizing an objective function based on the training data. Different definitions of the objective functions and optimization techniques lead to different relational learning to rank algorithms. For example, in algorithm R-LTR [32], the likelihood of the training queries is maximized using stochastic gradient descent. In algorithm PAMM [28], the loss function upper bounding the diversity evaluation measure is constructed and optimized with structured Perceptron.

Relational learning to rank models depend on a set of handcrafted features for characterizing the novelty of a document. However, how to design the features that can effectively capture the complex document relationship is still an unsolved problem. Unlike the conventional learning to rank in which a large number effective relevance features have been developed [21], it is much harder to find novelty features for search result diversification. As a result, the relational learning to rank algorithms of R-LTR and PAMM utilized only seven features in their experiments, as have listed in Table 1. We can see that most of these features are calculated based on the predefined similarities of two documents (represented as tf-idf vectors or topic distributions), and respectively applied to the document fields of title, body, and anchor.

Table 1: Novelty features used in R-LTR.

Name	Explanation
Subtopic diversity	document distance based on PLSA [13]
Text diversity	one minus cosine similarity of the tf-idf vectors on body text
Title diversity	text novelty feature based on title
Anchor text diversity	text novelty feature based on anchor
ODP-Based diversity	categorical distance based on ODP[1] taxomony
Link-based diversity	link similarity based on inlink/outlink
URL-based diversity	whether the two URLs belong to the same domain/site

In real world applications, the performances of the ranking algorithms heavily depend on the effectiveness of these handcrafted features and different ranking tasks need different features. It is necessary to develop a method that can learn the document novelty automatically and release people from the handcrafted novelty features.

4.2 Relational learning to rank algorithms based on neural tensor network

In this subsection, based on the technique of modeling the document novelty with neural tensor network, we develop two new relational learning to rank algorithms that can learn the document novelty function automatically.

4.2.1 The ranking model

Following the notations used in Section 3.2 and Section 4.1, let $X = \{d_1, \cdots, d_M\}$ denotes the set of documents retrieved by a query q. Each query-document pair (q, d) is represented with the relevance feature vector $\mathbf{x} \in \mathbb{R}^{l_\mathbf{x}}$. Each document $d \in X$ is characterized with its preliminary representation vector $\mathbf{v} \in \mathbb{R}^{l_\mathbf{v}}$. Assuming that at one iteration of the sequential document selection, a set of documents S have been selected. We define the marginal relevance score of a candidate document d as:

$$\begin{aligned} f(d, S) =& g_r(\mathbf{x}) + g_n(\mathbf{v}, S) \qquad (4) \\ =& \omega^T \mathbf{x} + \mu^T \max\left\{\tanh\left(\mathbf{v}^T \mathbf{W}^{[1:z]} \left[\mathbf{v}_1, \ldots, \mathbf{v}_{|S|}\right]\right)\right\}, \end{aligned}$$

where $g_r(\mathbf{x})$ is the relevance of d w.r.t. query q, which is further defined as a linear combination of the relevance features; $g_n(\mathbf{v}, S)$ is the novelty of d w.r.t. the documents in S, which is further defined as a neural tensor network, as have been shown in Section 3.2. The model parameters ω, μ, and $\mathbf{W}^{[1:z]}$ can be learned with the training data.

In the online ranking, a diverse ranking can created with the sequential document selection process, similar to the procedure shown in Algorithm 1.

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

The main advantage of using neural tensor network to model document novelty is that the tensor can relate the candidate document and the selected documents multiplicatively, instead of only through a predefined similarity function (as that of in heuristic approaches) or through a linear combination of novelty features (as that of in learning approaches and shown in Equation (3)). Intuitively, the model can be explained that each slice of the tensor is responsible for one aspect or subtopic of a query. Each tensor slice settles the diversity relationship between the candidate document and the selected documents set differently. Thus, with multiple tensor slices, the model calculates the novelty scores based on multiple diversity aspects.

4.2.2 General loss function

The parameters of the ranking model can be determined with supervised learning methods, which amounts to optimizing the objective function built upon the labeled training data.

In training procedure, given the labeled data with N queries as: $(X^{(1)}, J^{(1)}), (X^{(2)}, J^{(2)}), \cdots, (X^{(N)}, J^{(N)})$, where $X^{(n)} = \{d_j^{(n)}\}_{j=1}^{M^{(n)}}$, where $M^{(n)}$ denotes the number of documents related with the n-th query. Let $\mathbf{x}_j^{(n)} \in \mathbb{R}^{l_\mathbf{x}}$ denote the relevance feature vector for the n-th query and document $d_j^{(n)}$, $\mathbf{v}_j^{(n)} \in \mathbb{R}^{l_\mathbf{v}}$ the preliminary representation of document $d_j^{(n)}$, and $J^{(n)}$ the human labels on documents which is in the form of a binary matrix. $J_{js}^{(n)} = 1$ if document $d_j^{(n)}$ contains the s-th subtopic of the query and 0 otherwise[2]. The learning process amounts to minimizing the total loss with respect to the given training data:

$$\min_{f \in \mathcal{F}} \sum_{n=1}^{N} \ell\left(\pi\left(X^{(n)}, f\right), J^{(n)}\right),$$

where $\pi\left(X^{(n)}, f\right)$ denotes the ranking generated by the ranking model f in Equation (4), for the documents in $X^{(n)}$. The generated ranking π is then compared with the human labels $J^{(n)}$ by the loss function ℓ. Intuitively, the learning process can be interpreted as finding an optimal ranking model f from some functional space $\mathcal{F}$ so that for each training query the difference between the generated permutation π and the human labels J is minimal.

