Supervised Search Result Diversification via Subtopic Attention

Zhengbao Jiang, Zhicheng Dou, *Member, IEEE,* Wayne Xin Zhao, *Member, IEEE,* Jian-Yun Nie, *Member, IEEE,* Ming Yue, Ji-Rong Wen, *Senior Member, IEEE,*

Abstract—Search result diversification aims to retrieve diverse results to satisfy as many different information needs as possible. Supervised methods have been proposed recently to learn ranking functions and they have been shown to produce superior results to unsupervised methods. However, these methods use implicit approaches based on the principle of Maximal Marginal Relevance (MMR). In this paper, we propose a learning framework for explicit result diversification where subtopics are explicitly modeled. Based on the information contained in the sequence of selected documents, we use attention mechanism to capture the subtopics to be focused on while selecting the next document, which naturally fits our task of document selection for diversification. As a preliminary attempt, we employ recurrent neural networks and max pooling to instantiate the framework. We use both distributed representations and traditional relevance features to model documents in the implementation. The framework is flexible to model query intent in either a flat list or a hierarchy. Experimental results show that the proposed method significantly outperforms all the existing search result diversification approaches.

Index Terms—search result diversification, subtopics, attention.

1 Introduction

In real search scenario, queries issued by users are usually ambiguous or multi-faceted. In addition to being relevant to the query, the retrieved documents are expected to be as diverse as possible in order to cover different information needs. For example, when users issue "apple", the underlying intents could be the IT company or the fruit. The retrieved documents should cover both topics to increase the chance to satisfy users with different information needs.

Traditional approaches to search result diversification are usually unsupervised and adopt manually defined functions with empirically tuned parameters. Depending on whether the underlying intents (or subtopics) are explicitly modeled, they can be categorized into implicit and explicit approaches [1], [2]. Implicit approaches [3] do not model intents explicitly. They emphasize novelty, i.e. the following document should be "different" from the former ones based on some similarity measures. Instead, explicit approaches [4], [5], [6], [7], [8], [9] model intents (or subtopics) explicitly. They aim to improve intent coverage, i.e. the following document should cover the intents not satisfied by previous ones. Intents or subtopics can be determined by techniques such as query reformulation [10], [11], [12], [13] and query clustering based on query logs and other types of information. Existing studies showed that explicit approaches have better performance [5], [6], [7], [8], [9] than implicit approaches due to several reasons: on the one hand, they provide a more natural way to handle subtopics than implicit approaches; on the other hand, their ranking functions are closer to the diversity evaluation metrics which are mostly based on explicit subtopics. Furthermore, most similarity measures used in the implicit approaches, e.g., those based on language model or vector space model, are determined globally on the whole documents, regardless of possible search intents. This might be problematic for search result diversification: two documents could contain similar words and considered globally similar, but this similar part may be unrelated to underlying search intents.

To avoid heuristic and handcrafted functions and parameters, a new family of research work using supervised learning is proposed. They try to learn a ranking function automatically. Their major focus lies in the modeling of diversity, including structural prediction [14], rewarding functions for novel contents [15], measure-based direct optimization [16], and neural network based method [17]. Regardless of diversity modeling and optimization methods, all these solutions inherit the spirit of MMR which is an implicit approach and do not take intents into consideration. Although the learning methods may result in a better similarity measure, they are hindered by the gap between reducing document redundancy and improving intent coverage. They suffer from similar problems with implicit unsupervised approaches. Without modeling subtopics explicitly, they cannot directly improve intent coverage. Hence, there is a need to incorporate explicit subtopic modeling into supervised diversification methods.

To address the above issue, we propose to model subtopics in a general supervised learning framework. Our framework combines the strengths of both explicit unsupervised approaches and (implicit) supervised approaches. First, subtopics are explicitly modeled, allowing us to im-

- *Zhengbao Jiang, Zhicheng Dou, Wayne Xin Zhao, Ming Yue, and Ji-Rong Wen are with the School of Information, Beijing Key Laboratory of Big Data Management and Analysis Methods, and DEKE, Renmin University of China, Beijing 100872, P.R. China. E-mail: rucjzb@163.com, dou@ruc.edu.cn, batmanfly@gmail.com, yomin@ruc.edu.cn, jirong.wen@gmail.com*
- *Jian-Yun Nie is with DIRO, Université de Montréal, Québec, C.P. 6128, Succ Centre-Ville Montréal, Québec, Canada. E-mail: nie@iro.umontreal.ca*

Manuscript received July 12, 2017; revised July 12, 2017.

TABLE 1
Subtopic relevance example.

doc\subtopic	i_1	i_2	i_3
d_1	√	√	×
d_2	√	√	×
d_3	×	×	√
d_4	×	√	×
d_5	×	×	√

TABLE 2
Categorization of diversification approaches.

	unsupervised	supervised
implicit	MMR	SVM-DIV, R-LTR, PAMM, NTN
explicit	IA-Select, xQuAD, PM2, TxQuAD, TPM2, HxQuAD, HPM2, 0-1 MSKP	**DSSA** (our approach)

prove intent coverage in a proactive way. Second, it automatically learns the diversification ranking function, and is able to capture complex interaction among documents and subtopics. We call this framework Document Sequence with Subtopic Attention (DSSA). More specifically, to select the next document, we first model the sequence of selected documents in order to capture their contents as well as their relationship with the subtopics. Then based on the information contained by previous documents, attention mechanism is used to determine the under-covered subtopics to which we have to pay attention in selecting the next document. Attention mechanism has been successfully used to deal with various problems in image understanding [18] and NLP [19], [20]. This mechanism corresponds well to the document selection problem in search result diversification: attention on subtopics changes along with the addition of a document in the result list. For example. Assume that we have 3 subtopics and 4 documents whose relevance judgments are shown in Table 1. Given that we have selected d_1 and d_2, which cover subtopics i_1 and i_2, the attention for next choice should incline to i_3 which is not covered, thus d_3 is a better choice than d_4 at this position. We will show that the DSSA framework is general enough to cover the ideas of previous unsupervised explicit methods.

We propose a specific implementation of DSSA using recurrent neural networks (RNN) and max-pooling to leverage both distributed representations and relevance features, which we call DSSA-RNNMP. We further extend this model to introduce hierarchical subtopics. The basic idea is that subtopics inherently exist as a hierarchical structure, where subtopics on high levels represents general user intents while subtopics on low levels are more specific [8], [21]. Only considering coarse or fine-grained subtopics may result in suboptimal intent coverage. In particular, attention is calculated for subtopics on different levels. A document's matching scores to subtopics on the same level are combined by attention to obtain the score of this level. The final score is the weighted sum of the scores of different levels. We call this hierarchical model HDSSA-RNNMP. Experimental results on TREC Web Track data show that DSSA-RNNMP outperforms the existing methods significantly and HDSSA-RNNMP further improves the performance. To our knowledge, this is the first time that a supervised learning framework with attention mechanism is used to model subtopics explicitly for search result diversification.

2 Related Work

2.1 Implicit Diversification Approaches

The basic assumption of implicit diversification approaches is that dissimilar documents are more likely to satisfy different information needs. The most representative approach is MMR [3]:

$$S_{\text{MMR}}(q, d, \mathcal{C}) = (1-\lambda)S^{\text{rel}}(d, q) - \lambda \max_{d_j \in \mathcal{C}} S^{\text{div}}(d, d_j), \quad (1)$$

where S^{rel} and S^{div} model document d's relevance to the query q and its similarity to a selected documents d_j respectively. To gain high ranking score, a document should not only be relevant, but also be dissimilar from the selected documents. The definition of measures for relevance and document similarity is crucial, which is done manually in this approach. Based on desirable facility placement principle [22], [23] proposes first clustering the candidate documents then composing the diverse result set, which achieves a good balance between effectiveness and efficiency.

Recently, machine learning methods have been leveraged to learn score functions. Yue and Joachims [14] proposed SVM-DIV which uses structural SVM to learn to identify a document subset with maximum word coverage. However, word coverage may be different from intent coverage. Optimizing the former may not necessarily lead to optimizing the latter. Similar to MMR, Zhu et al. [15] proposed relational learning-to-rank model (R-LTR) which learns to score a document based on both relevance and novelty automatically, in order to maximize the probability of optimal rankings. Based on R-LTR score function, Xia et al. [16] proposed a perceptron algorithm using measures as margins (PAMM) to directly optimize evaluation metrics by enlarging the score margin of positive and negative rankings. They further proposed to use a neural tensor network (NTN) [17] to measure document similarity automatically from document representations, which avoids the burden to define handcrafted diversity features.

The above supervised approaches are shown to outperform the unsupervised counterparts. However, they are all implicit approaches without using subtopics. In this paper, we propose a learning-based explicit approach which models subtopics explicitly.

2.2 Explicit Diversification Approaches

Explicit approaches model subtopics underlying a query, aiming at returning documents covering as many subtopics as possible. These approaches leverage external resources to explicitly represent information needs in subtopics. IA-Select [4] uses classified topical categories based on ODP taxonomy. xQuAD [5] is a probabilistic framework that uses query reformulations as intent representations. PM2 [6] tackles search result diversification problem from the perspective of proportionality. TxQuAD and TPM2 [7] represent intents by terms and transform intent coverage to term coverage. Hu et al. [8] proposed to use a hierarchical

structure for subtopics instead of a flat list, which copes with the inherent interaction among subtopics. The benefit of hierarchical subtopics lies in that user intents of different granularities are modeled simultaneously. Two specific models, namely HxQuAD and HPM2, were proposed using hierarchical structure. Yu et al. [9] formulated diversification task as a 0-1 multiple subtopic knapsacks (0-1 MSKP) problem where documents are chosen like filling up multiple subtopic knapsacks. To tackle this NP-hard problem, max-sum belief propagation is used.

As summarized in Table 2, all existing explicit approaches are unsupervised and the functions and parameters are defined heuristically. In this paper, we use supervised learning to model the interaction among documents and subtopics simultaneously.

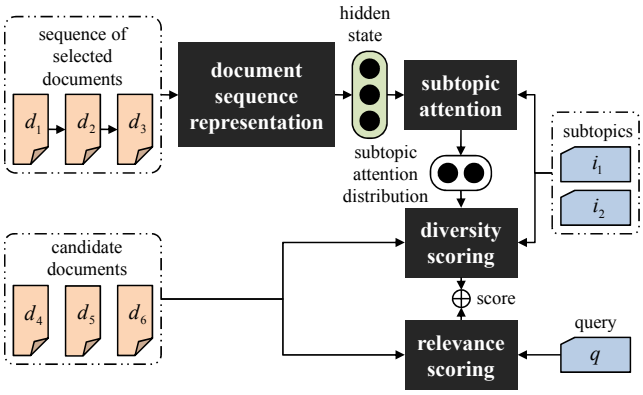


Fig. 1. Illustration of DSSA framework.

