Diversifying Search Results using Self-Attention Network

Xubo Qin[2], Zhicheng Dou[1], and Ji-Rong Wen[3,4]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
[4]Key Laboratory of Data Engineering and Knowledge Engineering, MOE
qratosone@live.com,dou@ruc.edu.cn,jirong.wen@gmail.com

ABSTRACT

Search results returned by search engines need to be diversified in order to satisfy different information needs of different users. Several supervised learning models have been proposed for diversifying search results in recent years. Most of the existing supervised methods greedily compare each candidate document with the selected document sequence and select the next local optimal document. However, the information utility of each candidate document is not independent with each other, and research has shown that the selection of a candidate document will affect the utilities of other candidate documents. As a result, the local optimal document rankings will not lead to the global optimal rankings. This problem is especially serious when the selected document sequence is short or empty in the early stage of ranking, since almost any of the candidate documents can be estimated as "satisfying new user intents" following on the selected document sequence. In this paper, we propose a new supervised diversification framework to address this issue. Based on a self-attention encoder-decoder structure, the model can take the whole candidate document sequence as input, and simultaneously leverage both the novelty and the subtopic coverage of the candidate documents. Comparing with existing supervised methods, this framework can model the interactions between all candidate documents and return their diversification scores based on the whole candidate document sequence. Experimental results show that our proposed framework outperforms existing methods. These results confirm the effectiveness of modeling all the candidate documents for the overall novelty and subtopic coverage globally, instead of comparing every single candidate document with the selected sequence document selection.

KEYWORDS

Search Result Diversification; Self Attention

1 INTRODUCTION

Research shows that most queries issued by users are short [1–4], and these queries could be ambiguous or vague. For example, a user who issues the query "Java" may expect a result about "Java island", while another user with the same query may want information about "JAVA programming language". Even for a same user, she may also want diversified results which cover different aspects of the information she is looking for (for example, seeking for different cooking methods for "roast beef"). Search result diversification is proposed to solve the above problem by returning a diversified document list that can satisfy different information needs.

Existing search result diversification models can be divided into supervised and unsupervised models depending on whether supervised learning approaches are used. Most of the traditional approaches to search result diversification are unsupervised and they are based on handcrafted features and functions [5–9]. While in recent years, more and more researchers tried to use machine learning methods in search result diversification in order to learn an optimized ranking function automatically [10–14]. To generate diversified results, these methods either explicitly model subtopic coverage of the results [6–9, 14] (i.e., explicit approaches), or directly reduce result redundancy by comparing document-document similarity regardless the use of subtopics [5, 10–13] (i.e., implicit approaches).

To simplify the problem and accelerate the online ranking, existing methods usually formalize the diverse ranking process as the greedy document sequential selection. Those methods compare each of the candidate document with the selected document sequence and select the best candidate document which can provide the maximum additional information utility for the current selected document sequence. However, researchers [15] have already proved that this greedy document selection mechanism may not lead to the global optimal rankings. This is because the previous methods only model the interaction between every single candidate document and the selected document sequence, ignoring the candidate document's interactions with other candidate documents. While the information utilities of all the candidate documents are not independent, when a candidate document is selected, the utilities of other documents will be affected. As a result, the sequential selection of every locally optimal document may not lead to a global optimal document ranking.

This problem may be even more serious when the selected sequence is short or empty in the early stage of ranking. For example, assuming there are three candidate documents d_1, d_2, d_3, with d_1 covering the subtopic q_1, d_2 covering q_2, q_3 and d_3 covering q_1, and the three documents has got similar relevance scores to the given query. A greedy selection based model may select d_1 for the first ranking position. Since the selected sequence is empty, any of the candidate documents can be seen as a "diversified document" after the empty selected sequence. However, the diverse ranking task aims to satisfy more user intents at former position, in the view

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411914

of intent-based diversification metrics e.g. $\alpha-$nDCG, d_2 may be a better selection comparing with d_1. In order to achieve the global optimal ranking, the model has to search all the ranking space, which is an NP-hard problem. Feng et al. [15] proposed the M2DIV model with Monte-Caro Tree Search (MCTS) in order to explore a larger ranking space and raise the probability of selecting the global optimal document ranking. However, as a deep reinforced learning model with MCTS, the M2DIV is difficult to train since MCTS is so time consuming that the M2DIV propose another raw policy without MCTS [15] in adaption to some online ranking tasks.

In this paper, we propose a new search result diversification framework to address the issues above. This framework can model all the candidate documents as a whole sequence, and leverage both the novelty and the subtopic coverage of every candidate document simultaneously. More specifically, we use a self-attention based encoder-decoder structure to model the interactions between candidate documents and subtopics. We call this framework Diversity Encoder with Self-Attention (DESA). The self-attention network has been widely used in order to learn the context-aware representations of words. Comparing with CNN and RNN, the self-attention network allows every item in the whole sequence to interact directly with each other simultaneously, and it can learn the long-range dependency well. In the task of search result diversification, we use the self-attention network to build an encoder-decoder framework for modeling the candidate document sequence and the subtopics. The encoder component can learn the document interactions globally in the whole candidate document sequence, indicating the novelty of every candidate document. And the decoder component can learn the matching distributions between the documents and the subtopics. Instead of comparing every single candidate document with the selected document sequence, the framework will model the whole candidate document sequence and jointly return the ranking scores of all the candidate documents in the ranking task. We also give a theoretical analysis of how self-attention mechanism works in the task of search result diversification. Since self-attention network is suitable for parallel computing, the proposed DESA model is easy to train. Experimental results with the TREC Web Track data show that the model outperforms the state-of-the-art diversification models significantly.

The contributions of the paper are summarized as follows:
(1) We propose a framework which can take the whole candidate document sequence as input and model the interactions between all the candidate documents for measuring their information utilities globally. Comparing with the greedy sequential selection approaches, this framework will get a higher probability of achieving the global optimal ranking.
(2) More specifically, we use a self-attention based encoder-decoder structure and model both the novelty and the subtopic coverage of the candidate documents. This self-attention based model is suitable for parallel computing and can be trained in limited time.
(3) We theoretically analyze why self-attention is suitable to the search result diversification task. Experimental results verify the effectiveness of the proposed model.

2 RELATED WORK

2.1 Implicit and Explicit Diversification Models

Existing search result diversification models can be divided into implicit and explicit ones depending on modeling the user intents (represented as subtopics) explicitly. The implicit ranking model calculates the similarity between every candidate document and the previous selected documents, and assume that the more dissimilar the candidate document is to the selected documents, the more diversified it will be. The most typical implicit model is the MMR (Max Margin Relevance) [5] model:

$$\text{Score}_{\text{MMR}} = \lambda\text{score}(d_i, q) - (1-\lambda)\max_{d_j \in S}\text{sim}(d_i, d_j),$$

where $\text{score}(d_i, q)$ is relevance score of the current document candidate d_i and the given query q, $\text{sim}(d_i|d_j)$ is the similarity between d_i and the selected document d_j in the selected set S. In the view of the MMR model, the less similar the candidate document is with the selected documents, the more diversified it will be. The final ranking score of the candidate document is the linear combination of the relevance score and the novelty score. Inheriting the spirit of MMR, researchers have also proposed supervised methods, such as SVM-DIV [10], R-LTR [11]), PAMM [12], and PAMM-NTN [13]), for learning a better document similarity function automatically.