Different objective functions and optimization techniques lead to different algorithms. In this section, based on the relational learning to rank algorithms of R-LTR [32] and PAMM [28], we construct two novel algorithms in which the document novelty is modeled with a neural tensor network, referred to as R-LTR-NTN and PAMM-NTN, respectively.

4.2.3 R-LTR-NTN

Based on the loss function defined for R-LTR [32], we derive the loss function of R-LTR-NTN, which is a negative logarithm likelihood of the training queries:

$$L^{\text{R-LTR-NTN}}(f) = -\sum_{n=1}^{N} \log \Pr\left(Y^{(n)} | X^{(n)}\right),$$

where $Y^{(n)}$ is the ground-truth ranking generated from the human label $J^{(n)}$. For any query, the probability $\Pr(Y|X)$ can be further defined as

$$\begin{aligned}\Pr(Y|X) =& \Pr(d_{Y(1)} d_{Y(2)} \cdots d_{Y(M)} | X) \\ =& \prod_{r=1}^{M-1} \Pr(d_{Y(r)} | X, S_{r-1}) \\ =& \prod_{r=1}^{M-1} \frac{\exp\{f(d_{Y(r)}, S_{r-1})\}}{\sum_{k=r}^{M} \exp\{f(d_{Y(k)}, S_{r-1})\}},\end{aligned} \tag{5}$$

where $Y(r)$ denotes the index of the document ranked at the r-th position in Y, $S_{r-1} = \{d_{Y(k)}\}_{k=1}^{r-1}$ is the documents ranked at the top $r-1$ positions in Y, $f(d_{Y(r)}, S_{r-1})$ is the marginal relevance score of document $d_{Y(r)}$ w.r.t. the selected documents in S_{r-1}, as defined in Equation (4), and S_0 is an empty set.

Stochastic gradient descent is adopted to conduct the optimization. Given a query q, the retrieved documents $X = \{d_j\}_{j=1}^{M}$, and the ranking Y generated by the ground-truth labels, the gradient of the model parameters can be written as

$$\nabla\omega = \sum_{r=1}^{M-1} \left\{ \frac{\sum_{k=r}^{M} \left\{\exp\left\{f\left(d_{Y(k)}, S_{r-1}\right)\right\} \mathbf{x}_{Y(k)}\right\}}{\sum_{k=r}^{M} \exp\{f(d_{Y(k)}, S_{r-1})\}} - \mathbf{x}_{Y(r)} \right\}, \tag{6}$$

$$\nabla\mu = \sum_{r=1}^{M-1} \left\{ \frac{\sum_{k=r}^{M} \left\{\exp\left\{f\left(d_{Y(k)}, S_{r-1}\right)\right\} \mathbf{t}_{Y(k)}\right\}}{\sum_{k=r}^{M} \exp\{f(d_{Y(k)}, S_{r-1})\}} - \mathbf{t}_{Y(r)} \right\}, \tag{7}$$

$$\nabla\mathbf{W}^{[i]} = \sum_{r=1}^{M-1} \left\{ \frac{\sum_{k=r}^{M} \left\{\exp\left\{f\left(d_{Y(k)}, S_{r-1}\right)\right\} \mu_i \Omega_{Y(k)}\right\}}{\sum_{k=r}^{M} \exp\{f(d_{Y(k)}, S_{r-1})\}} - \mu_i \Omega_{Y(r)} \right\}, \tag{8}$$

where $\mathbf{t}$ is defined in Equation (2), and

$$\Omega_{Y(r)} = \left\{1 - \tanh^2\left(\mathbf{v}_{Y(r)}^T \mathbf{W}^{[i]} \mathbf{v}_{\tau_i}\right)\right\} \mathbf{v}_{Y(r)} \mathbf{v}_{\tau_i}^T, \tag{9}$$

where $\Omega \in \mathbb{R}^{l_\mathbf{v} \times l_\mathbf{v}}$ and $\tau_i (1 \le \tau_i \le |S|)$ stands for the output of the max-pooling position for the i-th $(1 \le i \le z)$ tensor slice.

Algorithm 2 shows the pseudo code of the R-LTR-NTN.