2.3 RNN with Attention Mechanism

RNN can capture the interdependency between elements in a sequence. Attention mechanism, which is usually built on RNN, mimics human attention behavior focusing on different local region of the object (an image, a sentence, etc.) at different times. In computer vision, [18] used RNN with attention to extract information from an image by adaptively selecting a sequence of the most informative regions instead of the whole image. In NLP, attention mechanism is typically used in neural machine translation (NMT). Traditional encoder-decoder models encode the source sentence into a fixed-length vector from which the target sentence is decoded. Such fixed-length vector may not be powerful enough to reflect all the information of the source sentence. An attention-based model [19] was proposed to automatically pay unequal and varied attention to source words during decoding process. In particular, to decide the next target word, not only the fixed-length vector, but also the hidden states corresponding to source words relevant to the target word are used. Luong et al. [20] generalized the idea and proposed two classes of attention mechanism, namely global and local approaches. In this paper, attention mechanism is used on subtopics, which guides the model to emphasize different intents at different positions.

In the following section, we will first propose a general framework, then instantiate it with a specific implementation.

3 Document Sequence with Subtopic Attention Framework

Given a query set $\mathcal{Q}$, a document set $\mathcal{D}_q$ and a subtopic set $\mathcal{I}_q$ for each query $q \in \mathcal{Q}$, the goal of explicit methods is to learn a ranking function $f(q, \mathcal{D}_q, \mathcal{I}_q)$ which is expected to output a ranking of documents in $\mathcal{D}_q$ that is both *relevant* and *diverse*. The loss function could be written in the following general form:

$$\sum_{q \in \mathcal{Q}} L(f(q, \mathcal{D}_q, \mathcal{I}_q), \mathcal{Y}_q), \tag{2}$$

where L measures the quality gap between the ranking outputted by f and the best ranking $\mathcal{Y}_q$. Different from traditional retrieval tasks, diversity has to be considered in the ranking and evaluation process. Theoretically, diversity ranking is NP-hard [4], [24]. Hence, a common strategy is to make greedy selections [3], [5]: at the t-th position, we assume that $t-1$ documents have been selected and formed a document sequence $\mathcal{C}_{t-1}$. The task is to select a locally optimal document d_t from the remaining candidate documents based on a score function $S(q, d_t, \mathcal{C}_{t-1}, \mathcal{I}_q)$. Note that implicit supervised methods correspond to the case where $\mathcal{I}_q$ is an empty set. $\mathcal{C}_{t-1}$ should be modeled as a sequence instead of a set, which means that the order of documents matters. The reason is that users scan documents sequentially and better utility could be achieved by making adjacent documents diverse. For example, given $\mathcal{C}_2 = [d_1, d_3]$ as showed in Table 1, it is better to rank d_2 at the third position than d_5.

To motivate our approach, we start with the ideas of the unsupervised explicit approaches, which can be formulated as the following general form:

$$\begin{aligned} &S_{\text{unsupervised}}(q, d_t, \mathcal{C}_{t-1}, \mathcal{I}_q) = \\ &(1-\lambda) S^{\text{rel}}(d_t, q) + && \Rightarrow \text{relevance} \\ &\lambda \sum_{i_k \in \mathcal{I}_q} S^{\text{div}}(d_t, i_k) \underbrace{A(\mathcal{C}_{t-1}, \mathcal{I}_q)_k}_{\text{subtopic weights}}, && \Rightarrow \text{diversity} \end{aligned} \tag{3}$$

where $i_k \in \mathcal{I}_q$ is the k-th subtopic of q and S^{rel} and S^{div} calculate document d_t's relevance to a query and to a subtopic respectively. The essence of diversity lies in the function A which calculates the weights for subtopics $\mathcal{I}_q$ based on previous document sequence $\mathcal{C}_{t-1}$. For xQuAD, $A(\mathcal{C}_{t-1}, \mathcal{I}_q)_k = P(i_k|q) \prod_{d_j \in \mathcal{C}_{t-1}} (1 - P(d_j|i_k))$ where $P(i_k|q)$ is the initial importance of subtopic i_k, $P(d_j|i_k)$ is the probability that d_j is relevant to i_k. The weight of a subtopic is determined by the likelihood that previous documents are not relevant to this subtopic. PM2 mimics seats allocation of competing political parties to adjust subtopic weights after each selection, i.e. $A(\mathcal{C}_{t-1}, \mathcal{I}_q)$ is estimated according to the difference between the subtopic's distributions in $\mathcal{C}_{t-1}$ and in $\mathcal{I}_q$. All these methods don't model the selected documents as a sequence. In addition, the functions and parameters are heuristically defined, which may not best fit the final goal.

To tackle the above problems, we extend Equation (3) to

TABLE 3
Notations in DSSA.

Notation	Definition
r, d_t	a ranking, the t-th document.
q, i_k	the query, the k-th subtopic.
$\boldsymbol{v}_{d_t}$	representation of the document at the t-th position.
$\boldsymbol{v}_q$	representation of the query.
$\boldsymbol{v}_{i_k}$	representation of the k-th subtopic.
$\boldsymbol{h}_t$	hidden state of previous t documents.
$a_{t,k}$	attention on the k-th subtopic at the t-th position. $\sum_{k=1}^{K} a_{t,k} = 1, a_{t,k} \in [0,1]$ where K is the number of subtopics. A large value means that this subtopic is less satisfied by previous $t-1$ documents and thus needs more attention at the t-th position.
s_{d_t}	the final score of the document at the t-th position.

TABLE 4
Parameters in DSSA-RNNMP.

Notation	Definition
$\boldsymbol{W}^n$, $\boldsymbol{b}^n$	parameters of RNN with vanilla cell.
$\boldsymbol{W}^a$, $\boldsymbol{w}^p$	parameters used in subtopic attention.
$\boldsymbol{W}^s$, $\boldsymbol{w}^r$	parameters used in scoring.

the following general learning framework:

$$
\begin{aligned}
&S_{\text{DSSA}}(q, d_t, \mathcal{C}_{t-1}, \mathcal{I}_q) = s_{d_t} = \\
&(1-\lambda)\mathcal{S}^{\text{rel}}(\boldsymbol{v}_{d_t}, \boldsymbol{v}_q) + && \Rightarrow \text{relevance} \\
&\lambda \mathcal{S}^{\text{div}}\Bigg(\boldsymbol{v}_{d_t}, \boldsymbol{v}_{i_{(\cdot)}}, \underbrace{\mathcal{A}\Big(\mathcal{H}([\boldsymbol{v}_{d_1}, ..., \boldsymbol{v}_{d_{t-1}}]), \boldsymbol{v}_{i_{(\cdot)}}\Big)}_{\text{subtopic attention}}\Bigg), && \Rightarrow \text{diversity}
\end{aligned}
\tag{4}
$$

where documents, queries, and subtopics are denoted by their representations, as explained in Table 3. In this paper, we focus on learning a ranking function only and assume that these representations are given and will not be modified. There are three main components, namely (1) **document sequence representation** component $\mathcal{H}$, (2) **subtopic attention** component $\mathcal{A}$, and (3) **scoring** component $\mathcal{S}^{\text{rel}}$ and $\mathcal{S}^{\text{div}}$, which are also illustrated in Figure 1. This framework is inspired from the attention models used in image understanding [18] and neural machine translation [19], [20], however adapted to our diversification task.

Next, we briefly describe the three components. The document sequence representation component $\mathcal{H}$ encodes the information contained in document sequence $\mathcal{C}_{t-1}$ into a fixed-length hidden state $\boldsymbol{h}_{t-1}$, which could consider the interaction and dependency among these documents. $\boldsymbol{h}_{t-1}$ could be viewed as a comprehensive and high-level representation of $\mathcal{C}_{t-1}$. The subtopic attention $\boldsymbol{a}_{t,(\cdot)}$ is calculated by the subtopic attention component $\mathcal{A}$ using $\boldsymbol{h}_{t-1}$ and subtopic representations $\boldsymbol{v}_{i_{(\cdot)}}$. The attention evolves from the first to the last ranking position, driving the model to emphasize different subtopics based on previous document sequence. Finally, the scoring components $\mathcal{S}^{\text{rel}}$ and $\mathcal{S}^{\text{div}}$ calculate relevance and diversity scores respectively. Notice that $\mathcal{S}^{\text{div}}$ is not limited to be a weighted sum over all subtopics as Equation (3). It can incorporate more complex interaction among subtopics.

The essence of this framework can be summarized as follows. Along with the selection of more documents, we encode the information of previous document sequence, and the attention mechanism will monitor the degree of satisfaction for each subtopic. High scores are assigned to the documents relevant to less covered subtopics. Finally, multiple subtopics would be well covered by adaptively learning the attention. In this way, our framework builds an intuitive approach to explicitly model subtopics. We name the framework **Document Sequence with Subtopic Attention (DSSA)**. DSSA is a unified architecture that takes both relevance and diversity into consideration, and diversity is achieved by modeling the interaction among documents and subtopics.

4 Result Diversification using DSSA

In this section, we instantiate DSSA to a concrete form and articulate the training and prediction algorithms. The main idea of DSSA is to dynamically capture accumulative relevance information of previous document sequence, so as to calculate subtopic attention. Inspired by the recent progress on sequence data modeling, we adapt RNN to capture the information of previous document sequence based on distributed representations of documents. However, the effectiveness of distributed representation heavily depends on a large amount of training data. Typically, the representation is built automatically using the data to optimize an objective function [25]. We do not have such large data and we can only use unsupervised methods (e.g. doc2vec) to create representation, of which the effectiveness could be suboptimal. Indeed, our preliminary experiments using only the distributed representation created by unsupervised methods yield low effectiveness. To compensate this weakness, we also use traditional relevance features such as BM25 score, which are proven useful, to calculate subtopic attention and final score. Such a combination of distributed representations and features has been used in several previous works [17], [26]. In addition to RNN, we also adopt the way using max-pooling [17], which has been shown effective, to implement subtopic attention mechanism. We call this model DSSA-RNNMP (DSSA model using RNN and Max-Pooling), as illustrated in Figure 2. Based on this implementation, we further incorporate hierarchical subtopics to consider user intents on different granularities. In addition, we also propose a list-pairwise approach for optimization, which is different from the existing studies.

