Modeling Intent Graph for Search Result Diversification

Zhan Su[2], Zhicheng Dou[1*], Yutao Zhu[3], Xubo Qin[2], and Ji-Rong Wen[4,5]
[1] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2] School of Information, Renmin University of China, Beijing, China
[3] Université de Montréal, Montréal, Québec, Canada
[4] Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China
[5] Key Laboratory of Data Engineering and Knowledge Engineering, MOE, China
{suzhan,*dou}@ruc.edu.cn

ABSTRACT

Search result diversification aims to offer diverse documents that cover as many intents as possible. Most existing implicit diversification approaches model diversity through the similarity of document representation, which is indirect and unnatural. To handle the diversity more precisely, we measure the similarity of documents by their similarity of the intent coverage. Specifically, we build a classifier to judge whether two different documents contain the same intent based on the document's content. Then we construct an intent graph to present the complicated relationship of documents and the query. On the intent graph, documents are connected if they are similar, while the query and the document are gradually connected based on the document selection result. Then we employ graph convolutional networks (GCNs) to update the representation of the query and each document by aggregating its neighbors. By this means, we can obtain the context-aware query representation and the intent-aware document representations through the dynamic intent graph during the document selection process. Furthermore, we fuse these representations and intent graph features to diversity features. Combined with the traditional relevance features, we obtain the final ranking score that balances the relevance and the diversity. Experimental results show that this implicit diversification model significantly outperforms all existing implicit diversification methods, and it can even beat the state-of-the-art explicit models.

CCS CONCEPTS

• **Information systems → Information retrieval diversity**.

KEYWORDS

Intent Graph, Search Result Diversification, Graph Neural Network

ACM Reference Format:
Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. 2021. Modeling Intent Graph for Search Result Diversification. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3404835.3462872

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462872

1 INTRODUCTION

Search result diversification can efficiently alleviate the influence brought by the ambiguous query. In addition to improving the hit rate of different users' information needs, diversification can also meet the user's intrinsic diversity need. For example, by issuing a query "cook noodles", a user may look for different recipes for cooking noodles. To fulfill such kinds of needs, diversification approaches are expected to display documents with various subtopics while considering the relevance of the documents.

Existing approaches to search result diversification can be roughly categorized into explicit methods and implicit methods. Since the diversity ranking task is NP-hard, most approaches adopt greedy selection strategies as a compromise [31], *i.e.*, iteratively selecting the most diverse document from the candidate set at each step. Explicit approaches explicitly leverage subtopics distribution as the substitution of real intents to measure diversity of each document [9, 30], while implicit approaches focus on the document's novelty based on the similarity between documents and do not rely on subtopics [3]. As subtopic mining itself is a very challenging task, the implicit result diversification methods have received much attention in real search services [12, 32, 37, 38, 41, 46].

Although many implicit methods [37, 38, 46] have been proposed in recent years, most existing methods measure the document's novelty based on the dissimilarity between the candidate documents and the selected documents. For example, the typical implicit approach NTN [38] automatically learns a novelty function based on the preliminary representation (*e.g.*, doc2vec or PLSA) of the candidate document and the selected documents. There are two main drawbacks of these methods: (1) With the only diversity ranking loss, it is often difficult to tell if a wrong ranking stems from a wrong combination of features or the incompetent document novelty features, thus the document's novelty cannot always be learned to its best. Meanwhile, merely computing the document's novelty based on the preliminary representation is also inaccurate, since the document's content is an essential source for deriving the document's diversity information. (2) The novelty of a candidate document is measured by its dissimilarity with the selected documents. The intent coverage of selected documents on the query and the similarity among candidate documents are neglected. Under this circumstance, it is difficult to select an optimal document from the candidates to satisfy the user's intent.

From our perspective: (1) Diversity ranking models try to offer documents covering as many intents as possible. Therefore, the real intents of the documents are essential parts when considering the similarity of documents. For example, two documents can be

treated as more similar if they share more subtopics. Indeed, the real user intent is often hidden in the document content. So, our first challenge is how to take both the documents' content and their intent coverage into account for computing their similarity. (2) The information needs of the query may be partly satisfied according to the subtopics contained in the selected documents. Hence, diversity ranking models should capture the dynamic diversity needs of the query timely. Besides, the novelty of all the candidate documents is not independent. When a candidate document is selected, the novelty of the remnant documents will be affected. Therefore, our second challenge is how to consider the complicated and dynamic relation of the query and documents during the document selection process.