Algorithm 2 The R-LTR-NTN Algorithm

Input: training data $\{(X^{(n)}, J^{(n)})\}_{n=1}^{N}$ and learning rate η
Output: model parameter $(\omega, \mu, \mathbf{W}^{[1:z]})$

```
initialize {ω, μ, W^[1:z]} ← random values in [0, 1]
repeat
    Shuffle the training data
    for n = 1, ··· , N do
        calculate ∇ω^(n), ∇μ^(n) and ∇W^[1:z](n)
        {Equation (6), Equation (7), and Equation (8)}
        ω ← ω − η × ∇ω^(n)
        μ ← μ − η × ∇μ^(n)
        W^[1:z] ← W^[1:z] − η × ∇W^[1:z](n)
    end for
until convergence
return (ω, μ, W^[1:z])
```

4.2.4 PAMM-NTN

Based on the loss function defined for PAMM [28], we derive the loss function of PAMM-NTN, which is directly

[2]In this paper we assume that all labels are binary.

defined over a diversity evaluation measure:

$$\sum_{n=1}^{N} 1 - E\left(\pi\left(X^{(n)}, f\right), J^{(n)}\right), \tag{10}$$

where $E(\cdot,\cdot) \in [0,1]$ is a diversity evaluation measure such as α-NDCG or ERR-IA etc. It can be proved that the Equation (10) is upper bounded by

$$L^{\text{PAMM-NTN}}(f) = \sum_{n=1}^{N} \sum_{\substack{Y^+ \in \mathcal{Y}^{(n)+}; \\ Y^- \in \mathcal{Y}^{(n)-}}} \left[\!\!\left[\left(\Pr(Y^+|X^{(n)}) - \Pr(Y^-|X^{(n)})\right) \right.\right. \\ \left.\left. \leq \left(E(Y^+,J^{(n)}) - E(Y^-,J^{(n)})\right) \right]\!\!\right].$$

where $\mathcal{Y}^{(n)+}$ and $\mathcal{Y}^{(n)-}$ are the sets of positive and negative rankings generated from human labels $J^{(n)}$, respectively. $[\![\cdot]\!]$ is one if the condition is satisfied otherwise zero. $\Pr(\cdot|\cdot)$ stands for the probability of the ranking, as defined in Equation (5).

Also, stochastic gradient descent is adopted to conduct the optimization. At each iteration, we are given a query q, the retrieved documents $X = \{d_j\}_{j=1}^{M}$, a positive ranking Y^+, and a negative ranking Y^-. For convenience of calculation, we resort to the optimization problem of $\max \log \frac{\Pr(Y^+|X)}{\Pr(Y^-|X)}$. Thus, the gradients of the parameters can be written as

$$\nabla\omega = \sum_{r=1}^{M-1} \left\{ \frac{\sum_{k=r}^{M} \left\{ \exp\left\{ f(d_{Y^+(k)}, S_{r-1}) \right\} \mathbf{x}_{Y^+(k)} \right\}}{\sum_{k=r}^{M} \exp\left\{ f(d_{Y^+(k)}, S_{r-1}) \right\}} \right. \\ - \frac{\sum_{k=r}^{M} \left\{ \exp\left\{ f(d_{Y^-(k)}, S_{r-1}) \right\} \mathbf{x}_{Y^-(k)} \right\}}{\sum_{k=r}^{M} \exp\left\{ f(d_{Y^-(k)}, S_{r-1}) \right\}} \\ \left. - \mathbf{x}_{Y^+(r)} + \mathbf{x}_{Y^-(r)} \right\}, \tag{11}$$

$$\nabla\mu = \sum_{r=1}^{M-1} \left\{ \frac{\sum_{k=r}^{M} \left\{ \exp\left\{ f(d_{Y^+(k)}, S_{r-1}) \right\} \mathbf{t}_{Y^+(k)} \right\}}{\sum_{k=r}^{M} \exp\left\{ f(d_{Y^+(k)}, S_{r-1}) \right\}} \right. \\ - \frac{\sum_{k=r}^{M} \left\{ \exp\left\{ f(d_{Y^-(k)}, S_{r-1}) \right\} \mathbf{t}_{Y^-(k)} \right\}}{\sum_{k=r}^{M} \exp\left\{ f(d_{Y^-(k)}, S_{r-1}) \right\}} \\ \left. - \mathbf{t}_{Y^+(r)} + \mathbf{t}_{Y^-(r)} \right\}, \tag{12}$$

$$\nabla\mathbf{W}^{[i]} = \sum_{r=1}^{M-1} \left\{ \frac{\sum_{k=r}^{M} \left\{ \exp\left\{ f(d_{Y^+(k)}, S_{r-1}) \right\} \mu_i \Omega_{Y^+(k)} \right\}}{\sum_{k=r}^{M} \exp\left\{ f(d_{Y^+(k)}, S_{r-1}) \right\}} \right. \\ - \frac{\sum_{k=r}^{M} \left\{ \exp\left\{ f(d_{Y^-(k)}, S_{r-1}) \right\} \mu_i \Omega_{Y^-(k)} \right\}}{\sum_{k=r}^{M} \exp\left\{ f(d_{Y^-(k)}, S_{r-1}) \right\}} \\ \left. - \mu_i \Omega_{Y^+(r)} + \mu_i \Omega_{Y^-(r)} \right\}, \tag{13}$$

where $\mathbf{t}$ is defined in Equation (2), and Ω is defined in Equation (9). Algorithm 3 shows the pseudo code of the PAMM-NTN algorithm.