4.1 A Neural Network Implementation

We first describe the constitution of representations, namely $\boldsymbol{v}_{d_t}$, $\boldsymbol{v}_q$, and $\boldsymbol{v}_{i_k}$, then elaborate how we implement **document sequence representation**, **subtopic attention**, and **scoring** components. The parameters are listed in Table 4.

$\boldsymbol{v}_{d_t}$: the representation of a document is composed of two parts: distributed representations and relevance features. Distributed representation can be constructed in different ways. In this paper, we consider three methods: SVD, LDA [27], and doc2vec [28]. Relevance features are those used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2810873, IEEE Transactions on Knowledge and Data Engineering

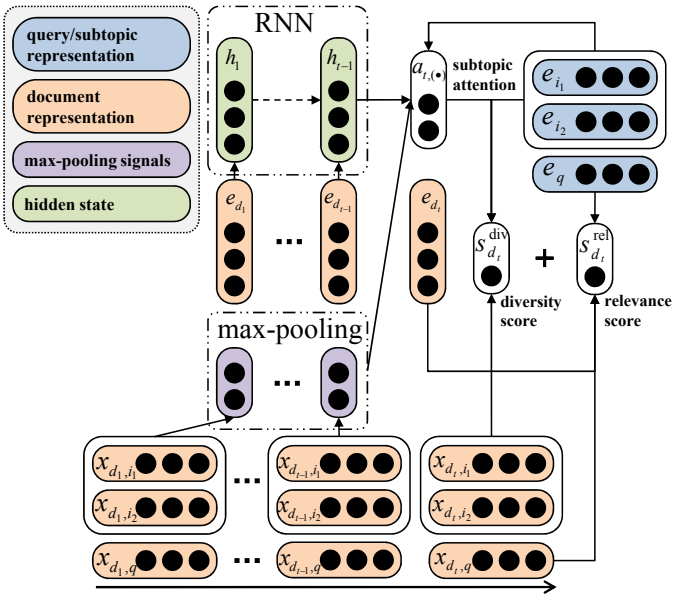


Fig. 2. Architecture of DSSA-RNNMP. Previous $t-1$ documents are encoded into $\boldsymbol{h}_{t-1}$ from distributed representations $\boldsymbol{e}_{d_1}, ..., \boldsymbol{e}_{d_{t-1}}$. Attention on the k-th subtopic $a_{t,k}$ is then calculated based on (1) hidden state $\boldsymbol{h}_{t-1}$ and subtopic representation $\boldsymbol{e}_{i_k}$ (2) max-pooling on relevance features $\boldsymbol{x}_{d_1,i_k}, ..., \boldsymbol{x}_{d_{t-1},i_k}$.

in traditional IR, such as BM25 score etc. Suppose that we have a distributed representation of size E_d, K subtopics, and R relevance features, the total size of $\boldsymbol{v}_{d_t}$ would be $E_d + R + KR$. We use $\boldsymbol{e}_{d_t} \in \mathbb{R}^{E_d}$, $\boldsymbol{x}_{d_t,q}$ and $\boldsymbol{x}_{d_t,i_k} \in \mathbb{R}^R$ to denote distributed representation, relevance features for a query and a subtopic respectively.

$\boldsymbol{v}_q$, $\boldsymbol{v}_{i_k}$: we first retrieve top Z documents using some basic retrieval model (such as BM25). These documents are concatenated as a pseudo document, then similar to $\boldsymbol{e}_{d_t}$, a distributed representation of size E_q is generated. For consistency, we also use $\boldsymbol{e}_q$ and $\boldsymbol{e}_{i_k} \in \mathbb{R}^{E_q}$ to represent these representations.

4.1.1 Document Sequence Representation

$\mathcal{H}$ is instantiated using RNN to encode the information of previous document sequence. Several types of RNN cell can be used, ranging from the simple vanilla cell, GRU cell [29], to LSTM cell [30]. For simplicity, we only show the vanilla cell here. At the t-th position, we derive the (accumulative) document sequence representation as follows:

$$\boldsymbol{h}_t = \tanh(\boldsymbol{W}^n[\boldsymbol{h}_{t-1}; \boldsymbol{e}_{d_t}] + \boldsymbol{b}^n), \tag{5}$$

where $\boldsymbol{W}^n \in \mathbb{R}^{U \times (U+E_d)}$ (U is the size of the hidden state), $\boldsymbol{b}^n \in \mathbb{R}^U$ and $[;]$ is a concatenation. The cell transforms previous hidden layer $\boldsymbol{h}_{t-1}$ and current document distributed representation $\boldsymbol{e}_{d_t}$ to another space, where a bias $\boldsymbol{b}^n$ is added and a non-linear activation (i.e. $\tanh$) then happens, producing the next hidden layer $\boldsymbol{h}_t$. $\boldsymbol{h}_0$ is initialized as a vector of zeros. The vanilla cell can be easily replaced by GRU and LSTM cells, whose results will be report in Section 6.2.

4.1.2 Subtopic Attention

By looking at $\boldsymbol{h}_{t-1}$ which stores the information of previous $t-1$ documents and $\boldsymbol{e}_{i_{(\cdot)}}$ which represents the meaning of each subtopic, we are capable of discovering which intents are not satisfied and thus need to be emphasized at the t-th position. To capture this idea, we use $\mathcal{A}'(\boldsymbol{h}_{t-1}, \boldsymbol{e}_{i_k})$ to measure the (unnormalized) importance of the k-th subtopic at the t-th position, which could be implemented in many ways. We consider the following two ways similar to [20]:

$$\mathcal{A}'(\boldsymbol{h}_{t-1}, \boldsymbol{e}_{i_k}) = \begin{cases} \boldsymbol{h}_{t-1}^\intercal \boldsymbol{W}^a \boldsymbol{e}_{i_k}, & \text{(general)} \\ -\boldsymbol{h}_{t-1}^\intercal \cdot \boldsymbol{e}_{i_k}, & \text{(dot)} \end{cases} \tag{6}$$

where $\boldsymbol{W}^a \in \mathbb{R}^{U \times E_q}$. The "general" operation uses bilinear tensor product to relate two vectors multiplicatively through its nonlinearity [31]. The "dot" product requires both vectors to be in the same space. Similar $\boldsymbol{h}_{t-1}$ and $\boldsymbol{e}_{i_k}$ mean that previous documents are likely to satisfy this subtopic, and thus a lower attention score will be attributed to it. The above way mainly relies on distributed representations, which may not always be effective, especially under limited data.

Hence, we further leverage relevance features to enhance the subtopic attention. $\boldsymbol{x}_{d_t,i_k}$ directly reflects the degree of satisfaction for a subtopic-document pair and is combined linearly using $\boldsymbol{w}^p$ to form an explicit signal. To derive the accumulative information of the document sequence, we adopt commonly used max-pooling to select the most significant signal from previous documents, which is similar to the max operation used in MMR. Max-pooling could also be interpreted as a regularizer, which reduces the number of parameters and thus avoids overfitting:

$$\mathcal{A}''(\boldsymbol{x}_{d_1,i_k}, ..., \boldsymbol{x}_{d_{t-1},i_k}) = \max([\boldsymbol{x}_{d_1,i_k}^\intercal \cdot \boldsymbol{w}^p, ..., \boldsymbol{x}_{d_{t-1},i_k}^\intercal \cdot \boldsymbol{w}^p]), \tag{7}$$

where $\mathcal{A}''(\boldsymbol{x}_{d_1,i_k}, ..., \boldsymbol{x}_{d_{t-1},i_k})$ measures the degree of satisfaction of the k-th subtopic based on relevance features through max-pooling. Lower value indicates that the previous documents are more likely to be relevant to this subtopic. Note that if we view the signals produced by max-pooling (i.e. the vectors in "max-pooling" section of Figure 2) as a part of the general hidden states, our concrete implementation fit in DSSA framework.

We adopt an addictive way to integrate both parts and then use softmax to produce (normalized) attention distribution:

$$\begin{aligned} a'_{t,k} &= \mathcal{A}'(\boldsymbol{h}_{t-1}, \boldsymbol{e}_{i_k}) + \mathcal{A}''(\boldsymbol{x}_{d_1,i_k}, ..., \boldsymbol{x}_{d_{t-1},i_k}), \\ a_{t,k} &= \frac{w_{i_k} \exp(a'_{t,k})}{\sum_{j=1}^K w_{i_j} \exp(a'_{t,j})} \quad (w_{i_j} \geq 0, \forall j). \end{aligned} \tag{8}$$

softmax is modified to include the initial subtopic importance w_{i_k}, which encodes our intuition that important subtopics are more likely to gain attention.

4.1.3 Scoring

The final score consists of relevance score $s_{d_t}^{\text{rel}}$ and diversity score $s_{d_t}^{\text{div}}$, which are combined by a coefficient λ:

$$s_{d_t} = (1-\lambda) s_{d_t}^{\text{rel}} + \lambda s_{d_t}^{\text{div}} \quad (0 \leq \lambda \leq 1). \tag{9}$$