The explicit approaches model the underlying user intents of the issued query, those intents are represented as subtopics. In the view of explicit diverse ranking, comparing with the selected document sequence, a diversified candidate document should cover as many new subtopics under the given query which has not been covered by the selected document as possible. Nowadays both unsupervised and supervised explicit approaches are proposed e.g. xQuAD [7], PM2 [8], HxQuAD/HPM2 [9] and DSSA [14].

Those existing approaches used greedy document sequential selection. They compare every single candidate document with the selected document sequence, and choose the locally optimal document one-by-one to fill in the document ranking list. Since the information utilities of the candidate documents are not independent, this strategy may not lead to global optimal rankings. Based on the reinforced learning approach MDP-DIV [14], Feng [15] proposed the M2DIV model with the Monte-Caro Tree Search (MCTS) to search a larger ranking space and minimize the gap between the local optimal and global optimal rankings. However, M2DIV is difficult to train since MCTS is time consuming [15], and M2DIV only models the document novelty, ignoring the subtopic coverage.

2.2 Self-Attention in Information Retrieval

The self-attention mechanism is a kind of attention mechanism modeling each position in a sequence and compute the representation for each hidden state of the sequence. Recently, the models fully based on self-attention mechanism (denoted as self-attention network), such as Transformer [16] in the Neural Machine Translation (NMT) task, have achieved great successes on many NLP tasks. Researchers have used self-attention networks, e.g. GPT [17], BERT [18] and ERNIE [19], as alternatives to RNNs and CNNs in many NLP tasks. However, to the best of our knowledge, only a few researchers [20, 21] have tried to use the self-attention network in the information retrieval tasks. There are no self-attention based models designed for the search result diversification task.

Table 1: Notations used in this paper

Notation	Description
q	the input query
$\mathcal{I}$,q_i	subtopics corresponding to q, $q_i \in \mathcal{I}$
$\mathcal{D}$	the candidate document sequence
$\boldsymbol{D}$	embeddings of the candidate document sequence
$\boldsymbol{I}$	embeddings of all the subtopics
$\mathcal{R}$	the returned document rank list
x_q	document relevance features to the query
x_{q_i}	document relevance features to the i-th subtopics
$\boldsymbol{d}_t$	initial document embedding for the t-th document
$\boldsymbol{q}_t$	initial subtopic embedding for the t-th subtopic
$\boldsymbol{h}_t^{\text{enc}}$	the encoder output for t-th document
$\boldsymbol{h}_t^{\text{dec}}$	the decoded output for t-th document
s_{q_i}	relevance score of the document to the i-th subtopic
$\boldsymbol{v}_{d_t,q,q_i}$	document vector for generating the ranking score
$[;]$	concatenation operation

In this paper, we propose using self-attention to model the interactions of candidate documents and subtopics for search result diversification.

3 THE DIVERSIFICATION FRAMEWORK

In this section, we will first describe the overall structure of the self-attention based diversification framework DESA, then introduce the details of each component and the optimization process. We also propose a theoretical analysis to explain why self-attention is suitable to the search result diversification task. Finally, we compare DESA with existing models and discuss their relationships.

3.1 Problem Formulation

Table 1 shows the notations and their descriptions used in this paper. Given a query q, we have k subtopics $\mathcal{I}$ representing different user intents, and q_i is the $i-$th subtopic ($i \in [1, k]$ and $q_i \in \mathcal{I}$). Suppose $\mathcal{D}$ is a list of candidate documents for q, the target of search result diversification is to return a new ranked document list $\mathcal{R}$ based on initial ad-hoc rank list $\mathcal{D}$, where diverse documents covering different subtopics are ranked higher in $\mathcal{R}$ and redundant documents are ranked lower.

Different from the ad-hoc retrieval task which is solely designed for returning relevant documents, search result diversification needs to consider both the relevance of each single document and the similarity between them. As introduced in Section 1, most existing diversification models used the greedy selection approach: they iteratively select the next best document by evaluating the relevance of each remaining document and its novelty bring to the results based on the list of documents which are already selected in early iterations.

3.2 DESA: the Overall Framework

In Figure 1, we show the overall structure of our proposed DESA framework. Different from existing approaches which greedily select the next best documents and sequentially generate $\mathcal{R}$, DESA calculates all diverse ranking scores for each candidate document simultaneously, then sorts the documents based on the scores and gets the diverse ranking list directly. i.e., DESA directly gets a list of ranking scores $\mathcal{S}_D$ by:

$$\mathcal{S}_D = \text{DESA}(\mathcal{D}, q, \mathcal{I}). \tag{1}$$

DESA takes the whole candidate document sequence as input and models the interactions between all the candidate documents for measuring their information utilities globally. Comparing with the greedy sequential selection approaches, this framework will get a higher probability of achieving the global optimal ranking. More specifically, we use an encoder-decoder structure based on self attention to model the relationship between each document in $\mathcal{D}$ and each subtopic $q_i \in \mathcal{I}$. The encoder component takes the whole candidate document sequence $\mathcal{D}$ as input, and returns the representations of all the documents simultaneously. After interacting with every other candidate document, the document representations can indicate the novelty or dissimilarity of a document. Then the decoder component takes both document sequence and subtopics as input, returning the decoded document representations indicating the subtopic coverage of the documents. Finally, those representations will be used by a learning-to-rank function to judge the diverse ranking scores of the documents. Key components of the framework are briefly introduced as follows.

(1) Document Representations. Suppose d_t is the t-th document in $\mathcal{D}$, $\boldsymbol{d}_t$ is the initial distributed representation of the document d_t. In order to avoid overfitting, we follow [22] and use unsupervised methods doc2vec [23] to generate the initial document representations instead of building the document representations automatically.

(2) Self-attention Encoder. The self-attention encoder in the framework of DESA takes $\boldsymbol{D}$ as input and returns the representations $\boldsymbol{H}_D^{\text{enc}}$ of the whole document sequence. The encoder also takes the embeddings of subtopics $\boldsymbol{I}$ as input and returns the representations $\boldsymbol{H}_I^{\text{enc}}$ for all the subtopic. i.e., we have:

$$\boldsymbol{H}_D^{\text{enc}} = \text{SelfAttnEnc}(\boldsymbol{D}),$$
$$\boldsymbol{H}_I^{\text{enc}} = \text{SelfAttnEnc}(\boldsymbol{I}),$$

where the self-attention encoder is denoted as SelfAttnEnc, which will be introduced in the next section.