4.2.5 Time complexities

We analyzed time complexities of R-LTR-NTN and PAMM-NTN. The learning process of R-LTR-NTN (Algorithm 2) is of order $\mathcal{O}(T \cdot N \cdot M^2 \cdot (l_{\mathbf{x}} + l_{\mathbf{v}} \cdot Z))$, where T denotes the number of iterations, N the number of queries in training data, M the maximum number of documents per training query, $l_{\mathbf{x}}$ the number of relevance features, $l_{\mathbf{v}}$ the dimensions of the preliminary document representation, and Z the number of tensor slices. The learning process of PAMM-NTN (Algorithm 3) is of order $\mathcal{O}(T \cdot N \cdot \tau^+ \cdot \tau^- \cdot M^2 \cdot (l_{\mathbf{x}} + l_{\mathbf{v}} \cdot Z))$, where τ^+ denotes the number of positive rankings per query and τ^- the number of negative rankings per query.The time complexity of online ranking prediction (Algorithm 1) is of order $\mathcal{O}(M \cdot K \cdot (l_{\mathbf{x}} + l_{\mathbf{v}} \cdot Z))$, where M is the number of candidate documents for the query and K denotes the number documents need to be ranked.

Algorithm 3 The PAMM-NTN algorithm

Input: training data $\{(X^{(n)}, J^{(n)})\}_{n=1}^{N}$,
parameter: learning rate η, diversity evaluation measure E, number of positive/negative rankings per query τ^+/τ^-.
Output: model parameter $(\omega, \mu, \mathbf{W}^{[1:z]})$

```
for n = 1 to N do
    PR^(n) ← Sample positive rankings {[28]}
    NR^(n) ← Sample negative rankings {[28]}
end for
initialize (ω, μ, W^[1:z]) ← random values in [0, 1]
repeat
    for n = 1 to N do
        for all {Y+, Y−} ∈ PR^(n) × NR^(n) do
            ΔP ← Pr(Y+|X^(n)) − Pr(Y−|X^(n))
            {Pr(Y|X) is defined in Equation (5)}
            if ΔP ≤ E(Y+,J^(n)) − E(Y−,J^(n)) then
                calculate ∇ω, ∇μ and ∇W^[1:z]
                {Equation (11), Equation (12), and Equation (13)}
                ω ← ω + η × ∇ω
                μ ← μ + η × ∇μ
                W^[1:z] ← W^[1:z] + η × ∇W^[1:z]
            end if
        end for
    end for
until convergence
return (ω, μ, W^[1:z])
```

5. EXPERIMENTS

5.1 Experimental settings

We conducted experiments to test the performances of R-LTR-NTN and PAMM-NTN using three TREC benchmark datasets for diversity task: TREC 2009 Web Track (WT2009), TREC 2010 Web Track (WT2010), and TREC 2011 Web Track (WT2011). Each dataset consists of queries, corresponding retrieved documents, and human judged labels. Each query includes several subtopics identified by the TREC assessors. The document relevance labels were made at the subtopic level and the labels are binary[3]. Statistics on the datasets are given in Table 2.

All the experiments were carried out on the ClueWeb09 Category B data collection[4], which comprises of 50 million English web documents. Porter stemming, tokenization, and stop-words removal (using the INQUERY list) were applied to the documents as preprocessing. We conducted 5-fold cross-validation experiments on the three datasets. For each dataset, we randomly split the queries into five even subsets.

[3]The graded judgements in WT2011 was treated as binary.
[4]http://boston.lti.cs.cmu.edu/data/clueweb09

Table 2: Statistics on WT2009, WT2010 and WT2011.

Dataset	#queries	#labeled docs	#subtopics per query
WT2009	50	5149	$3 \sim 8$
WT2010	48	6554	$3 \sim 7$
WT2011	50	5000	$2 \sim 6$

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.

The TREC official evaluation metrics for the diversity task were used in the experiments, including the ERR-IA [5], α-NDCG [6], and NRBP [7]. They measure the diversity of a result list by explicitly rewarding novelty and penalizing redundancy observed at every rank. Following the default settings in official TREC evaluation program, the parameters α and β in these evaluation measures are set to 0.5. We also used traditional diversity measures of Precision-IA (denoted as "Pre-IA") [1], and Subtopic Recall (denoted as "S-recall") [31]. All of the measures are computed over the top-k search results ($k = 20$).

We compared R-LTR-NTN and PAMM-NTN with several types of baselines. The baselines include three heuristic approaches to search result diversification.

MMR [3] : a heuristic approach in which the document ranking is constructed via iteratively selecting the document with the maximal marginal relevance.

xQuAD [26] : a representative heuristic approach to search result diversification which explicitly accounts for the various aspects associated to an under-specified query.

PM-2 [8] : a method of optimizing proportionality for search result diversification.

Note that these baselines require a prior relevance function to implement their diversification steps. In our experiments, ListMLE [16, 18] was chosen as the relevance function.

The baselines also include state-of-the-art learning approaches to search result diversification.

SVM-DIV [29] : a learning approach in which structural SVMs was used to optimize the subtopic coverage.

R-LTR [32] : a state-of-the-art learning approach developed in the relational learning to rank framework.

PAMM [28] : another state-of-the-art learning algorithm that directly optimizes diversity evaluation measure.

Following the practice in [32], for the baseline of R-LTR, we used the results of R-LTR$_{\text{min}}$ in which the relation function $h_S(R)$ was defined as the minimal distance of the candidate document to the selected documents.