The relevance and diversity score are calculated as follows:

$$s_{d_t}^{\text{rel}} = \mathcal{S}^{'}(\boldsymbol{e}_{d_t}, \boldsymbol{e}_q) + \boldsymbol{x}_{d_t,q}^{\intercal} \cdot \boldsymbol{w}^r,$$
$$s_{d_t}^{\text{div}} = \boldsymbol{a}_{t,(\cdot)}^{\intercal} \cdot \begin{bmatrix} \mathcal{S}^{'}(\boldsymbol{e}_{d_t}, \boldsymbol{e}_{i_1}) + \boldsymbol{x}_{d_t,i_1}^{\intercal} \cdot \boldsymbol{w}^r \\ \vdots \\ \mathcal{S}^{'}(\boldsymbol{e}_{d_t}, \boldsymbol{e}_{i_K}) + \boldsymbol{x}_{d_t,i_K}^{\intercal} \cdot \boldsymbol{w}^r \end{bmatrix}, \tag{10}$$

where $\boldsymbol{w}^r \in \mathbb{R}^R$ and $\boldsymbol{a}_{t,(\cdot)}$ is the attention derived from subtopic attention component. The diversity score is calculated as a weighted combination of the document's relevance to each subtopic by attention distribution. We use the same way to calculate document's relevance to a query and to its subtopics using both distributional representations and relevance features, although different ways can be used. Specifically, d_t's relevance to a query q (or a subtopic i_k) is calculated based on both the similarity between two distributed representations $\mathcal{S}^{'}(\boldsymbol{e}_{d_t}, \boldsymbol{e}_q)$ (or $\mathcal{S}^{'}(\boldsymbol{e}_{d_t}, \boldsymbol{e}_{i_k})$) and relevance features $\boldsymbol{x}_{d_t,q}$ (or $\boldsymbol{x}_{d_t,i_k}$). $\mathcal{S}^{'}$ intends to produce a matching score between two representations and $\boldsymbol{w}^r$ linearly combines features. Similar to $\mathcal{A}^{'}$, $\mathcal{S}^{'}$ could also be implemented as:

$$\mathcal{S}^{'}(\boldsymbol{e}_{d_t}, \boldsymbol{e}_{i_k}) = \begin{cases} \boldsymbol{e}_{d_t}^{\intercal} \boldsymbol{W}^s \boldsymbol{e}_{i_k}, & \text{(general)} \\ \boldsymbol{e}_{d_t}^{\intercal} \cdot \boldsymbol{e}_{i_k}, & \text{(dot)} \end{cases} \tag{11}$$

where $\boldsymbol{W}^s \in \mathbb{R}^{E_d \times E_q}$. Then the score of a ranking r is calculated by summing up all the $|r|$ documents' scores:

$$s_r = \sum_{t=1}^{|r|} s_{d_t}. \tag{12}$$

Vector interaction operations $\mathcal{A}^{'}$ and $\mathcal{S}^{'}$ could be implemented using more complex models, such as multilayer perceptron (MLP), to model the interaction between two vectors more accurately. We could also use convolutional neural network (CNN) instead of RNN to model the interaction among a sequence of documents and encode their information. We deliberately choose to use simple mechanisms in this implementation in order to show that the general framework is capable of capturing the essence of diversification even without complex operations. More complex implementations will be examined in future work.

4.2 Hierarchical Diversification

Inspired by the idea of organizing subtopics in a hierarchical structure proposed by Hu et al. [8], we introduce hierarchical subtopics into DSSA. The intuition behind hierarchical subtopics is that user intents have different granularities. It would be biased if we consider only coarse or fine-grained subtopics. To better understand the deficiency of using subtopics without considering different granularities, we use the query "defender" (query #20) as example, of which the hierarchical subtopics mined from search engine's query suggestions are shown in Figure 3. If we only use first-level subtopics which are crude, we cannot identify the difference between the document relevant to $i_{1,1,1}$ and the document relevant to $i_{1,1,3}$, as they are relevant to the same first-level subtopic $i_{1,1}$. In other words, diversification algorithms have risks to select multiple documents covering $i_{1,1,1}$ while ignoring documents corresponding to $i_{1,1,3}$ because they

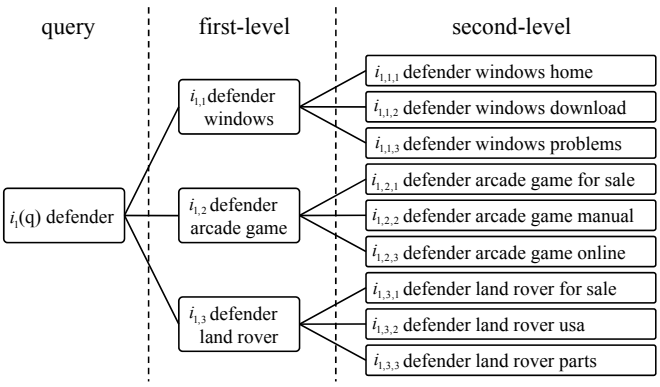


Fig. 3. Two-level hierarchical subtopics of query "defender".

cannot perceive the subtle difference between these two subtopics. In contrast, if we only consider second-level subtopics which are fine-grained, the algorithms may select three documents covering $i_{1,1,1}$, $i_{1,1,2}$, and $i_{1,1,3}$ respectively without realizing that these subtopics are generally similar. Comparing to choosing documents with subtle difference, it is more reasonable to select documents satisfying $i_{1,1}$, $i_{1,2}$, and $i_{1,3}$ respectively to cover a wider range of intents.

The model incorporating hierarchical subtopics is denoted as HDSSA-RNNMP. Assume that we have already obtained hierarchical subtopics using query suggestions provided by search engines (like Figure 3), we need to decide how to derive the attention paid on all subtopics of different levels and how to calculate the final score. Inspired by [8], we also calculate the attention and the score in a layer-wise approach. The hierarchical subtopic tree, with the query as the root, is split into several layers according to the depth of the nodes. All the subtopics of different layers are organized as a flat list, then attention of each subtopic is calculated just like Section 4.1.2. A document's matching scores to the subtopics on the same layer are combined by attention to get the score of this layer. Final score is the weighted sum of the score of each layer. In particular, the first layer only consists of the query; all the suggestions of the query constitute the second layer; the third layer is the suggestions of the subtopics of the second layer. Because the query acts as the "root subtopic" (i_1 in Figure 3), we can treat the query in the same way as other subtopics, which means that the attention is also calculated for the query. In consequence, parameter λ in Equation (9) is no longer necessary. In other words, the attention on the query serves as "query-wise λ", which controls the trade-off between relevance and diversity in a query-aware approach.

Formally, we use $i_{k_1,.,k_l}$ to represent the k_l-th child subtopic of parent subtopic $i_{k_1,.,k_{l-1}}$. Query is denoted as i_1. Its child subtopics are denoted as $i_{1,1}$, $i_{1,2}$, and etc. The subscript of a subtopic completely conveys its path to the root. The attention for each subtopic is calculated as follow:

$$a_{t,(k_1,.,k_l)}^{'} = \mathcal{A}^{'}(\boldsymbol{h}_{t-1}, \boldsymbol{e}_{i_{k_1,.,k_l}}) + \mathcal{A}^{''}(\boldsymbol{x}_{d_1,i_{k_1,.,k_l}}, ..., \boldsymbol{x}_{d_{t-1},i_{k_1,.,k_l}}),$$
$$a_{t,(k_1,.,k_l)} = \frac{w_{i_{k_1,.,k_l}} \exp(a_{t,(k_1,.,k_l)}^{'})}{\sum_{m=1}^{L} \sum_{k_1,.,k_m} w_{i_{k_1,.,k_m}} \exp(a_{t,(k_1,.,k_m)}^{'})}, \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2810873, IEEE Transactions on Knowledge and Data Engineering

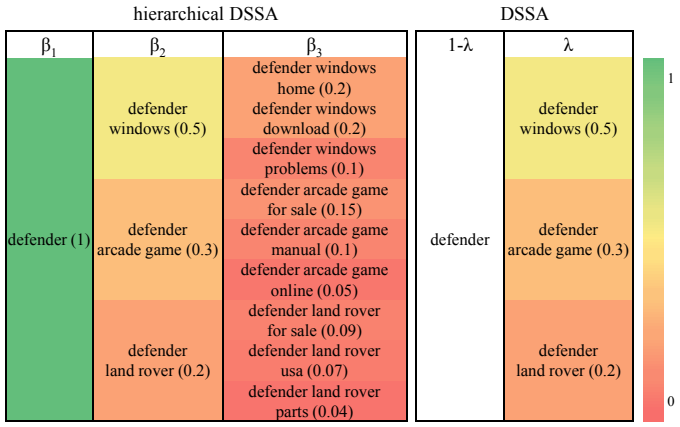


Fig. 4. Difference between HDSSA and DSSA. The initial weight of each subtopic is shown in parentheses and visualized via color scale. The colored subtopics are the ones to pay attention to.

where the attention score for subtopic $i_{k_1,.,k_l}$ is also calculate using both distributed representation and relevance features. The denominator of the softmax sums over all the subtopics of different layers. Consistent with [8], the initial weight of $i_{k_1,.,k_{l-1}}$ is the sum of the weights of its child subtopics:

$$w_{i_{k_1,.,k_{l-1}}} = \sum_{k_l} w_{i_{k_1,.,k_l}}. \tag{14}$$

To guarantee that each layer has the same initial weight, all the leaf nodes must appear on the deepest layer. If a subtopic without a child is not on the deepest layer, we treat itself as the only descendant. The final score of a document is calculated as follow:

$$s_{d_t} = \sum_{l=1}^{L} \beta_l \sum_{k_1,.,k_l} a_{t,(k_1,.,k_l)} (\mathcal{S}'(\boldsymbol{e}_{d_t}, \boldsymbol{e}_{i_{k_1,.,k_l}}) + \boldsymbol{x}^{\mathsf{T}}_{d_t,i_{k_1,.,k_l}} \cdot \boldsymbol{w}^r), \tag{15}$$

where the outer sum is over all L layers (in this paper $L = 3$) and the inner sum is over all subtopics of the l-th layer. In order to control the importance of the subtopics of different granularities, we use another layer-wise parameter β. We investigate two ways of defining β: balanced and unbalanced. In balanced setting, all the layers have the same weights, while in the unbalanced setting, we learn the weights of each layer simultaneously with other parameters. Both results are reported in Section 6. Figure 4 depicts the difference between HDSSA and DSSA. In DSSA, the importance of each subtopic is affected by its attention and λ, while the importance of the query is only affected by λ. However, in HDSSA, the importance of both query and subtopics is controlled by attention and layer-wise weight β, which is more flexible. DSSA based on a flat list of subtopics is a special case of HDSSA.

4.3 A List-pairwise Approach for Optimization

Liu [32] classifies LTR approaches into three categories: pointwise, pairwise, and listwise. Search result diversification is naturally a listwise problem because the score of a document depends on the previous documents. Take Table 1 as an example, under no previous documents, d_2 is better than d_3 because d_2 covers one more subtopic (subtopics are of equal weight). However, given that we have selected d_1, which is similar to d_2 while dissimilar to d_3, d_3 becomes superior because it provides additional information.

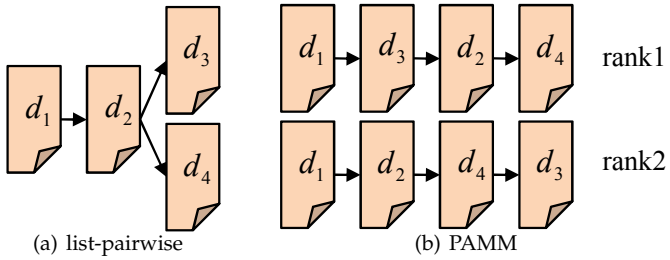


Fig. 5. Pair sample examples of (a) list-pairwise and (b) PAMM. Both samples are positive.