To tackle the challenges above, we propose to model the document's similarity through the similarity of the document's intent coverage directly rather than the similarity of document representations. However, due to the subtopic mining is a very challenging task in explicit methods [9, 14, 16, 31], we dedicate to implicit methods in this paper. In fact, we can derive two document's intent similarity from the document's content without exactly knowing what the subtopics are. To fully leverage the abundant information of the document's content, we design a document relation classifier to judge whether two documents cover the same intent based on the content. Additionally, to further enhance the weak relation of documents extracted from the classifier and model the similarity of documents with a global view, we present the complicated relationship of documents and the query on the graph. Specifically, we build an intent graph where two documents are connected if they share the same intent. The selected document is also connected to the query so as to distinguish it from remnant candidate documents. Moreover, the intent graph could be dynamically adjusted according to the selected documents for a better representation of the query's information needs. With the help of the graph structure, we can derive the local diversity features when we focus on the document nodes and their neighbors, while the global features are also easy to obtain by aggregating features of the entire intent graph. Motivated by the powerful aggregating capability of the graph convolutional networks [18] (GCN), we adapt GCN to this dynamic intent graph for learning the intent-aware document representations and the context-aware query representation. Combining the features extracted from the intent graph, we can model the document's diversity in a direct and precise way. To our best knowledge, we are the first to leverage the **Graph** to represent the relationship between the query and documents **for search result diversification**. Therefore, our method is named Graph4DIV. The experimental results show that our approach outperforms the state-of-the-art implicit method by 12.2% in terms of α-nDCG@20.

The main contributions of this paper are summarized as follows:

(1) We propose to model documents' similarity directly through their intent coverage. This brand new idea offers a better way for capturing the essence of the document's novelty and result diversity without the explicit use of subtopics.

(2) We use a dynamic intent graph to model the complicated query-document and document-document relationships simultaneously and timely. We leverage GCN to learn better representations of the query and documents from the intent graph. As far as we know, this is the first method of adapting GCN to search result diversification.

Table 1: Categorization of diversification approaches.

	Unsupervised	Supervised
Explicit	IA-Select, HxQuAD, PM2, TPM2, TxQuAD, xQuAD, HPM2	DSSA, DESA, DVGAN
Implicit	MMR	SVM-DIV, R-LTR, PAMM, NTN, **Graph4DIV** (our approach)

(3) Our implicit diversification approach largely improves the state-of-the-art performance of implicit methods and outperforms all explicit diversity approaches, which makes a huge advance for diversification approaches to be applied in real scenarios.

The rest of the paper is organized as follows. We review some related work in Section 2. Then we introduce the Graph4DIV framework in Section 3. Section 4 presents the experimental settings and results. In Section 5, we analyse different settings and influences of the experiment. Finally, we conclude the paper in Section 6.

2 RELATED WORK

2.1 Search Result Diversification

Search result diversification can be categorized into explicit approaches and implicit approaches depending on whether they use subtopics or not. From another perspective, diversification methods include heuristic (unsupervised) methods and supervised methods as shown in Table 1. In this section, we will briefly introduce the major diversification approaches in terms of the features they use.

Implicit Diversification Approaches Most implicit methods obey the framework of MMR [3], which balances the relevance and novelty of the document with a parameter λ. The novelty is mainly measured by dissimilarity between retrieved documents. It provides a balanced strategy for ranking the documents returned by search engines, which becomes the foundation of many implicit and explicit approaches [10, 31, 44]. Yue and Joachims [42] proposed SVM-DIV that uses structural SVM to measure the diversity of the documents. R-LTR [46] was a relational learning-to-rank approach based on the relation of documents. To solve the problem that loss functions loosely related to the evaluation measures, Xia et al. [37] proposed the PAMM approach to directly optimize diversity evaluation measures. Then neural tensor network (NTN) [38] was introduced to automatically learn the relation functions of the documents. As an implicit approach, our model also follows the framework of MMR. Different from the previous implicit methods, we obtain the diversity features automatically learned from the graph structure that contains the intent information.

Explicit Diversification Approaches Instead of the similarity between documents, most explicit models leverage subtopic coverage to measure documents' diversity. The representative traditional explicit approaches are xQuAD [31] and PM2 [9]. Many further studies are carried out based on them, such as HxQuAD, HPM2 [14], TxQuAD, and TPM2 [10]. To avoid handcrafted functions and parameters, several supervised approaches have been proposed recently. For example, DSSA [16] proposed a list-pairwise loss for training the diversity ranking model. Besides, DSSA also introduced recurrent

neural networks (RNNs) and the attention mechanism to model the subtopic coverage of the document sequence.