For the baseline PAMM (and our approach PAMM-NTN), we configure them to directly optimize α-NDCG@20 because it is one of the most widely used performance measures. Thus, the baseline of PAMM is denoted as PAMM(α-NDCG). Following the practice in [28], we set the number of sampled positive rankings per query $\tau^+ = 5$ and the number of sampled negative rankings per query $\tau^- = 20$.

5.2 Relevance features and preliminary document representations

As for the relevance features, we adopted the features used in R-LTR experiments [21], including the typical weighting models (e.g., TF-IDF, BM25, LM) and term dependency model [19]. Table 3 summarized the relevance features. For all the query-document matching features, they were applied in five fields: body, anchor, title, URL, and the whole document, resulting in 5 features in total. Note that the MRF feature has two variations: ordered phrase and unordered phrase [19]. Thus the total number of MRF features becomes 10.

Table 3: Relevance features used in the experiments. Each of the first 4 features is applied to the fields of body, anchor, title, URL, and the whole documents. [32]

Name	Description	# Features
TF-IDF	The tf-idf model	5
BM25	BM25 with default parameters	5
LMIR	LMIR with Dirichlet smoothing	5
MRF [19]	MRF with ordered/unordered phrase	10
PageRank	PageRank score	1
#inlinks	number of inlinks	1
#outlinks	number of outlinks	1

The neural tensor network need preliminary representations of the documents as its inputs. In the experiments, we used the document vector generated by the topic model of probabilistic latent semantic analysis (PLSA) [13] or the deep learning model of doc2vec [15], both are trained on all of the documents in ClueWeb09 Category B data collection and the number of latent dimensions are set to 100. For training the doc2vec model, we used the distributed bag of words (DBOW) model[5]. In all of the experiments, the learning rate is set to 0.025 and the window size is set to 8.

Our approaches (R-LTR-NTN and PAMM-NTN) with the settings of using the PLSA or doc2vec as document representations are denoted with the corresponding subscripts. For example, the R-LTR-NTN that using PLSA as document representations is denoted as R-LTR-NTN$_{\text{plsa}}$. Thus, in all of the experiments, our approaches include R-LTR-NTN$_{\text{plsa}}$, R-LTR-NTN$_{\text{doc2vec}}$, PAMM-NTN(α-NDCG)$_{\text{plsa}}$, and PAMM-NTN(α-NDCG)$_{\text{doc2vec}}$. Please note in all of the experiments, PAMM-NTN was configured to direct optimize the evaluation measure of α-NDCG@20.

5.3 Experimental results

Table 4, Table 5, and Table 6 report the performances of the proposed methods and baselines in terms of 5 diversity metrics (ERR-IA@20, α-NDCG@20, NRBP@20, Pre-IA@20, and S-recall@20) on the datasets of WT2009[6], WT2010, and WT2011, respectively. Boldface indicates the highest score among all runs. For all of our approaches, the number of tensor slices z is set to 7.

From the results we can see that, on all of the three datasets and in terms of the five diversity evaluation metrics, our approaches (R-LTR-NTN$_{\text{plsa}}$, R-LTR-NTN$_{\text{doc2vec}}$, PAMM-NTN(α-NDCG)$_{\text{plsa}}$, and PAMM-NTN(α-NDCG)$_{\text{doc2vec}}$) can outperform all of the baselines. We conducted significant testing (t-test) on the improvements of our approaches over the baselines. The results indicate that the improvements of R-LTR-NTN$_{\text{plsa}}$ and R-LTR-NTN$_{\text{doc2vec}}$ over R-LTR are significant (p-value < 0.05), in terms of all of the

[5]http://radimrehurek.com/gensim/models/doc2vec.html

[6]The performances of XQuAD reported in Table 4 are different to that of reported in [26]. It may caused by the different splitting of the dataset in cross validation.

Table 4: Performance comparison of all methods for WT2009.

Method	ERR-IA@20	α-NDCG@20	NRBP@20	Pre-IA@20	S-recall@20
MMR	0.2022	0.3083	0.1715	0.0918	0.4698
xQuAD	0.2316	0.3437	0.1956	0.0984	0.4931
PM-2	0.2294	0.3369	0.1788	0.0949	0.4876
SVM-DIV	0.2408	0.3526	0.2073	0.1075	0.5101
R-LTR	0.2714	0.3964	0.2339	0.1233	0.5511
R-LTR-NTN$_{\text{plsa}}$	0.3015	0.4444	0.2563	0.1588	0.5743
R-LTR-NTN$_{\text{doc2vec}}$	0.3117	0.4503	0.2578	0.1670	**0.5910**
PAMM(α-NDCG)	0.2842	0.4271	0.2411	0.1265	0.5612
PAMM-NTN(α-NDCG)$_{\text{plsa}}$	0.3081	0.4377	**0.2642**	0.1661	0.5755
PAMM-NTN(α-NDCG)$_{\text{doc2vec}}$	**0.3135**	**0.4555**	0.2626	**0.1745**	0.5772

Table 5: Performance comparison of all methods for WT2010.