4.3.1 List-pairwise Training

We propose a list-pairwise training approach. We call it list-pairwise because a sample in our algorithm consists of a pair of rankings (r_1, r_2): r_1 and r_2 are totally identical except the last document. The sample can be written as $(\mathcal{C}, d^1, d^2)$, where $\mathcal{C}$ is the shared previous document sequence. The pairwise preference ground-truth is generated based on an evaluation metric M, such as α-nDCG. If $M(r_1) > M(r_2)$, it is positive, otherwise it is negative. Our approach is similar to pairwise approaches because it aims to compare a pair of documents, but this is done within some context. Similarly to pairwise, the loss function can be defined as binary classification logarithmic loss:

$$L_{\text{list-pairwise}} = \sum_{q \in \mathcal{Q}} \sum_{o \in \mathcal{O}_q} w^{(o)} \Big(y^{(o)} \log \Big(P(r_1^{(o)}, r_2^{(o)}) \Big) + (1 - y^{(o)}) \log \Big(1 - P(r_1^{(o)}, r_2^{(o)}) \Big) \Big), \tag{16}$$

where $\mathcal{O}_q$ is all the pair samples of query q, $y^{(o)} = 1$ indicates positive and 0 for negative, and $P(r_1^{(o)}, r_2^{(o)})$ is the probability of being positive calculated by $\frac{1}{1+\exp(s_{r_2^{(o)}} - s_{r_1^{(o)}})}$. To enhance effectiveness, we weight pairs with $w^{(o)} = |M(r_1^{(o)}) - M(r_2^{(o)})|$, which means that the bigger the metric score gap, the more important the pair.

Because DSSA calculates document d's score $s_d^{\mathcal{C}}$ based on previous document $\mathcal{C}$, we could also use Maximum Likelihood Estimation (MLE) or PAMM to optimize our model. We use Plackett-Luce model [33] to estimate the probability of a ranking r:

$$P(r) = \prod_{i=1}^{|r|} \frac{\exp(s_{d_i}^{r[:i-1]})}{\sum_{j=i}^{|r|} \exp(s_{d_j}^{r[:i-1]})}, \tag{17}$$

where $r[: i-1]$ means the top $i-1$ documents of ranking r. Then the loss functions could be written as:

$$L_{\text{MLE}} = \sum_{q \in \mathcal{Q}} -\log(P(r_q^+)), \tag{18}$$

$$L_{\text{PAMM}} = \sum_{q \in \mathcal{Q}} \sum_{r_q^+, r_q^-} [\![P(r_q^+) - P(r_q^-) \leq M(r_q^+) - M(r_q^-)]\!], \tag{19}$$

where $[\![\text{condition}]\!]$ is 1 if the condition is satisfied, 0 otherwise, MLE maximizes the probability of positive rankings, and PAMM enlarges the probability margin between positive and negative rankings according to an evaluation metric. For MLE, the number of best rankings is usually small if we only have hundreds of queries, which may not be enough to train adequately the parameters. PAMM uses preferences between very different rankings that are not comparable (see Figure 5(b)). In contrast, list-pairwise method only allows the last document to be different (Figure 5(a)). This corresponds better to the decision-making situation in which we have to choose a document under a given context. It is expected that such a pair sample allows us to better train the ranking function. Experiments will show that our approach works better.

As shown in Figure 2, our architecture is a unified neural network and the attention function is continuous, so the gradient of the loss function can be backpropagated directly to train the model. We use mini-batch gradient descent to facilitate training process.

Unfortunately, it is impossible to acquire all the list-pairwise samples, which has in total $|\mathcal{D}_q|!$ ($|\mathcal{D}_q|$ is the number of candidate documents) different permutations. So we develop a sampling strategy similar to negative sampling [34] as described in Algorithm 1: for each query q, we sample a large number of pairs of rankings, whose length ranges from 1 to $|\mathcal{D}_q|$. We first obtain some contexts $\mathcal{C}$ from both best rankings and randomly sampled negative rankings (rankings that are not optimal). Then under each $\mathcal{C}$, a pair of documents (d^1, d^2) are sampled from the remaining documents $\mathcal{D}_q \setminus \mathcal{C}$ if and only if they lead to different metric scores.

4.3.2 Prediction

In prediction stage, for each query, we sequentially and greedily choose the document with the highest score and append it to the ranking list. Specifically, the first document is selected under initial subtopic importance from the whole candidate set $\mathcal{D}_q$. Once the top $t-1$ documents have been selected (i.e. $|\mathcal{C}| = t-1$), we feed each document in $\mathcal{D}_q \setminus \mathcal{C}$ into DSSA at the t-th position one by one and choose the one with the highest s_{d_t}. This process continues until all the documents in $\mathcal{D}_q$ are ranked.

4.3.3 Time Complexities

The training time complexity with vanilla cell and "general" operation is $\mathcal{O}(V \cdot |\mathcal{Q}| \cdot \Gamma \cdot |\mathcal{D}_q| \cdot \Theta)$ where V is the number of iterations, $|\mathcal{Q}|$ is the number of training queries, $\Gamma = N \cdot |\mathcal{D}_q|^2$ is the number of sampled pairs where N is the number of random permutations, $|\mathcal{D}_q|$ is the number of candidate documents, and Θ is the complexity for each position:

$$\Theta = \underbrace{U(U+E_d)}_{\substack{\text{document sequence}\\ \text{representation}}} + \underbrace{KUE_q + KR}_{\substack{\text{subtopic}\\ \text{attention}}} + \underbrace{KE_dE_q + KR}_{\text{scoring}}, \quad (20)$$

where the dominating terms are KUE_q and KE_dE_q which are proportional to the number of subtopics K. How to efficiently handle a large number of subtopics is our future work. The prediction complexity is $\mathcal{O}(|\mathcal{D}_q|^2\Theta)$ for each query. We can limit $|\mathcal{D}_q|$ to a small number (say 50), so the prediction time can be reasonable. On a 24 core 2.1 GHz CPU server, the ranking takes about 150ms pre query. The ranking time of xQuAD and C-GLS are:

$$\begin{aligned} \text{xQuAD:} \quad & \mathcal{O}(|\mathcal{D}_q|^2 \cdot KR), \\ \text{C-GLS:} \quad & \mathcal{O}(V \cdot |\mathcal{D}_q| \cdot M^3), \end{aligned} \quad (21)$$

where we use R features and ListMLE to calculate the relevance between documents and queries (subtopics) for xQuAD; M is the size of the diverse subset (say 20) selected from the candidate set $\mathcal{D}_q$. Comparing to xQuAD, the matrix multiplication of distributed representation is the bottleneck, which could be accelerated by GPU. C-GLS uses pre-calculated distance to promote efficiency.

Algorithm 1 A List-pairwise Approach For Optimization

procedure LIST-PAIRWISE TRAINING
input: loss function L, learning rate r, epochs V, query set $\mathcal{Q}$, document set $\mathcal{D}$, evaluation metric M, random permutation count N
output: DSSA with trained parameters θ
  initialize θ
  for i from 1 to V **do**
    for batch $b \in$ GetSamples$(\mathcal{Q}, \mathcal{D}, M, N)$ **do**
      $g \leftarrow$ GetGradient$(L(b, \theta))$
      $\theta \leftarrow \theta - rg$
  return DSSA$_\theta$

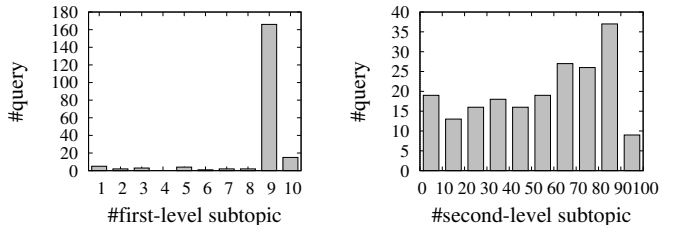
procedure GETSAMPLES
input: query set $\mathcal{Q}$, document set $\mathcal{D}_q$ for each query q, evaluation metric M, random permutation count N
output: a set of ranking pairs with weight and preference $\{(q^1, \mathcal{C}^1, d_1^1, d_2^1, w^1, y^1), (q^2, \mathcal{C}^2, d_1^2, d_2^2, w^2, y^2), ...\}$
include: GetPerms$(\mathcal{D}_q, l, N, M)$ return a best ranking (under metric M) and N random permutations of length l.
  $\mathcal{R} \leftarrow \emptyset$
  for query q in $\mathcal{Q}$ **do**
    for l from 0 to $|\mathcal{D}_q| - 1$ **do**
      for perm $\mathcal{C}$ in GetPerms$(\mathcal{D}_q, l, N, M)$ **do**
        $\mathcal{R} \leftarrow \mathcal{R} \cup$ GetPairs$(q, \mathcal{D}_q, \mathcal{C}, M)$
  return $\mathcal{R}$

procedure GETPAIRS
input: query q, document set $\mathcal{D}_q$, selected documents list $\mathcal{C}$, evaluation metric M
output: a set of ranking pairs with weight and preference $\{(q, \mathcal{C}^1, d_1^1, d_2^1, w^1, y^1), (q, \mathcal{C}^2, d_1^2, d_2^2, w^2, y^2), ...\}$
  $\mathcal{R} \leftarrow \emptyset$
  for all doc pair (d_1, d_2) in $\mathcal{D}_q \setminus \mathcal{C}$ **do**
    $r_1 \leftarrow [\mathcal{C}, d_1]$, $r_2 \leftarrow [\mathcal{C}, d_2]$
    if $M(r_1) \neq M(r_2)$ **then**
      $w \leftarrow |M(r_1) - M(r_2)|$
      $y \leftarrow [\![M(r_1) > M(r_2)]\!]$
      $\mathcal{R} \leftarrow \mathcal{R} \cup (q, \mathcal{C}, d_1, d_2, w, y)$
  return $\mathcal{R}$

5 EXPERIMENTAL SETTINGS

5.1 Data Collections

We use the same dataset as [8] which consists of Web Track dataset from TREC 2009 to 2012. There are 198 queries (query #95 and #100 are dropped because no diversity judgments are made for them), each of which includes 3 to 8



(a) hist of #first-level subtopics (b) hist of #second-level subtopics

Fig. 6. Histogram of the number of subtopics.

subtopics identified by TREC assessors. The relevance rating is given in a binary form at subtopic level. All experiments are conducted on ClueWeb09 [35] collection.