(3) Self-attention Decoder. The decoder will take the encoded representation of document sequence $\boldsymbol{H}_D^{\text{enc}}$ and subtopics $\boldsymbol{H}_I^{\text{enc}}$ as an input, and return the decoded representations $\boldsymbol{H}_D^{\text{dec}}$ for all the documents. These decoded representations model the subtopic coverage of the documents. This step can be described as the following equations, and the decoder is denoted as SelfAttnDec:

$$\boldsymbol{H}_D^{\text{dec}} = \text{SelfAttnDec}(\boldsymbol{H}_D^{\text{enc}}, \boldsymbol{H}_I^{\text{enc}}),$$
$$\boldsymbol{h}_t^{\text{enc}} = \boldsymbol{H}_D^{\text{enc}}[\text{index(t)}],$$
$$\boldsymbol{h}_t^{\text{dec}} = \boldsymbol{H}_D^{\text{dec}}[\text{index(t)}],$$

where $index(t)$ is the operation of getting the vector at index t. For the $t-$th document, the encoded and decoded representations $\boldsymbol{h}_t^{\text{enc}}$ and $\boldsymbol{h}_t^{\text{dec}}$ are used to get the document's ranking score.

(4) Subtopic Document Ranking. We use learning-to-rank to learn the relevance score of the i-th subtopic s_{q_i} through the subtopic relevance features $\boldsymbol{x}_{q_i}$:

$$s_{q_i} = \boldsymbol{x}_{q_i}^T \boldsymbol{w}_r (i \in [1, k]).$$

Here $\boldsymbol{w}_r$ is a learnable parameter. We use the same relevance features as the previous work [14] for $\boldsymbol{x}_q$ and $\boldsymbol{x}_{q_i}$, including BM25, TF-IDF, language model scores, Page Rank, the numbers of incoming links and outgoing links, et al. More details about these features can be found in [14] and we omit the details due to space limitation. In the future, we plan to explore more neural-based features.

(5) The Final Ranking. The summarized document feature vectors $\boldsymbol{v}_{d_t,q,q_i}^T$ are concatenated by the following components: the query relevance features $\boldsymbol{x}_q$, the encoded document representation $\boldsymbol{h}_t^{\text{enc}}$ and decoded document representation $\boldsymbol{h}_t^{\text{dec}}$, and the relevance scores of all the k subtopics $[s_{q_1}, \ldots, s_{q_k}]$. Note that we use the same set of ranking features for query q with those used for subtopics as introduced in step (4). Given the document feature vectors $\boldsymbol{v}_{d_t,q,q_i}^T$, we use learning-to-rank to train the final ranking models. The ranking model then returns the ranking score $s_t \in \mathcal{S}_D$ for the $t-$th document d_t. We then generate the diversified ranking list $\mathcal{R}$ by sorting all the candidate documents with their ranking scores in $\mathcal{S}_D$. Recall that different from those greedy sequential selection based models, DESA doesn't depend on the sequential selection process. This is similar to some ad-hoc ranking models such as SetRank [20].

This process is formulated as the following equations:

$$\boldsymbol{v}_{d_t,q,q_i} = [\boldsymbol{x}_q; \boldsymbol{h}_t^{\text{enc}}; \boldsymbol{h}_t^{\text{dec}}; s_{q_1}, \ldots, s_{q_k}], \tag{2}$$

$$s_t = \boldsymbol{v}_{d_t,q,q_i}^T \boldsymbol{w}_v. \tag{3}$$

where $\boldsymbol{w}_v$ is a learnable parameter, $[;]$ means the concatenation.

In the remaining part of this section, we will introduce the components in details.

3.3 The Self-Attention Encoder Component

The whole self-attention encoder component denoted as SelfAttnEnc takes all the candidate document embeddings as a whole document sequence $\boldsymbol{D}$, and returns all the document representations as a whole matrix denoted as $\boldsymbol{H}_D^{\text{enc}}$ in parallel. The representations will indicate the novelty of each document comparing with other candidate documents. In this section we will introduce the implementation of self-attention encoder in details.

3.3.1 The Attention Function. In the search result diversification task, the vector representations of documents are used as input to the self-attention layer. Different from RNN, self-attention network will not model the sequence information explicitly, so the standard Transformer structure also includes an optional component of positional encoding to incorporate the sequence information. Here we deploy an optional learnable position embeddings for capturing the sequence information of the documents and concatenate them with the document embeddings.

We use the multi-layer encoder block of the Transformer to implement DESA's self-attention component, based on the scaled dot-product attention function denoted as Attn follows:

$$\text{Attn}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{q}\boldsymbol{K}^T}{\sqrt{d}})\boldsymbol{V}. \tag{4}$$

where $\boldsymbol{q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ denote the query, key and value matrices of the attention function. It should be addressed that the concept "query" here represents the query in dot-product attention, which is not the "query" in information retrieval. In search result diversification tasks, the model will take the sequence of document representations $\boldsymbol{D}$ as an input, and the query matrix can be defined as $\boldsymbol{q} = \boldsymbol{D}$.

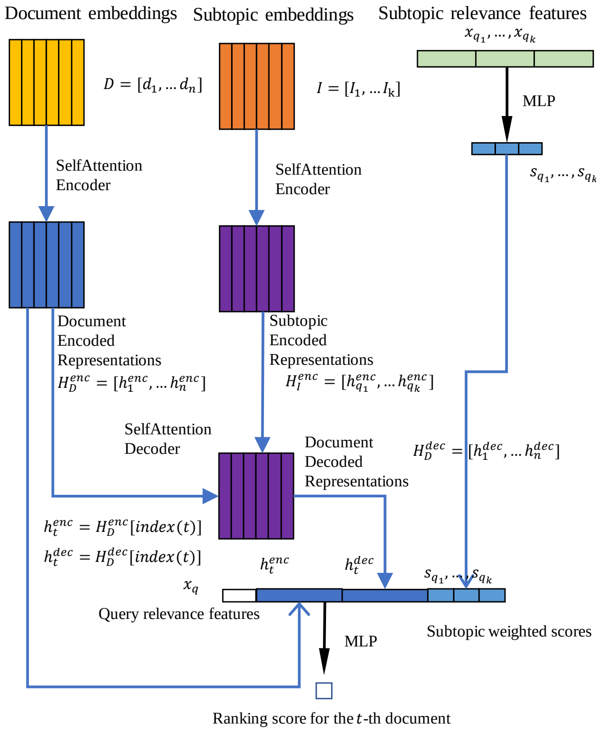
Document embeddings
Subtopic embeddings
Subtopic relevance features
$x_{q_1}, \ldots, x_{q_k}$
$D = [d_1, \ldots d_n]$
$I = [I_1, \ldots I_k]$
MLP
SelfAttention Encoder
SelfAttention Encoder
$s_{q_1}, \ldots, s_{q_k}$
Document Encoded Representations
$H_D^{enc} = [h_1^{enc}, \ldots h_n^{enc}]$
Subtopic Encoded Representations
$H_I^{enc} = [h_{q_1}^{enc}, \ldots h_{q_k}^{enc}]$
SelfAttention Decoder
Document Decoded Representations
$H_D^{dec} = [h_1^{dec}, \ldots h_n^{dec}]$
$h_t^{enc} = H_D^{enc}[index(t)]$
$h_t^{dec} = H_D^{dec}[index(t)]$
h_t^{enc}
h_t^{dec}
$s_{q_1}, \ldots, s_{q_k}$
x_q
Query relevance features
MLP
Subtopic weighted scores
Ranking score for the t-th document