Method	ERR-IA@20	α-NDCG@20	NRBP@20	Pre-IA@20	S-recall@20
MMR	0.2735	0.4036	0.2252	0.1722	0.6444
xQuAD	0.3278	0.4445	0.2872	0.1883	0.6732
PM-2	0.3296	0.4478	0.2901	0.1885	0.6749
SVM-DIV	0.3331	0.4593	0.2934	0.1925	0.6774
R-LTR	0.3647	0.4924	0.3293	0.2042	0.6893
R-LTR-NTN$_{\text{plsa}}$	0.3876	0.5311	0.3333	0.2341	0.6912
R-LTR-NTN$_{\text{doc2vec}}$	**0.3932**	0.5376	**0.3623**	**0.2418**	0.6994
PAMM(α-NDCG)	0.3802	0.5249	0.3431	0.2111	0.6832
PAMM-NTN(α-NDCG)$_{\text{plsa}}$	0.3898	0.5379	0.3479	0.2264	0.7006
PAMM-NTN(α-NDCG)$_{\text{doc2vec}}$	0.3901	**0.5407**	0.3553	0.2386	**0.7032**

Table 6: Performance comparison of all methods for WT2011.

Method	ERR-IA@20	α-NDCG@20	NRBP@20	Pre-IA@20	S-recall@20
MMR	0.4284	0.5302	0.3913	0.3176	0.7567
xQuAD	0.4753	0.5645	0.4274	0.3299	0.7683
PM-2	0.4873	0.5786	0.4318	0.3405	0.7743
SVM-DIV	0.4898	0.5910	0.4475	0.3468	0.7750
R-LTR	0.5389	0.6297	0.4982	0.3921	0.8512
R-LTR-NTN$_{\text{plsa}}$	0.5483	0.6537	0.5050	0.4011	0.8543
R-LTR-NTN$_{\text{doc2vec}}$	0.5538	0.6555	**0.5223**	0.4125	**0.8590**
PAMM(α-NDCG)	0.5417	0.6433	0.5012	0.3955	0.8518
PAMM-NTN(α-NDCG)$_{\text{plsa}}$	0.5496	0.6469	0.5111	0.4169	0.8524
PAMM-NTN(α-NDCG)$_{\text{doc2vec}}$	**0.5554**	**0.6566**	0.5212	**0.4177**	0.8533

performance measures. The results also indicate that the improvements of PAMM-NTN(α-NDCG)$_{\text{plsa}}$ and PAMM-NTN(α-NDCG)$_{\text{doc2vec}}$ over all of the baselines are significant, in terms of all of the performance measures. The results indicate that the neural tensor network is effective for modeling the document novelty information, and thus can improve the performances.

5.4 Discussions

We conducted experiments to show the reasons that our approaches outperformed the baselines and impacts of different parameter settings, using the results of R-LTR-NTN$_{\text{plsa}}$ and R-LTR-NTN$_{\text{doc2vec}}$ on WT2009 dataset as examples.

5.4.1 Ability to learn better document dissimilarities

We found that the learned neural tensor network can help to distinguish the relevant documents in terms of different subtopics, by learning a better dissimilarity (novelty) function for documents. That is one of the reasons why our approaches can outperform the baselines.

Specifically, the dissimilarities between two documents can be calculated based on the preliminary document representations, either using the Euclidean distance or using the learned neural tensor network (the novelty score of a document w.r.t. another document). That is, given two documents represented with the preliminary presentations $\mathbf{v}_i$ and $\mathbf{v}_j$, the dissimilarity score can calculated either based on the Euclidean distance:

$$d_e(\mathbf{v}_i, \mathbf{v}_j) = \|\mathbf{v}_i - \mathbf{v}_j\|_2,$$

or based on the learned neural tensor network:

$$d_n(\mathbf{v}_i, \mathbf{v}_j) = g_n(\mathbf{v}_i, \{\mathbf{v}_j\}) = \mu^T \tanh\left(\mathbf{v}_i^T \mathbf{W}^{[1:z]} \mathbf{v}_j\right)$$

where μ and $\mathbf{W}^{[1:z]}$ are learned with the R-LTR-NTN algorithms. Here we can ignore the max operation because there is only one document $\mathbf{v}_j$ at the righthand of $\mathbf{W}^{[1:z]}$.

Suppose we are given a set of queries and the associated relevant documents. For each query, the relevant documents can be grouped into several clusters, each corresponds a subtopic of the query. Thus, all of the associated documents from all queries are grouped into different clusters, each corresponds to a subtopic. We calculated the ratio of average inter-cluster documents dissimilarities to average intra-cluster document dissimilarities. It is obvious that in search result diversification, a good document dissimilarity function would get large inter-cluster document dissimilarities and small intra-cluster document dissimilarities (large ratio value). This is because such a dissimilarity function could discriminate the subtopics well.

Table 7 shows the ratios calculated based on different dissimilarity definitions and different preliminary document representations. From the results, we can see that the ratio of "d_n with PLSA" (documents represented with PLSA topics and dissimilarities are calculated with neural tensor

Table 7: Ratio of average inter-cluster documents dissimilarities to average intra-cluster document dissimilarities. The documents are grouped according to their associated subtopics.