We use query suggestions of Google search engine as the first-level subtopics. Then the first-level subtopics are issued as queries to Google to retrieve their suggestions as the second-level subtopics. Finally, 1,696 first-level subtopics and 10,527 second-level subtopics are collected, which are released by Hu et al. [8] on their website[1]. Almost all queries have 9 or 10 first-level subtopics. But the first-level subtopics have variant number of second-level subtopics. The histogram of the number of subtopics of a query is shown in Figure 6. For DSSA, we only use the first-level subtopics. For HDSSA, we use both first and second level subtopics. Following the existing work [8], we simply use uniform weights in DSSA. For HDSSA, the weight of the root subtopic (the query) is 1. Then the weight is evenly distributed to child subtopics in a top-down manner. Note that the absolute value of the initial weights will not affect the final score because of the normalization of softmax.

5.2 Evaluation Metrics

We use ERR-IA [36], α-nDCG [37], and NRBP [38], which are official diversity evaluation metrics used in Web Track. They measure the diversity by explicitly rewarding novelty and penalizing redundancy. D♯-measures [39], the primary metric used in NTCIR Intent [40] and IMine task [41], is also included. We also use traditional diversity measures Precision-IA (denoted as Pre-IA) [4] and Subtopic Recall (denoted as S-rec) [42]. Consistent with existing works [15], [16], [17] and TREC Web Track, all these metrics are computed on top 20 results of a ranking. We use two-tailed paired t-test to conduct significance testing with p-value < 0.05.

5.3 Baseline Models

We compare DSSA and HDSSA[2] to various unsupervised and supervised diversification methods. The non-diversified baseline is denoted as **Lemur**. We use **C-GLS** [23], **xQuAD** [5], **PM2** [6], **TxQuAD**, **TPM2** [7], **HxQuAD**, and **HPM2** [8] as our unsupervised baselines. We use **ListMLE** [43], **R-LTR** [15], **PAMM** [16], and **NTN** [17] as our supervised baselines. Top 20 results of Lemur are used to train supervised methods. Top 50 (i.e. $|\mathcal{D}_q|$) results of Lemur are used for diversity re-ranking. To construct the representation of a query or a subtopic, we use the top 20 (Z) documents. We use 5-fold cross validation to tune the parameters in all experiments based on α-nDCG@20, which is one of the most widely used metrics. A brief introduction to these baselines is as follows:

TABLE 5
Relevance features. Each of the first 3 features is applied to body, anchor, title, URL, and the whole documents.

Name	Description	#Features
TF-IDF	the TF-IDF model	5
BM25	BM25 with default parameters	5
LMIR	LMIR with Dirichlet smoothing	5
PageRank	PageRank score	1
#inlinks	number of inlinks	1
#outlinks	number of outlinks	1

TABLE 6
Diversity features. Each feature is extracted over a pair of documents.

Name	Description
subtopic diversity	euclidean distance based on SVD
text diversity	cosine-based distance on term vector
title diversity	text diversity on title
anchor text diversity	text diversity on anchor
link-based diversity	link similarity of document pair
URL-based diversity	URL similarity of document pair

Lemur. We use the same non-diversified results as [8]. They are produced by language model and retrieved using the Lemur service[3] of which the spams are filtered. These results are released by Hu et al. [8] on the website[1].

ListMLE. ListMLE is a representative listwise LTR method without considering diversity.

C-GLS. We use k-means for clustering and tune the parameters λ, b, and α in the same way as [23].

xQuAD, PM2, TxQuAD, TPM2, HxQuAD, and HPM2. These are competitive unsupervised explicit diversification methods, as introduced in Section 2.2. All these methods use λ to control the importance of relevance and diversity. HxQuAD and HPM2 use an additional parameter α to control the weight of each layer of the hierarchical structure. Both λ and α are tuned using cross validation. They all require a prior relevance function to fulfill diversification re-ranking. Following [15], we use ListMLE.

R-LTR, PAMM, and NTN. For PAMM, we use α-nDCG@20 as the optimization metric. We optimize NTN based on both R-LTR and PAMM, denoted as R-LTR-NTN and PAMM-NTN respectively.

To achieve optimal results, for R-LTR and PAMM, we tune the relational function $h_S(R)$ from minimal, maximal, and average. For PAMM, we tune the number of positive rankings τ^+ and negative rankings τ^- per query. For NTN, the number of tensor slices is tuned from 1 to 10. LDA is used to generate distributed representations of size 100 for NTN and DSSA. For all these supervised methods, the learning rate r is tuned from 10^{-7} to 10^{-1}. For DSSA, we have different settings possible. In our first set of results, we will use "general" as the implementation of vector interaction operations $\mathcal{A}^{'}$ and $\mathcal{S}^{'}$, LSTM with hidden size

1. hierarchical diversification: http://www.playbigdata.com/dou/hdiv
2. data and code: http://www.playbigdata.com/dou/DSSA/
3. Lemur: http://boston.lti.cs.cmu.edu/Services/clueweb09_batch/

of 50 as the cell of RNN. We set random permutation count as 10 in list-pairwise sampling. Similarly, λ of DSSA is tuned by cross validation. We also test the impact of different model settings and permutation counts on performance in Section 6.2 and Section 6.3 respectively. For HDSSA, we investigate both the balanced (denoted as HDSSA-B) and the unbalanced settings (denoted as HDSSA). To avoid overfitting, we use dropout [44] with probability 0.5 and L2 regularization. The dataset is split into three parts, namely training, validation, and testing. If the α-nDCG did not improve on the validation set after a certain number of epochs or the maximun epochs is reached, we stop the training process.

Similar to [15], we implement 18 relevance features and 6 diversity features, as listed in Table 5 and 6 respectively. We collect the candidate and retrieved documents of all queries and subtopics to generate the distributed representations.

6 Experimental Results

6.1 Overall Results

The overall results are shown in Table 7. We find that DSSA significantly outperforms all implicit and explicit baselines, including both unsupervised and supervised. The improvements are statistically significant (two-tailed paired t-test) for all metrics, except S-rec. The results clearly show the superiority of DSSA. Using hierarchical subtopics further improves all metrics, which demonstrates the usefulness of leveraging hierarchical subtopics.

(1) DSSA vs. unsupervised explicit methods. **DSSA outperforms unsupervised explicit methods (xQuAD, PM2, TxQuAD, TPM2, HxQuAD, and HPM2) on all the measures.** The relative improvement of DSSA over HxQuAD and HPM2, the best unsupervised explicit approaches, is up to 8.3% and 8.6% respectively in terms of α-nDCG. The relative improvement of HDSSA over HxQuAD and HPM2 is 10.9% and 11.2% respectively. This comparison shows the great advantage of using supervised method for learning the ranking function.

(2) DSSA vs. supervised implicit methods. **DSSA also outperforms supervised implicit methods (R-LTR, PAMM, R-LTR-NTN, and PAMM-NTN) by quite large margins.** The improvement over R-LTR-NTN and PAMM-NTN, the best supervised implicit approaches is up to 9.9% and 9.4% respectively on α-nDCG. This result demonstrates the utility of taking into account subtopics explicitly in supervised approaches. The improvements are similar to those observed between explicit approaches and implicit approaches in unsupervised framework [5], [6], [7], [8]. The combination of the two observations suggests that explicit modeling of subtopics can improve result diversification, whether it is in a supervised or unsupervised framework.

(3) HDSSA vs. DSSA. **HDSSA outperforms DSSA on all the measures.** Through paying attention to subtopics of different granularities, HDSSA has the potential to detect the most unsatisfied intents and keep balance between general and fine-grained intents. It also indicates that our framework is flexible enough to model hierarchical subtopics. HDSSA outperforms HDSSA-B, which indicates that scoring documents with different and tunable layer weights is beneficial. However, the improvement of the hierarchical subtopics over the flat list of subtopics is not significant. A possible reason is that we calculate attention and score for each subtopic separately, which fail to fully explore the dependency among the subtopics of different granularities. A promising direction is that the calculation of attention and score of parent subtopic is directly dependent on its child subtopics. Modeling the subtopics in a more unified and integrated way is our future work.

TABLE 7
Performance comparison of all methods. The best result is in bold. Statistically significant differences between DSSA and baselines are marked with various symbols. ★ indicates significant improvement over all baselines ($p < 0.05$).

Methods	ERR-IA	α-nDCG	NRBP	D♯-nDCG	Pre-IA	S-rec
Lemur①	.271	.369	.232	.424	.153	.621
ListMLE❶	.287	.387	.249	.430	.157	.619
C-GLS②	.288	.391	.246	.435	.153	.640
xQuAD③	.317	.413	.284	.437	.161	.622
TxQuAD④	.308	.410	.272	.441	.155	.634
HxQuAD⑤	.326	.421	.294	.441	.158	.629
PM2⑥	.306	.411	.267	.450	.169	.643
TPM2⑦	.291	.399	.250	.443	.161	.639
HPM2⑧	.317	.420	.279	.455	.172	.645
R-LTR❷	.303	.403	.267	.441	.164	.631
PAMM❸	.309	.411	.271	.450	.168	.643
R-LTR-NTN❹	.312	.415	.275	.451	.166	.644
PAMM-NTN❺	.311	.417	.272	.457	.170	.648
DSSA	.356★	.456★	.326★	.473★	.185★	.649①③❶
HDSSA-B	.366★	.465★	.335★	.475★	.186★	.648①③❶
HDSSA	**.369★**	**.467★**	**.337★**	**.478★**	**.187★**	**.653①③❶**

6.2 Effects of Different Settings

We conduct experiments with different settings of DSSA to investigate whether the performance is sensitive to these settings. Different aspects of settings are listed follow. For simplicity, when investigating the impact of each aspect, we keep other aspects the same as the settings specified in Section 5.3.

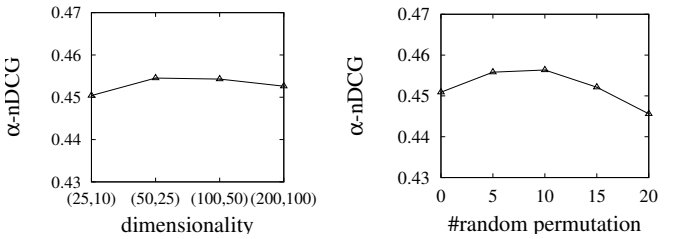
1) Representation generation methods: SVD, LDA, and doc2vec with window size of 5.
2) Implementation of vector interaction operations $\mathcal{A}'$ and $\mathcal{S}'$: "general" and "dot".
3) RNN cell: vanilla, GRU, and LSTM cell.
4) Dimensionality: we test several representative settings on the size of distributed representations E_d and E_q, the size of hidden state U as (25, 10), (50, 25), (100, 50), (200, 100).
5) Max-pooling: we experiment without using max-pooling (denoted as DSSA-RNN) in subtopic attention component.