Figure 1: The Overall Structure of DESA. The framework takes the whole candidate document sequence and subtopics together as input, and returns the encoded and decoded representations of every candidate document simultaneously. For the $t-$th document d_t, the learning-to-rank function takes the query relevance features x_q, the encoded and decoded representation h_t^{enc} and h_t^{dec} and subtopic relevance scores s_{q_i} as input, and returns the ranking score s_t

3.3.2 The Multi-Head Attention Component. Following by some previous work e.g. SetRank [20], we use the multi-head strategy in order to learn multiple aspects of different documents. The multi-head attention strategy denoted as MultiHead will first project the inputs $\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}$ into h different heads with the dimension $\hat{E} = E/h$:

$$\text{MultiHead}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) = [\boldsymbol{a}_1; \ldots; \boldsymbol{a}_h], \tag{5}$$

where $\boldsymbol{a}_i$ is defined by:

$$\boldsymbol{a}_i = \text{Attn}(\boldsymbol{q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V), i \in [1, h]. \tag{6}$$

Here all those $\boldsymbol{W}$ parameters are learnable. Previous research [20] has shown that using the multi-head strategy may help the self-attention network to learn better document similarity distribution at multi aspects. For the self-attention, $\boldsymbol{q} = \boldsymbol{K} = \boldsymbol{V}$.

3.3.3 *The Overall Structure of the Self-Attention Encoder.*

The overall structure of the encoder component is a multi-layer stack of multi-head self-attention block. Similar as the original Transformer encoder block, each of those self-attention encoder layers contains a dropout layer and a fully connected feed-forward network (denoted as FeedForward) with ReLU function as activation function. The i−th layer of the block is denoted as MSB_i and the encoder component SelfAttnEnc with L layers can be described as follows:

$$\mathrm{SelfAttnEnc}(\boldsymbol{D}) = \mathrm{MSB}_{\mathrm{L}}(\mathrm{MSB}_{\mathrm{L-1}}(\ldots \mathrm{MSB}_1(\boldsymbol{D})), \quad (7)$$

$$\mathrm{MSB}(\boldsymbol{H}_{\mathrm{prev}}) = \mathrm{LayerNorm}(\boldsymbol{X} + \mathrm{FeedForward}(\boldsymbol{X})), \quad (8)$$

$$\boldsymbol{X} = \mathrm{LayerNorm}(\boldsymbol{H}_{\mathrm{prev}} + \mathrm{MultiHead}(\boldsymbol{H}_{\mathrm{prev}}, \boldsymbol{H}_{\mathrm{prev}}, \boldsymbol{H}_{\mathrm{prev}})), \quad (9)$$

where LayerNorm denotes the layer normalization operation [24], $\boldsymbol{H}_{\mathrm{prev}}$ is the output hidden state matrix of the previous encoder layer, and $\boldsymbol{H}_{\mathrm{prev}} = \boldsymbol{D}$ is for the first layer.

After multi-layers of multi-head attention interactions, the output hidden state of n input documents $\boldsymbol{H}_{\mathrm{output}} = [\boldsymbol{h}_1^{\mathrm{enc}}, \ldots, \boldsymbol{h}_n^{\mathrm{enc}}]$ can be used as the encoded document representations $\boldsymbol{H}_D^{\mathrm{enc}}$. This representation can indicate the novelty of a document, and the learning-to-rank function can take this representations to judge if a candidate document is novel or redundant comparing with other candidate documents.

3.4 The Self-Attention Decoder Component

As we described in Section 3.2, the encoder can also take the subtopics as inputs, and return the encoded representation of the subtopics. This is because the subtopic embeddings we used are actually the document embeddings. We use the subtopic embeddings released by Jiang et al. [14] based on doc2vec. The subtopic embeddings is produced from the pseudo documents of those corresponding subqueries: retrieve top Z documents with traditional IR model (e.g. BM25) first, and then concatenate these documents together to produce the pseudo documents. Those embeddings of the pseudo documents will be used as the embeddings of the subtopics.

The encoded subtopic representations are important to the decoder, since these representations include the attention distributions of the subtopics. In diverse ranking tasks, the available subtopics are mined from the query and they are usually more than the actual user intents. Comparing with the user intents, the subtopics may still contain redundancy and mislead the diversification model. And the encoded subtopic representations include the encoder attention distributions of the subtopics, these distributions can be used to leverage the subtopics' potential redundancy and minimize the misleading.

The decoder structure will take the representation of documents as query matrix, and subtopics as key and value matrix, returning the $\boldsymbol{H}_{\mathbf{dec}}$ representation matrix for the documents with multi-head attention:

$$\boldsymbol{H}_{\mathbf{dec}} = \mathrm{SelfAttnDec}(\boldsymbol{H}_D^{\mathrm{enc}}, \boldsymbol{H}_I^{\mathrm{enc}}) \quad (10)$$

$$\mathrm{SelfAttnDec}(\boldsymbol{H}_D^{\mathrm{enc}}, \boldsymbol{H}_I^{\mathrm{enc}}) = \mathrm{MultiHead}(\boldsymbol{H}_D^{\mathrm{enc}}, \boldsymbol{H}_I^{\mathrm{enc}}, \boldsymbol{H}_I^{\mathrm{enc}}). \quad (11)$$

The output of the decoder $\boldsymbol{h}_t^{\mathrm{dec}}$ will be the subtopic representation of the document d_t, this representation models the subtopics coverage of document d_t. The rest part of the decoder component is just the same as the encoder component, including feed-forward network, ReLU activation function and layer normalization.

3.5 Training and Optimization

In this section we will introduce the training and optimization process of DESA in details. As we described above, DESA will take the document sequence and subtopics as input, and return the ranking scores of all those documents in the given document sequence. In the training phase, the score of a ranking r is calculated by summing up all the scores of documents in r:

$$s_r = \sum_{i=1}^{|r|} s_i. \quad (12)$$

3.5.1 *The list-pairwise sampling.*

Since the dataset of search result diversification task is limited, we inherit the list-pairwise sampling approach from Jiang et al. [14] in order to get enough training samples. We are using pairs of training samples (C, d_1, d_1) with common context C, appending document pair d_1 and d_2 to generate the document sequence pair r_1 and r_2, and the metric(e.g. α-nDCG) of positive ranking $M(r_1)$ should be better than the negative ranking $M(r_2)$.

The sampling process is described as follows: first the contexts C with different lengths are obtained from both ideal rankings and random sampled rankings, then the rest of the candidate documents are traversed, sampling a pair of document (d_1, d_2) when $[C, d_1]$ and $[C, d_2]$ are leading to different metrics.