Method	average dissimilarity ratio
d_e with PLSA	1.65
d_n with PLSA	**2.73**
d_e with doc2vec	2.10
d_n with doc2vec	**4.32**

network) is larger than the ratio of "d_e with PLSA" (documents represented with PLSA topics and dissimilarities are calculated as Euclidean distance), and the ratio of "d_n with doc2vec" is larger than the ratio of "d_e with doc2vec". The results indicates that the dissimilarity functions learned by the tensor neural network are better than the Euclidean distances, in terms of discriminating the query subtopics.

The conclusion is quite intuitive and nature because the parameters of neural tensor network are determined based on the labeled data and thus can be adapted to the specific dataset and task, while the Euclidean distance is a predefined function for all datasets and tasks. Therefore, we can conclude that R-LTR-NTN (and also PAMM-NTN) can improve the performances through learning a better document dissimilarity function which distinguishes the documents with different subtopics effectively.

5.4.2 Ability to improve queries with high ambiguity

We also conducted experiments to show on which kinds of queries our approaches can perform well. Specifically, in each fold of the experiments on WT2009, we trained an R-LTR-NTN$_{\text{doc2vec}}$ model, an R-LTR, and a PAMM(α-NDCG) model on the training data and tested them on the test data. We then grouped the queries in the test datasets according to the number of subtopics they associated. We compared the performances of these three models in terms of α-NDCG@20 on each of the query groups and the results are shown in Figure 3. Boldface indicates the number of associated subtopics by the candidate documents, and the numbers in the parentheses indicate the proportion of queries in that group to the number of all queries. Please note that in Figure 3 some queries associated with only one or two subtopics while in Table 2 all queries have at least 3 subtopics associated. This is because we used the Indri[7] toolkit to retrieve the top 1000 documents as the candidates. Some labeled documents may not be ranked at top 1000 and thus be eliminated from the candidate set.

From the results reported in Figure 3, we can see that for those queries that associated with only one or two subtopics, R-LTR-NTN performed worse than the baselines of R-LTR and PAMM(α-NDCG). However, for those queries that associated with three or more subtopics (queries with high ambiguity), R-LTR-NTN outperformed the baselines. We also observed the trends that larger improvements R-LTR-NTN can achieve on the queries with more subtopics. The results is also intuitive because the document relations are more complex for ambiguous queries and neural tensor network can model the complex document relationship better. Thus, we can conclude that R-LTR-NTN can improve the baselines through improving the high ambiguity queries.

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

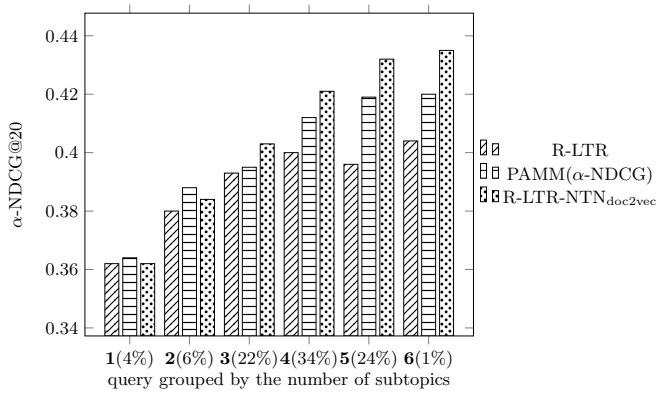

Figure 3: Performances with respect to query groups with different number of subtopics. The numbers in the parentheses indicate the proportion of queries in that group to the number of all queries.

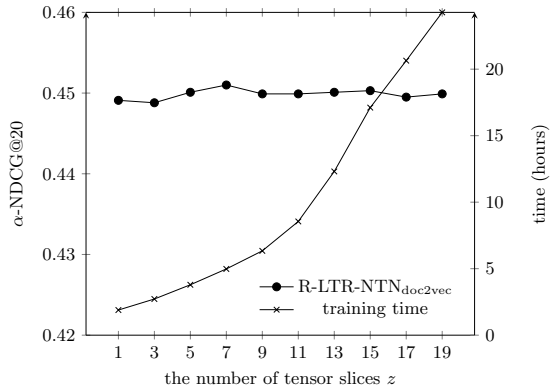

Figure 4: Ranking accuracies and training time with respect to the number of tensor slices z.

5.4.3 Effects of the number of tensor slices

Finally, we conducted experiments to test if the proposed algorithms are sensitive to the model parameters. One of the most important parameters in the proposed method is the number of tensor slices z. Thus, in the experiments we tested if R-LTR-NTN$_{\text{doc2vec}}$ is sensitive to different settings of z values. Specifically, we tuned z by varying the values of parameter z from 1 to 19, with step 2 and fixing other model parameters to the default or optimal values. Figure 4 shows the performances of R-LTR-NTN$_{\text{doc2vec}}$ with respect to number of slices z, in terms of α-NDCG@20. The training time (in hours) with respect to z are also shown in the figure.

From the results, we can see that the performances did not change much with different z values, which indicates R-LTR-NTN$_{\text{doc2vec}}$ (and other proposed algorithms) are robust and not sensitive to the parameter settings. In all of the experiments the number of tensor slices was set to the optimal value 7.