The results are reported in Table 8. We can observe that DSSA does not heavily rely on specific settings. As for different representation generation methods, LDA has slightly better results. doc2vec could have been more appropriate if we had large datasets with more queries. The "general" operation yields slightly better results. A possible reason is that it is bilinear and thus is more powerful than "dot" to model the interaction. GRU and LSTM cells yield slightly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2810873, IEEE Transactions on Knowledge and Data Engineering

TABLE 8
Effects of different settings.

Methods	ERR-IA	α-nDCG	NRBP	D♯-nDCG	Pre-IA	S-rec
SVD	.348	.450	.315	.470	.184	.646
LDA	.356	.456	.326	.473	.185	.649
doc2vec	.351	.452	.318	.471	.184	.646
general	.356	.456	.326	.473	.185	.649
dot	.347	.450	.314	.470	.184	.647
vanilla	.354	.454	.322	.471	.184	.649
GRU	.357	.457	.326	.473	.185	.649
LSTM	.356	.456	.326	.473	.185	.649
DSSA-RNN	.342	.445	.306	.466	.172	.657
DSSA-RNNMP	.356	.456	.326	.473	.185	.649
HDSSA	.369	.467	.337	.478	.187	.653



(a) α-nDCG *w.r.t.* different size (b) α-nDCG *w.r.t.* different random permutation count

Fig. 7. Performance tendency of different settings.

better results than vanilla cell because of their ability of modeling long-term dependency. The difference is however small. This may be due to that with a limited number of training data, a model is unable to take advantage of its higher complexity to capture the fine-grained subtlety. Results with different size of distributed representation and hidden state shown in Figure 7(a) also indicate no strong correlation between performance and settings. α-nDCG remains above 0.45 using different sizes. The best performance is achieved using 100-dimensional representation and 50-dimensional hidden state. This suggests that high dimensionality may result in overfitting. Without using max-pooling, α-nDCG drops to 0.445, which demonstrates the usefulness of using max-pooling to enhance subtopic attention. The small differences between different settings suggest that DSSA is a stable and robust framework. Note that we use both distributed representations and relevance features, which are complementary to each other. This may be one of the reasons of the stability.

6.3 Effects of Different Optimization Methods

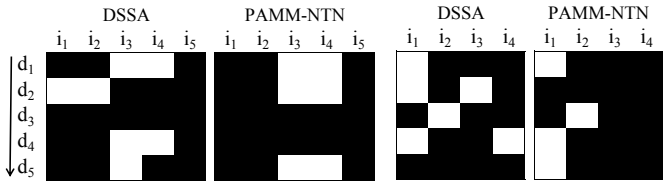
Results in Table 9 shows that list-pairwise is more effective than MLE and PAMM. This confirms our earlier intuition

TABLE 9
Effects of different optimization methods.

Methods	ERR-IA	α-nDCG	NRBP	D♯-nDCG	Pre-IA	S-rec
MLE	.349	.446	.315	.462	.176	.644
PAMM	.348	.445	.315	.463	.175	.644
list-pairwise	**.356**	**.456**	**.326**	**.473**	**.185**	**.649**

TABLE 10
Effects of different layers in HDSSA.

Methods	ERR-IA	α-nDCG	NRBP	D♯-nDCG	Pre-IA	S-rec
HDSSA-1	.356	.457	.323	.475	.186	.654
HDSSA-2	.364	.464	.333	**.480**	**.190**	**.655**
HDSSA	**.369**	**.467**	**.337**	.478	.187	.653



(a) ranking of query #58 (b) ranking of query #182

Fig. 8. Case study for DSSA and PAMM-NTN. White means relevant and black means irrelevant.

that list-pairwise optimization corresponds better to the situation of diversification ranking than the two other methods. Note that even using MLE or PAMM as optimization methods, DSSA could also achieve state-of-the-art performances, which confirms the effectiveness of our explicit learning framework from another perspective.

We vary the number of random permutations used in list-pairwise sampling from 0 to 20 to investigate its effect. As depicted in Figure 7(b), the performance does not heavily rely it. The best performance is achieved around 10. More permutations lead to lower effectiveness, which could be explained by model overfitting.

6.4 Effects of Different Layers in HDSSA

We further investigate the effects of using different layers in HDSSA in Table 10, where HDSSA-1 only uses first-level subtopics and HDSSA-2 only uses second-level subtopics. Basically, using both levels yields the best results and the second-level subtopics yield better performance than the first-level. This experimental result is consistent with learned parameter β for controlling layer importance: β_3(second-level) $>$ β_1(query) $>$ β_2(first-level). A possible explanation is that the second-level subtopics are more informative and specific than the first-level so that it can consider subtle variation of intent coverage (see Section 6.6 for further illustration). Note that using hierarchical subtopics outperforms both models using a single level of subtopics in terms of ERR-IA and α-nDCG. This indicates the effectiveness of modeling intents on different granularities.

6.5 Visualization and Discussion of DSSA

We visualize the ranking results of DSSA and the variation of subtopic attention to better understand why DSSA performs well.

We show the top 5 ranking results of query #58 and #182 in Figure 8 to illustrate why DSSA outperforms implicit learning methods. We choose PAMM-NTN as comparison method, which is the best existing learning method. In Figure 8, white means relevant and black means irrelevant. For query #58, DSSA ranks a document relevant to subtopics i_3 and i_4 first and a document relevant to i_1 and i_2

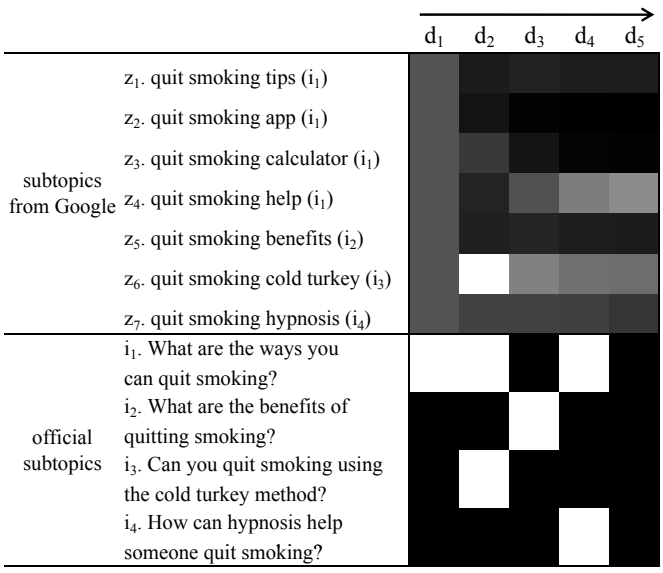
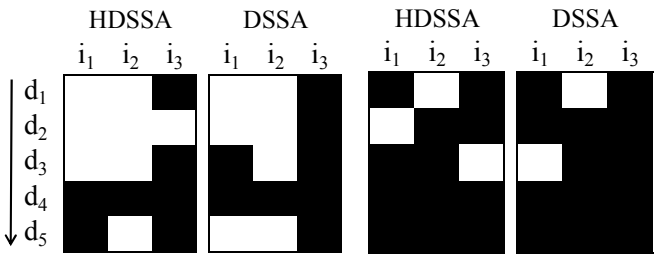


Fig. 9. Subtopic attention variation of query #182. The top part is attention and the bottom part is relevance judgment.



(a) ranking of query #103 (b) ranking of query #139

Fig. 10. Case study for HDSSA and DSSA. White means relevant and black means irrelevant.

at the second position, while the first two documents of PAMM-NTN cover the same subtopics. Note that there is no document covering i_5 in the candidate set. For query #182, DSSA successively chooses documents that cover i_1, i_3, i_2, and i_4. One additional intent is satisfied at every position. PAMM-NTN, however, just covers i_1 and i_2 by top 5 documents, which is obviously not optimal. We see that the unequal and varied subtopic attention is capable of discovering unsatisfied subtopics at different positions, which eventually leads to more subtopic coverage.

To study attention mechanism, we further visualize the variation of subtopic attention of top 5 documents of query #182, namely "quit smoking", which has 4 official subtopics (i_1 to i_4), as shown in Figure 9. The top part is subtopic attention variation and the bottom part is relevance judgment. For attention part, the darker the cell is, the lower the attention (weight) on this subtopic is. Note that we actually leverage query suggestions of Google (z_1 to z_7) to serve as subtopics, which do not match official ones exactly. We manually align subtopics mined from Google to official ones. At the beginning, all the subtopics have equal attention. The first selected document d_1 is relevant to i_1, i.e. to the Google subtopics z_1, z_2, z_3 and z_4. We see that the attention to these latter decreases at second position. Then the document d_2 is selected, which is relevant to uncovered i_3. We see that the attention to the corresponding z_6 begins to diminish from the third position. d_3 and d_4 satisfy additional i_2 and i_4 respectively, which leads to the reduction of attention on z_5 and z_7 at the following position. The subtopic attention, initialized as uniform distribution, ends up with more emphasis on z_4, z_6, and z_7. This example illustrates how the unequal and varied attention drives the model to emphasize different subtopics at different positions, which is crucial in explicit diversification. This example also shows a potential problem inherent for any method using automatically discovered subtopics: those topics may be different from the ones defined by human assessors. Equal distribution is assumed on all the subtopics z_i. However, this implies an unequal distribution among the manually defined subtopics (more emphasis is put on i_1). Assuming an equal distribution at the beginning may not necessarily be the best approach. We will deal with this problem in our future work.

6.6 Visualization and Discussion of HDSSA

We also visualize the ranking results of HDSSA and the variation of subtopic attention to investigate the effect of using hierarchical subtopics.

We show the top 5 ranking results of query #103 and #139 in Figure 10 to illustrate the superiority of HDSSA over DSSA. For query #103, HDSSA covers all subtopics within 5 results, while DSSA fails to satisfy i_3. For query #139, HDSSA successfully selects 3 adjacent documents that cover i_2, i_1, and i_3 respectively, while DSSA ignores i_3. The above cases demonstrate the usefulness of the hierarchical subtopics, which can detect unsatisfied intents based on different granularities.