When using the list-pairwise samples, the original loss function can be defined as a binary classification log-loss formation:

$$Loss = \sum_{q \in Q} \sum_{s \in S_q} |\Delta M| [y_s \log(P(r_1, r_2)) + (1 - y_s) \log(1 - P(r_1, r_2))] \quad (13)$$

where s is a pair of samples and S_q is all the sample pairs of query q, Q is the set of all the queries, $y_s = 1$ for positive and 0 for negative, $P(r_1, r_2) = \sigma(s_{r_1} - s_{r_2})$ for the probability of being positive. $\Delta M = M(r_1) - M(r_2)$ represents the weights of this sample, meaning that if the metric gap between the positive and negative rankings is larger, the sample will be more important.

3.5.2 *The sequence mask for training.*

In the training phase, both the positive and the negative samples are the ground truth rankings, not the candidate document sequence. So the self-attention components are modified with a sequence mask used in the original Transformer decoder structure. Similar as the behavior of the users, the diverse ranking task is a top-down process and the evaluation metrics of the document at position i should not be affected by the document at position $j (j > i)$. The sequence mask will prevent the unexpected self-attention interactions and make sure every document will only interact with itself and those documents at former positions. The scores of documents at former position will not be affected by the documents at latter position. Notice that the sequence mask will only take effect in the training phase.

3.5.3 The context-based pairwise loss function. As we described in Equation (12), the scores of a ranking r is the sum of all the document ranking scores in the sequence. For the sampling pair $r_1 = [C, d_1]$ and $r_2 = [C, d_2]$ we've got $s_{r_1} = \sum_{i \in C_1} s_i + s_{d_1}$ and $s_{r_2} = \sum_{i \in C_2} s_i + s_{d_2}$. Here $C_1 = C_2 = C$.

As we described above, ignoring the effect of dropout layers, the sequence mask in training phase will strictly ensure that $\sum_{i \in C_1} s_i = \sum_{i \in C_2} s_i$. So we've got:

$$\begin{aligned} s_{r_1} - s_{r_2} &= s_{d_1} - s_{d_2}, \\ P(r_1, r_2) &= P(d_1, d_2). \end{aligned} \tag{14}$$

Denoting the binary classification log-loss function as *LogLoss*, Equation (13) can be simplified as:

$$Loss = \sum_{q \in Q} \sum_{[C,(d_1,d_2)] \in S_q} |\Delta M| \mathrm{LogLoss}(\mathrm{P}(\mathrm{d}_1, \mathrm{d}_2)). \tag{15}$$

This is the definition of the context-based pairwise function. For search result diversification task, the scores of d_1 and d_2 depends on the context C, but since the metrics of the context documents should not be affected by the latter documents, the ranking scores of $\sum_{i \in C} s_i$ will not affect the loss function. This means that the context-based pairwise loss function is actually a pairwise loss function for the document pair (d_1, d_2), not a listwise function. The target of the model optimization is to maximize the distance between positive document d_1 and negative document d_2. When the model is being trained, the goal of optimization is to improve the model's ability of indicating if a single document in the candidate sequence is novel and covers more subtopics comparing with the other candidate documents.

3.6 The Ranking Process with Self-Attention

As we described above, research shows that the diversity ranking is NP-hard and for most of the previous models, greedy sequential selection is a common solution [25]. Those models will compare every candidate document with the selected sequence and select the best candidate document one-by-one appending it into the selected sequence. For our self-attention based framework, when the ranking process starts, the model will take all the non-diversified candidate document ranking as an initial input, and jointly return the diversified ranking scores of all those documents. Similar as some other self-attention based ad-hoc ranking approach e.g. SetRank [20], the model can return the ranking list with sorting all the candidate documents with their ranking scores. Different from the greedy document sequential selection models, DESA doesn't depend on the selected document sequnece.

With globally measuring all the candidate documents, DESA will outperform the previous models especially at former ranking positions. We will take the example in Section 1 to explain. Assuming there are three candidate documents d_1, d_2, d_3, with d_1 covering the subtopic q_1, d_2 covering q_2, q_3 and d_3 covering q_1, and the three documents has got similar relevance scores to the given query. Comparing with the greedy document selection models, DESA will return a higher ranking scores for d_2. since d_2 is novel and covers more subtopics comparing with d_1 and d_3. Then the d_2 will be put on a former ranking position. This process indicates the advantage of DESA, because the goal of search result diversification is to satisfy more user intents at former ranking positions, and promoting the documents covering more subtopics to the former ranking positions will be more suitable to improve the user experience.

3.7 Theoretical Analysis

In this section we will analyze the effect of self-attention in the encoder-decoder structure of DESA. Here we will describe why self-attention is suitable in the diverse ranking task in details. For simplicity, we will first focus on a single-layer self-attention function in the encoder component and ignore those assist strategies e.g. positional embedding, multi-head attention or layer normalization.

The self-attention interaction of the document sequence D is calculated in parallel as a whole matrix, and the attention score can be written as the following equation focusing on the t-th document d_t represented as q_t, discarding the scalar factor $\sqrt{d}$:

$$\mathrm{Score}_{\mathrm{Attn}}(\boldsymbol{q_t}, \boldsymbol{K}) = \mathrm{Softmax}(\boldsymbol{q_t} \boldsymbol{K}^T). \tag{16}$$

As we described above, for self-attention, it can be approximated that $\boldsymbol{q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{D}$, and $\boldsymbol{q_t} \approx \boldsymbol{d_t}$. The $\boldsymbol{q_t} \boldsymbol{K}^T$ in Equation (16) can be seen as the dot product scores between the t-th document and each document in the sequence including itself. With the softmax function, those scores will be converted into weights. Since the dot product of two documents can represent the similarity score between the two documents, those weights model the similarity distribution between d_t and every document in the sequence. The self-attention output of d_t is defined as follow:

$$\begin{aligned} h_t &= \mathrm{Softmax}([s_1, \ldots, s_n])^T \boldsymbol{V} \\ &= [w_1, \ldots, w_n] \boldsymbol{V} \\ &= W_t^T \boldsymbol{V}. \end{aligned}$$

Here n is the length of document sequence, s_i is the dot product between document d_t and d_i, and w_i is the similarity weight converted from s_i. Section 3.5.1 shows the details of list-pairwise sampling for training. With shared selected context document sequence C , positive and negative document pair $d_{\mathrm{pos}}, d_{\mathrm{neg}}$, the positive and negative samples can be written as $[C, d_{\mathrm{p}}]$ and $[C, d_{\mathrm{n}}]$.

Due to the property of softmax function, $\sum_{i=1}^{n} w_i = 1$, for the weights distribution of document d_t, the equation can be written as:

$$\sum_{i \in C} w_i + w_t = 1.$$

In the view of MMR, comparing with the context C, the positive document d_{pos} should be a novel document, which means that d_{pos} should be dissimilar with the documents in the context C. The dot product scores of d_{pos} with other documents $d_i (i \in C)$ should be significantly smaller than the scores of d_{pos} with itself, indicating $s_{\mathrm{pos}} >> \sum_{i \in C} s_i (i \in C)$. After the softmax function, it has got $w_{\mathrm{pos}} >> \sum_{i \in C} w_i (i \in C)$. For the negative document d_{neg}, since it's a redundant document, the dot product scores with the context documents will be close to the score with itself, and $w_{\mathrm{neg}} >> \sum_{i \in C} w_i (i \in C)$ is no longer valid.