One of the negative effects of increasing z values is that the training time increased dramatically with the creased z values, as shown in Figure 4. This is because much more operations are needed for training the model if z is increased. Please refer to Section 4.2.5 for the time complexities of the proposed algorithms.

6. CONCLUSIONS

How to model the *novelty* of a candidate document with respect to other documents is one of the key problems in search result diversification. Existing approaches have been hurt from the necessaries of predefining a document similarity function or a set of novelty features, which are usually hard in real applications. In this paper we proposed to model the novelty of a document with a neural tensor network, which enables us to automatically learns a nonlinear novelty function based on the preliminary representations of the candidate document and other documents. Under the framework of relational learning to rank, new diverse learning to rank models have been derived, by replacing the novelty term in the original objective function with the neural tensor network. Experimental results based on three benchmark datasets showed that the proposed models significantly outperformed the baseline methods, including the state-of-the-art relational learning to rank models. Experimental results also showed that the proposed algorithms can improve the baselines via learning a document dissimilarity function that matches well with the query subtopics. The results also showed that more improvements can be achieved on the queries with high ambiguity.

As future work, we would like to verify the effectiveness of the proposed algorithms on applications other than search result diversification such as multi-document summarization etc. We also want to study the approaches to learning the relevance features and novelty features simultaneously.

7. ACKNOWLEDGMENTS

The work was funded by the 973 Program of China under Grants No. 2014CB340401 and 2012CB316303, the 863 Program of China under Grants No. 2014AA015204, the National Natural Science Foundation of China (NSFC) under Grants No. 61232010, 61472401, 61433014, 61425016, and 61203298, the Key Research Program of the Chinese Academy of Sciences under Grant No. KGZD-EW-T03-2, and the Youth Innovation Promotion Association CAS under Grants No. 20144310 and 2016102.

8. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of ACM WSDM '09*, pages 5–14, 2009.

[2] S. Bhatia. Multidimensional search result diversification: Diverse search results for diverse users. In *Proceedings of ACM SIGIR '11*, pages 1331–1332, 2011.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of ACM SIGIR '98*, pages 335–336, 1998.

[4] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of ACM CIKM '09* pages 1287–1296, 2009.

[5] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of ACM CIKM '09*, pages 621–630, 2009.

[6] 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 ACM SIGIR '08*, pages 659–666, 2008.

[7] C. L. Clarke, M. Kolla, and O. Vechtomova. An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of ICTIR '09*, pages 188–199, 2009.

[8] V. Dang and W. B. Croft. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of ACM SIGIR '12*, pages 65–74, 2012.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[10] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proceedings ofWWW '09*, pages 381–390, 2009.

[11] S. Guo and S. Sanner. Probabilistic latent maximal marginal relevance. In *Proceedings ofACM SIGIR '10*, pages 833–834, 2010.

[12] J. He, V. Hollink, and A. de Vries. Combining implicit and explicit topic representations for result diversification. In *Proceedings of ACM SIGIR '12*, pages 851–860, 2012.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of ACM SIGIR '99*, pages 50–57, 1999.

[14] S. Hu, Z. Dou, X. Wang, T. Sakai, and J.-R. Wen. Search result diversification based on hierarchical intents. In *Proceedings of ACM CIKM '15*, pages 63–72, 2015.

[15] Q. V. Le and T. Mikolov. Distributed Representations of Sentences and Documents. *ArXiv e-prints*, May 2014.

[16] H. Li. *Learning to rank for information retrieval and natural language processing; 2nd ed.* Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publ., San Rafael, CA, 2014.

[17] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of WWW '09*, pages 71–80, 2009.

[18] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.

[19] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of ACM SIGIR '05*, pages 472–479, 2005.

[20] L. Mihalkova and R. Mooney. Learning to disambiguate search queries from short sessions. In W. Buntine, M. Grobelnik, D. MladeniÄĞ, and J. Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5782 of *Lecture Notes in Computer Science*, pages 111–127. Springer Berlin Heidelberg, 2009.

[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.*, 13(4):346–374, Aug. 2010.

[22] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of ACM SIGIR '06*, pages 691–692, 2006.

[23] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of ACM ICML '08*, pages 784–791, 2008.

[24] D. Rafiei, K. Bharat, and A. Shukla. Diversifying web search results. In *Proceedings of WWW '10*, pages 781–790, 2010.

[25] K. Raman, P. Shivaswamy, and T. Joachims. Online learning to diversify from implicit feedback. In *Proceedings of ACM SIGKDD '12*, pages 705–713, 2012.

[26] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of WWW '10*, pages 881–890, 2010.

[27] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 926–934. Curran Associates, Inc., 2013.

[28] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of ACM SIGIR '15*, pages 113–122, 2015.

[29] Y. Yue and T. Joachims. Predicting diverse subsets using structural svms. In *Proceedings of ACM ICML '08*, pages 1224–1231, 2008.

[30] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of ACM ICML '09*, pages 1201–1208, 2009.

[31] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of ACM SIGIR '03*, pages 10–17, 2003.

[32] Y. Zhu, Y. Lan, J. Guo, X. Cheng, and S. Niu. Learning for search result diversification. In *Proceedings of ACM SIGIR '14*, pages 293–302, 2014.