In Figure 11, we visualize the variation of subtopic attention of HDSSA on query #103 ("madam cj walker"), which has 3 official subtopics listed at the bottom. We collected 9 first-level subtopics and 35 second-level subtopics to construct the hierarchical structure. The first column of attention (attention of d_1) is the initial weight of all subtopics, which is allocated in a top-down manner so that each layer obtains equal attention at the beginning. After selecting a document satisfying i_1 and i_2, the attention of next position should turn to i_3 which is "Madam C. J. Walker's involvement in the political and social issues". If we only use the first-level subtopics which are not specific enough, we cannot find suitable subtopics to match i_3. However in second-level subtopics, $z_{1,6,4}$ ("timeline madam cj walker achievements") has the potential to cover her social and political achievements. Using hierarchical subtopics is useful to perceive the subtle variation of unsatisfied intents. So it can further improve the diversity of the search results. The attention on different levels and subtopics keeps changing throughout document selection process in Figure 11, which is beneficial to introduce diversity on different granularities. In fact, the latent intents of real users are much more than 3 official ones. Hierarchical subtopics keep balance between crude and fine-grained intents, which have potential to achieve more diversity even beyond the official judgement.

Fig. 11. Subtopic attention variation of query #103. The top part is attention and the bottom part is relevance judgment. To save space, we use "~" to replace the query string "madam cj walker" in all subtopics.

7 Conclusions

In this paper, we propose a general learning framework DSSA to model subtopics explicitly for search result diversification. Based on the sequence of selected documents, unequal and varied subtopic attention is calculated, driving the model to emphasize different subtopics at different positions. This is the first time that attention mechanism is used to model the process. We further instantiate DSSA using RNN and max-pooling to handle both distributed representations and relevance features, which outperforms significantly the existing approaches. The results confirm that modeling subtopics explicitly in a learning framework is beneficial and effective and this also avoids heuristically defined functions and parameters. Through using hierarchical subtopics, performance is further improved because of the consideration of subtopics of different granularities. However, accurately modeling the interaction among documents and subtopics is still challenging. There are many other more complex implementations besides our particular way, which will be investigated in future work. The proposed model contains a number of parameters to be learned. This requires a large number of training data. Collecting more training data to fully unlock the potential of the model is another direction. Finally, this work only deals with the learning of a ranking function, assuming that subtopics have been obtained in advance and document and query representations have already been created. In practice, mining subtopics and learning these representation are another interesting aspects, which could be incorporated into our framework, provided with sufficient training data.

Acknowledgments

Zhicheng Dou is the corresponding author. This work was funded by the National Natural Science Foundation of China under Grant No. 61502501 and 61502502, the National Key Basic Research Program (973 Program) of China under Grant No. 2014CB340403, and the Beijing Natural Science Foundation under Grant No. 4162032.

References

[1] R. L. Santos, J. Peng, C. Macdonald, and I. Ounis, "Explicit search result diversification through sub-queries," in *ECIR*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 87–99.

[2] R. L. Santos, C. Macdonald, I. Ounis *et al.*, "Search result diversification," *Foundations and Trends® in Information Retrieval*, vol. 9, no. 1, pp. 1–90, 2015.

[3] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *SIGIR*. New York, NY, USA: ACM, 1998, pp. 335–336.

[4] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *WSDM*. New York, NY, USA: ACM, 2009, pp. 5–14.

[5] R. L. Santos, C. Macdonald, and I. Ounis, "Exploiting query reformulations for web search result diversification," in *WWW*. New York, NY, USA: ACM, 2010, pp. 881–890.

[6] V. Dang and W. B. Croft, "Diversity by proportionality: An election-based approach to search result diversification," in *SIGIR*. New York, NY, USA: ACM, 2012, pp. 65–74.

[7] V. Dang and B. W. Croft, "Term level search result diversification," in *SIGIR*. New York, NY, USA: ACM, 2013, pp. 603–612.

[8] S. Hu, Z. Dou, X. Wang, T. Sakai, and J.-R. Wen, "Search result diversification based on hierarchical intents," in *CIKM*, ser. CIKM '15. New York, NY, USA: ACM, 2015, pp. 63–72.

[9] H.-T. Yu and F. Ren, "Search result diversification via filling up multiple knapsacks," in *CIKM*. New York, NY, USA: ACM, 2014, pp. 609–618.

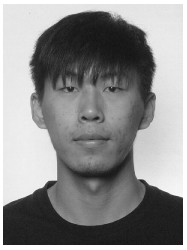
[10] J. Yi and F. Maghoul, "Query clustering using click-through graph," in *WWW*. New York, NY, USA: ACM, 2009, pp. 1055–1056.

[11] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza, "Query recommendation using query logs in search engines," in *Current Trends in Database Technology EDBT 2004 Workshops*, 2004, pp. 588–596.

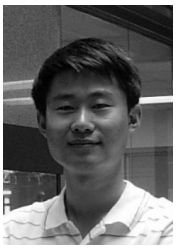
[12] Z. Zhang and O. Nasraoui, "Mining search engine query logs for query recommendation," in *WWW*. New York, NY, USA: ACM, 2006, pp. 1039–1040.

[13] A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall, S. Riezler, and E. Alfonseca, "Generalized syntactic and semantic models of query reformulation," in *SIGIR*. New York, NY, USA: ACM, 2010, pp. 283–290.

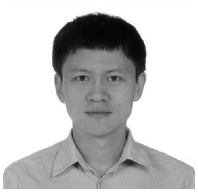
[14] Y. Yue and T. Joachims, "Predicting diverse subsets using structural svms," in *ICML*. New York, NY, USA: ACM, 2008, pp. 1224–1231.
[15] Y. Zhu, Y. Lan, J. Guo, X. Cheng, and S. Niu, "Learning for search result diversification," in *SIGIR*. New York, NY, USA: ACM, 2014, pp. 293–302.
[16] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Learning maximal marginal relevance model via directly optimizing diversity evaluation measures," in *SIGIR*. New York, NY, USA: ACM, 2015, pp. 113–122.
[17] ——, "Modeling document novelty with neural tensor network for search result diversification," in *SIGIR*. New York, NY, USA: ACM, 2016, pp. 395–404.
[18] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *NIPS*, 2014, pp. 2204–2212.
[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.
[20] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015, pp. 1412–1421.
[21] X. Wang, Z. Dou, T. Sakai, and J.-R. Wen, "Evaluating search result diversity using intent hierarchies," in *SIGIR*, ser. SIGIR '16. New York, NY, USA: ACM, 2016, pp. 415–424. [Online]. Available: http://doi.acm.org/10.1145/2911451.2911497
[22] G. Zuccon, L. Azzopardi, D. Zhang, and J. Wang, "Top-k retrieval using facility location analysis," in *ECIR*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 305–316.
[23] K. D. Naini, I. S. Altingovde, and W. Siberski, "Scalable and efficient web search result diversification," *ACM Trans. Web*, vol. 10, no. 3, pp. 15:1–15:30, Aug. 2016.
[24] B. Carterette, "An analysis of np-completeness in novelty and diversity ranking," in *ICTIR*, 2009, pp. 200–211.
[25] P. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. P. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *CIKM*, 2013, pp. 2333–2338.
[26] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *SIGIR*. New York, NY, USA: ACM, 2015, pp. 373–382.
[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[28] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014, pp. 1188–1196.
[29] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.
[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[31] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *NIPS*, 2013, pp. 926–934.
[32] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.
[33] J. I. Marden, *Analyzing and modeling rank data*. CRC Press, 1996.
[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.
[35] J. Callan, M. Hoy, C. Yoo, and L. Zhao, "Clueweb09 data set," http://boston.lti.cs.cmu.edu/Data/clueweb09/, 2009.
[36] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan, "Expected reciprocal rank for graded relevance," in *CIKM*. New York, NY, USA: ACM, 2009, pp. 621–630.
[37] 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 *SIGIR*, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 659–666.
[38] C. L. A. Clarke, M. Kolla, and O. Vechtomova, "An effectiveness measure for ambiguous and underspecified queries," in *ICTIR*, 2009, pp. 188–199.
[39] T. Sakai and R. Song, "Evaluating diversified search results using per-intent graded relevance," in *SIGIR*. New York, NY, USA: ACM, 2011, pp. 1043–1052.
[40] T. Sakai, Z. Dou, T. Yamamoto, Y. Liu, M. Zhang, R. Song, M. Kato, and M. Iwata, "Overview of the ntcir-10 intent-2 task." in *NTCIR*, 2013.
[41] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. P. Kato, H. Ohshima, and K. Zhou, "Overview of the ntcir-11 imine task." in *NTCIR*. Citeseer, 2014.
[42] C. X. Zhai, W. W. Cohen, and J. Lafferty, "Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval," in *SIGIR*. New York, NY, USA: ACM, 2003, pp. 10–17.
[43] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank: Theory and algorithm," in *ICML*. New York, NY, USA: ACM, 2008, pp. 1192–1199.
[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.



Zhengbao Jiang is a graduate student from School of Information at Renmin University of China. His research interests include information retrieval, natural language processing, big data management, and data mining.



Zhicheng Dou is an associate professor at School of Information, Renmin University of China. He received his Ph.D. and B.S. degrees in computer science and technology from the Nankai University in 2008 and 2003, respectively. He worked at Microsoft Research as a researcher from July 2008 to September 2014. His research interests include information retrieval, data mining, and big data analytics. His homepage is http://www.playbigdata.com/dou/.



Wayne Xin Zhao received the PhD degree from Peking University in 2014. He is currently an assistant professor in the School of Information, Renmin University of China. His research interests are web text mining and natural language processing. He has published several referred papers in international conferences journals such as ACL, EMNLP, COLING, SIGIR, etc.

Jian-Yun Nie is a professor at the Universit de Montral, Canada. He has published more than 150 papers in information retrieval and natural language processing in journals and conferences. He has served as a general cochair of the ACM-SIGIR conference in 2011. He is currently on the editorial board of seven international journals. He has been an invited professor and researcher at several universities and companies.

Ming Yue is a graduate student from School of Information at Renmin University of China. His research interests include information retrieval, natural language processing, and big data management.

Ji-Rong Wen is a professor in Renmin University of China. He received B.S. and M.S. degrees from Renmin University of China, and received his Ph.D. degree in 1999 from the Chinese Academy of Science. He is a senior researcher and research manager at Microsoft Research from 2000 to 2014. His main research interests are web data management, information retrieval (especially web IR), and data mining.