As a result, a positive document will gain an attention distribution concentrated to the document itself, while a negative document will gain an average distribution. This is identical to the spirit of MMR, since a novel document should be dissimilar with the other documents, and its similarity scores with other documents should be much smaller than the score with itself.

With the context-based pairwise optimization, the attention distribution distance gap the positive and negative documents will get bigger, and the learning-to-rank function of the model will be trained to return a ranking score of d_{pos} higher than d_{neg}. With more self-attention layers, the distribution distance between the positive and negative samples will be expanded and the learning-to-rank function will be more effective to judge the novelty of a candidate document.

This analysis mainly depends on the self-attention encoder, and the principle of the decoder component is similar as the encoder:

$$\begin{aligned}\boldsymbol{h}_t^{\text{dec}} &= \text{Attn}(\boldsymbol{h}_t^{\text{enc}}, \boldsymbol{H}_I^{\text{enc}}, \boldsymbol{H}_I^{\text{enc}}) \\ &= [w_1^{\text{dec}}, \ldots, w_n^{\text{dec}}]\boldsymbol{H}_I^{\text{enc}} \\ &= (W_t^{\text{dec}})^T \boldsymbol{H}_I^{\text{enc}}.\end{aligned}$$

Here the w_i^{dec} is the attention weights between document d_t and subtopic q_i. Similar as the encoder attention distribution, the decoder attention distribution of $\boldsymbol{h}_t^{\text{dec}}$ will be focusing on the subtopics relevant to d_t, and the irrelevant subtopics will be ignored with lower attention weights. The decoder attention distributions of positive and negative documents will be similar as the encoder attention. For the positive document the attention distribution will be concentrated to the relevant subtopics, and for the negative document, the distribution will be average since none of the subtopics are relevant to the document.

The decoder takes the encoded output representations of documents $\boldsymbol{H}_{\text{D}}^{enc}$ and subtopics $\boldsymbol{H}_{\text{I}}^{enc}$ as input, and a redundant document will also be affected by its encoder attention distribution, letting its decoder attention distribution more average than the decoder attention distribution for its original representation. This effect is also valid for subtopics. With a subtopic q_i with redundant encoder attention, since $\boldsymbol{h}_{q_i}^{\text{enc}} = (\boldsymbol{W}_{q_i}^{enc})^T \boldsymbol{I}$, its corresponding decoder attention weights w_i^{dec} will also be affected and weakened through the average distribution of $\boldsymbol{W}_{q_i}^{\text{enc}}$.

Taking $\boldsymbol{h}_t^{\text{dec}}$ as input, the learning-to-rank function will be able to model the subtopic coverage of $\boldsymbol{d}_t$ together with the relevance scores of subtopics.

3.8 Discussion

DESA is inspired by several existing models in IR based on self-attention e.g. SetRank [20]. And the implicit implementation of DESA with no subtopics can be seen as an adaption of SetRank for search result diversification task. While the properties of diverse ranking task is significantly different from ad-hoc ranking task, and the training dataset of diverse ranking task is very limited. Comparing with SetRank, DESA has got the following differences:

(1) SetRank is not designed for search result diversification task, it's Transformer encoder structure will be unable to take the subtopics into consideration. While DESA is using a full encoder-decoder structure, and it can leverage both the document novelty and subtopic coverage.

(2) DESA takes the preliminary representations of the document sequence as input, instead of the relevance features used in SetRank. The self-attention networks in DESA only focus on learning the representations of the documents to indicate if a document is novel and covers more subtopics, and the framework deals with the relevance features separately.

(3) SetRank is using the attention rank loss function as a listwise function, focusing on measuring the attention distribution of the whole ranking list. And DESA is using context-based pairwise function as a pairwise function, the attention distributions stand for the similarity distribution and subtopic satisfaction distribution of every single document.

4 EXPERIMENTAL SETTINGS

4.1 Data Collections and Evaluation Metrics

4.1.1 The data collections. In the experiments we are using the same dataset as many previous diversification models(e.g.HxQuAD, PAMM-NTN, DSSA) which includes the Web Track dataset from TREC 2009 to 2012. There are in total 200 queries and 198 queries are used since query #95 and #100 have got no diversity judgements to use. Each of them includes 3 to 8 annotated subtopics, and the relevance rating is marked as relevant or irrelevant at subtopic level. We conduct all the experiments on the ClueWeb09 dataset [26].

The subtopics used by the model come from the Google query suggestions provided by Hu et al.[1], and we only use the first level of the subtopics with no hierarchical subtopics. The max subtopic number of the queries is 10, and the average subtopic number is about 9.48. As those previous works do [9] we treat all those subtopics with uniform weights.

For a fair comparison, we are using the document relevance features and embeddings exactly the same as the DSSA, which have been released by Jiang et al. [14] in the repository on GitHub[2]. Those training data includes 18 relevance features for each query and subquery produced by traditional IR models e.g. BM25 and TF-IDF, and the document embeddings are generated by doc2vec with window size 5. In the future work we will try to import several deep-learning based technologies for feature extraction and document representation e.g. K-NRM [27] or BERT [18].

4.1.2 The evaluation metrics. The official diversity evaluation metrics of Web Track include ERR-IA [28], α-nDCG [29] and NRBP [30], which are used in our experiments. Besides the metrics above, we also include the metrics of Precision-IA [6] (denoted as Pre-IA) and Subtopic Recall [31] (denoted as S-rec). Inheriting the spirit of the previous works [11–14], all those metrics are computed on top 20 results of a document ranking list. Two-tailed paired t-test are used to conduct significance testing with p-value<0.05. In the significance testing, DESA is compared with the DSSA as the SOTA explicit supervised model.

4.2 Model Settings

On our GPU machine, the training phase of DESA with the training samples of 160 queries can be finished in 3 hours. We tune the layer number L of the self-attention network in order to avoid overfitting, here $L = L_{\text{enc}} + L_{\text{dec}}$, L_{enc} is the layer number of encoder component and L_{dec} is the number of decoder. We compare DESA with the undiversified baseline and those previous implicit/explicit

[1] http://www.playbigdata.com/dou/hdiv

[2] https://github.com/jzbjyb/dssa

supervised models, the detail settings of DESA will be described below. We use 5-fold cross validation to turn the parameters in all experiments with the widely used metrics α-nDCG@20.

4.2.1 Baseline models. The settings of the baseline models are described as follows:

Lemur. We use the search results produced by language model and retrieved by the Lemur service[3] as the non-diversified baseline. These results are released by Hu et at. [9] and can be found on the website[4].

xQuAD [7], PM2 [8], HxQuAD and HPM2 [9]. These are the unsupervised explicit baseline approaches for comparison. All the unsupervised methods use the parameter λ to combine the relevance and diversity linearly. HxQuAD and HPM2 requires extra parameter α to control the weights of the hierarchical subtopic layers. The parameters are tuned with cross validation and ListMLE [10] is used to learn a prior relevance function with no diversification.

R-LTR [11], PAMM [12] and PAMM-NTN [13]. Inspired by previous work [14], we use the metric of $\alpha - n$DCG@20 to tune the parameters. The neural tensor network(NTN) is used with both R-LTR and PAMM, denoted as R-LTR-NTN and PAMM-NTN. The number of tensor slices for NTN is tuned from 1 to 10, and the number of positive ranking τ^{+} and negative ranking τ^{-} are tuned per query for the PAMM. The distributed representations of documents here are 100-dimensional vectors generated by the LDA [32].

DSSA [14]. We train the DSSA model with the code and data released by Jiang et al. on GitHub[5], and use the following optimized settings described in the work of DSSA: LSTM cells, max-pooling on subtopic attention, hidden size 50, doc2vec embedding dimension 100 and random permutation count 10 for the list-pairwise samples. We do not use the embedding of LDA reported in the work, instead we use the doc2vec embedding released for a fair comparison. The result is denoted as DSSA (doc2vec).

Since the deep reinforced learning based models e.g. MDP-DIV [14] and M2DIV [15] are taking too much time to train, we do not take those models as baseline.

5 EXPERIMENTAL RESULTS

5.1 Overall Results

Table 2 shows the results overall of all the models above. DESA outperform all the baselines include the state-of-the-art implicit and explicit approaches. The performance improvement is statistically significant on all the metrics except the Pre-IA. These experimental results shows the advantage of DESA clearly. Comparing the state-of-the-art supervised approach, DESA's improvement over DSSA on $alpha-$nDCG is about 3%. As an explicit model, DSSA use the RNN and attention mechanism to select the best document satisfying the subtopics needed by the selected sequence. Since the RNN can't measure the interactions between each document directly, DESA outperforms DSSA by leveraging both document novelty and subtopic coverage simultaneously. And as a greedy sequential selection model, DSSA may select the local optimal document at each step, leading to a global suboptimal ranking. While DESA can learn the interactions between all the candidate documents and subtopics globally, significantly minimizing the gap between local and global optimal rankings.

Table 2: Performance comparison for all the approaches. Best Results are in bold. ★ indicates that the model significantly outperforms all baselines.

Methods	ERR-IA	α-nDCG	NRBP	Pre-IA	S-rec
Lemur	.271	.369	.232	.153	.621
xQuAD	.317	.413	.284	.161	.622
PM2	.306	.411	.267	.169	.643
HxQuAD	.326	.421	.294	.158	.629
HPM2	.317	.420	.279	.172	.645
R-LTR	.303	.403	.267	.164	.631
PAMM	.309	.411	.271	.168	.643
R-LTR-NTN	.312	.415	.272	.166	.644
PAMM-NTN	.311	.417	.272	.170	.648
DSSA (doc2vec)	.350	.452	.318	.184	.645
DESA	**.363**★	**.464**★	**.332**★	.184	**.653**★

5.2 Effects of Hyperparameter Settings

We produce several experiments in order to investigate the effects of different settings to the performance of DESA. Since DESA mainly depends on the effect of self-attention, we focus on the self-attention component and deploy different experiments to test the different settings of the self-attention encoder. The baseline settings of the self-attention component include the following items: the initial document/subtopic embedding in 100 dimensions projected into 256 dimensions as the input of the self-attention network, the $d_{\mathrm{FF}} = 400$ in the feed-forward network and the head number $H = 8$ for multi-head attention.

We test the effect of different encoder layer numbers L_{enc} from 1 to 3 with decoder layer numbers L_{dec}=1, and the effect of L_{dec}=1 and L_{dec}=2 with encoder layer number L_{enc}=1 or L_{enc}=2.

The effects of different settings are shown in Table 3. As we can see, different settings of the self-attention component may slightly influence the effect of the whole DESA framework. In our experiment, we find that the total number of self-attention layer L should be strictly limited in order to prevent over-fitting and ensure the performance. In the diverse ranking task, L_{enc}=2,L_{dec}=1 will lead to the best performance. Since the dataset is limited, more self-attention layers will cause more computational cost and may lead to overfitting.

5.3 Effects of Subtopic Settings

In DESA, the decoder component takes the subtopics as the key and value matrices and returns the decoded document representations indicating the coverage of the subtopics. Here we conduct several experiments to check the effect of different decoder settings.

The encoder-only framework can be seen as a simple adaption of SetRank to the diverse ranking task, so we proposed two settings

[3] Lemur service: http://boston.lti.cs.cmu.edu/Services/clueweb09_batch/

[4] http://www.playbigdata.com/dou/hdiv

[5] https://github.com/jzbjyb/dssa

Table 3: Effects of Different Settings

Settings	ERR-IA	α-nDCG	NRBP	Pre-IA	S-rec
L_{enc}=1, L_{dec}=1	.357	.457	.324	.183	.650
L_{enc}=2, L_{dec}=1	**.364**	**.464**	**.332**	**.184**	**.653**
L_{enc}=3, L_{dec}=1	.355	.455	.323	.182	.654
L_{enc}=1, L_{dec}=2	.361	.462	.329	.182	.658
L_{enc}=2, L_{dec}=2	.358	.460	.324	.180	.658
No Subtopics	.344	.445	.311	.177	.648
Relevance Scores	.357	.458	.326	.183	.653
Encoded Subtopics	**.364**	**.464**	**.332**	**.184**	**.653**
Original Subtopics	.349	.453	.313	.180	.655

in order to show the effect of the decoder components. The two settings are denoted as:"No Subtopics" and "Relevance Scores". The "No Subtopics" indicates the framework with encoder and query relevance features only, none of the subtopic information is used in ranking task. This framework can be seen as a simple adaption of SetRank to the diverse ranking task. Another expansion of the SetRank adaption is denoted as "Relevance Scores", where the relevance scores of the subtopics $s_{q_i} (i \in [1, k])$ are included, and neither decoder component nor subtopic embeddings are used in the framework. These two adaptions use the optimized settings of $H = 8$ and $L = L_{\text{enc}} = 3$.

We also propose two settings to check the effect of the encoded subtopic representations in the decoder component. The "Encoded Subtopics" denotes using the encoded subtopic representations $\boldsymbol{H}_I^{\text{enc}}$ produced by the encoder component, and "Original Subtopics" denotes using the original subtopic embeddings after projection. Their hyperparameter settings are $H = 8$ and $L_{\text{enc}} = 2, L_{\text{dec}} = 1$.

The experiment result shows that the full encoder-decoder structure of DESA outperforms the simple adaption of SetRank. The structure of SetRank is not designed for search result diversification task, and it can't take fully use of the subtopic embeddings. The results have prove the effectiveness of the full encoder-decoder structure specialized for explicit search resuld diversification. Comparing with the simple adaption of SetRank, the decoder component in DESA can measure the attention distribution of every document for the satisfaction of subtopics, leveraging both novelty and subtopic coverage together.

Besides the SetRank adaptions, the outperforming result of "Encoded Subtopics" proves that the encoded representation of the subtopics can be used to reduce the latent redundancy of the subtopics. As we described in Section 4.1.1, the average subtopic number is 9.48 among all the queries, however, the actual user intent numbers are only 3 to 8, which are smaller than the subtopic numbers in diverse ranking task. This result indicates that the subtopics used in ranking process may still contain redundancy and mislead the diversification model.

We can take the query #1 "obama family tree" in the TREC WebTrack dataset as an example. There are 10 subtopics based on Google query suggestions, while there are only 3 actual user intents in the TREC official subtopic annotations. And there are two subtopics q_1 for "obama family tree pictures" and q_2 for "obama family tree photos" in the query suggestions. While both q_1 and q_2 are corresponding to the same user intent, the redundant subtopics may mislead the model to select a document which covers the "different" subtopics and increase the actual redundancy.

Comparing with original subtopic embeddings, the encoded subtopic representations can integrate the subtopics' encoder attention distribution into the decoder attention. Similar as the document's attention distribution, the subtopic's attention distribution can also indicate the redundancy of a subtopic, and the decoder attention of the redundant subtopic will also be affected. As a result, the decoder attention will be adjusted to reduce the negative effect of the latent subtopic redundancy to the diverse ranking task.

Table 4: Metrics improvement per ranking position. "Total Imp." denotes the total improvement of DESA on all the 200 queries, "Avg Imp." denotes the average improvement per position.

Model	ERR-IA			α-nDCG		
	@5	@10	@20	@5	@10	@20
DSSA	.328	.344	.351	400	.428	.452
DESA	.343	.356	.363	.417	.439	.464
Total Imp.	2.98	2.51	2.64	3.50	2.06	2.37
Avg Imp.	.597	.252	.132	.701	.206	.118

5.4 Analysis of Former Ranking Position

As we described in Section 3.6, theoretically our proposed DESA framework will perform better than the greedy document sequential selection based model at former ranking positions. Here we analyze the effect of DESA at different ranking positions. We compare DESA with DSSA, the state-of-the-art greedy document sequential selection based model. In this experiment we use the ERR-IA and α-nDCG metrics computed on top 5, top 10 and top 20 results of a document ranking list to analyse the effect of DESA at former ranking positions. Results are shown in Table 4

We calculate the total metric improvement of DESA comparing with DSSA. For simplicity of calculation we use the metrics sum of all the 200 queries instead of the mean metrics, denoted as Total Imp. And we calculate the average metrics improvement per position to measure the improvement of DESA at different ranking position ranges, this value is denoted as Avg Imp. For example, the ERR-IA@5 of DESA and DSSA (doc2vec) is 0.343 and 0.328, the total improvement of ERR-IA@5 is calculated as $(0.343 - 0.328) \times 200 \approx 2.98$, and the average improvement of ERR-IA@5 is $2.98 \div 5 \approx 0.597$.

From this table it can be seen that the average metrics improvement numbers per position of short ranking lists are larger than the numbers of longer ranking lists. The experimental results are identical to the analysis in Section 3.6, indicating that in the early stage of ranking, the sequential selection based models may fail to select the globally best candidate document after a short or empty selected sequence. DESA can measure all the candidate documents globally and promote the diversified documents at former positions, satisfying the user intents earlier comparing with the greedy sequential selection based models.

6 CONCLUSIONS

In this paper, we propose a self-attention based supervised diversification framework leveraging both document novelty and subtopic coverage. Instead of greedy document sequential selection, this framework can model all the candidate documents globally and sort those documents jointly with their ranking scores to generate the ranking list. Comparing with the previous works of search result diversification, this is the first time to model all the candidate documents simultaneously and select the best candidate document globally without greedy sequential selection. The experimental results show that modeling the candidate document interaction between each other can significantly minimize the gap between the local and global optimal rankings. In this work our model is focusing on the candidate document sequence, ignoring the selected document sequence. Simultaneously modeling every document's interaction with both the other candidate documents and the selected document sequence may be a potential future work.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and No. 61832017, and Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098.

REFERENCES

[1] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, September 1999.

[2] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 581–590, New York, NY, USA, 2007. Association for Computing Machinery.

[3] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: A study and analysis of user queries on the web. *Inf. Process. Manage.*, 36(2):207–227, January 2000.

[4] Ruihua Song, Zhenxiao Luo, Ji-Rong Wen, Yong Yu, and Hsiao-Wuen Hon. Identifying ambiguous queries in web search. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 1169–1170, New York, NY, USA, 2007. Association for Computing Machinery.

[5] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. Association for Computing Machinery.

[6] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, New York, NY, USA, 2009. Association for Computing Machinery.

[7] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 881–890, New York, NY, USA, 2010. Association for Computing Machinery.

[8] Van Dang and W. Bruce Croft. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 65–74, New York, NY, USA, 2012. Association for Computing Machinery.

[9] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 63–72, New York, NY, USA, 2015. Association for Computing Machinery.

[10] Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural svms. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1224–1231, New York, NY, USA, 2008. Association for Computing Machinery.

[11] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. Learning for search result diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research I& Development in Information Retrieval*, SIGIR '14, pages 293–302, New York, NY, USA, 2014. Association for Computing Machinery.

[12] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 113–122, New York, NY, USA, 2015. Association for Computing Machinery.

[13] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Modeling document novelty with neural tensor network for search result diversification. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 395–404, New York, NY, USA, 2016. Association for Computing Machinery.

[14] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. Learning to diversify search results via subtopic attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 545–554, New York, NY, USA, 2017. Association for Computing Machinery.

[15] Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, page 125–134, New York, NY, USA, 2018. Association for Computing Machinery.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[17] Alec Radford. Improving language understanding by generative pre-training. 2018.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[19] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: enhanced language representation with informative entities. *CoRR*, abs/1905.07129, 2019.

[20] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xue qi Cheng, and Ji-Rong Wen. Setrank: Learning a permutation-invariant ranking model for information retrieval. *ArXiv*, abs/1912.05891, 2019.

[21] Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. Self-attentive document interaction networks for permutation equivariant ranking, 2019.

[22] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2333–2338, New York, NY, USA, 2013. Association for Computing Machinery.

[23] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.

[24] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[25] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. *Found. Trends Inf. Retr.*, 9(1):1–90, March 2015.

[26] Charles L Clarke, Nick Craswell, and Ian Soboroff. Overview of the trec 2009 web track. Technical report, DTIC Document, 2009.

[27] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. *CoRR*, abs/1706.06613, 2017.

[28] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621–630, New York, NY, USA, 2009. Association for Computing Machinery.

[29] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 659–666, New York, NY, USA, 2008. Association for Computing Machinery.

[30] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In Wolfgang Lindner, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena I. Vakali, editors, *Current Trends in Database Technology - EDBT 2004 Workshops*, pages 588–596, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[31] ChengXiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. *SIGIR Forum*, 49(1):2–9, June 2015.

[32] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, March 2003.