Ensemble Diversification Approaches Recently, researchers also consider using explicit (subtopic) features and implicit features together in the ranking process, which could be categorized into explicit methods. For instance, DVGAN [20] combined ranking signals learned by the generator and the discriminator in order to obtain a better ranking model. DESA [28] leveraged both document novelty and subtopic coverage based on self-attention. Compared to these models, our method also leverages the strength of supervised learning but without depending on extra subtopics, and thus it is an implicit method.

2.2 Graph in IR

The graph structure is a very common and natural way to present the relationship of documents, queries, and intents in IR literature. For example, PageRank [27] turns out to be a powerful and typical algorithm to measure the importance of the web pages based on the graph structure. Jiang et al. [15] learned the relevance of query and documents through the Web-scale Click Graph that presents user behavior, which demonstrates that the abundant information contained in the graph helps to improve the search result and the intent is also suitable to present on the graph.

Graph neural networks (GNNs) can efficiently leverage the structural information extracted from the graph. Owing to its aggregation and representation capability, it quickly becomes a powerful tool in many fields, such as computer vision [11], social network analysis [18, 34] and natural language processing [22, 26, 33, 43, 47].

Recently, graph-based learning methods make breakthroughs in the IR literature. Researchers leverage graph structure to enhance the representation of documents and queries. For example, Li et al. [19] learned text representation with graph structure that contains click behavior information. Zhang et al. [45] used graph embedding techniques to learn the representation of query and items in the product search.

Graph convolutional networks (GCNs) [18] can collect the neighbors' information by generalizing traditional convolutional operation from nodes with a fixed degree to ones with a scalable degree. The representations of the nodes on the graph will be enhanced by their neighbors after the GCN. Since implicit search result diversification approaches model document's novelty based on the dissimilarity of documents, we believe GCN could be a suitable tool to refine the document representation with its similar documents.

3 PROPOSED METHOD: GRAPH4DIV

Diversity ranking aims to offer diverse documents that cover as many intents as possible, while most existing implicit methods measure diversity by the dissimilarity of the document representations indirectly and roughly.

In this paper, we hope to directly model the diversity according to the intent contained in the documents. However, it is still a challenging task to mine the precise intent or subtopics from the documents or other data sources. Instead of using explicit subtopics, we propose to implicitly leverage the hidden query intents covered by the top results of the query. We build a classifier to judge whether two documents share the same intent and present the relation on the graph. With the graph structure, our approach can model the complicated relation of multiple documents based on these hidden intents simultaneously and extract both global and local features via GCN.

Table 2: Notations in Graph4DIV

Notation	Definition
$\mathcal{Q}, q$	the query set, the query in the set, $q \in \mathcal{Q}$
$\mathcal{D}$	documents set for the query q, $\lvert\mathcal{D}\rvert = n$
$\mathcal{S}$	selected document sequence for the query q
$\mathcal{C}$	remaining documents for query q, $\mathcal{C} = \mathcal{D} \setminus \mathcal{S}$
$\mathcal{G}_{D,S}$	the intent graph after $\mathcal{S}$ is selected from $\mathcal{D}$
N	the nodes set of the intent graph, $\lvert N\rvert = n + 1$
E	the edges set of the intent graph
v_q	the node of the query q in the intent graph
v_i	the node of the document d_i in the intent graph
$\mathcal{R}$	ranking sequence of the query q
$\mathbf{R}_i$	the relevance feature of i-th document d_i
$\mathbf{H}_i$	the diversity feature of i-th document d_i

3.1 Problem Formulation

The notations in this paper and their descriptions are shown in Table 2. Supposing q is the current query and $\mathcal{D}$ is a list of n candidate documents for q, the task of search result diversification is to generate a new ranked document list $\mathcal{R}$ based on the initial ad-hoc ranking list $\mathcal{D}$, where diverse documents are ranked higher in $\mathcal{R}$ and redundant ones are ranked lower.

Different from the ad-hoc retrieval task, which aims at returning relevant documents, search result diversification needs to consider both (1) the relevance between the query and the document; and (2) the similarity among the documents. As introduced in Section 1, most existing diversification methods apply the greedy selection strategy [37, 46], *i.e.*, iteratively selecting the next document by measuring its relevance with the current query and its novelty compared with the documents that have already been selected in the early iterations.

3.2 Overview of Graph4DIV

The overall structure of our proposed Graph4DIV is shown in Figure 1. Formally, at the time step t, supposing $\mathcal{S}$ is the set of documents already been selected, Graph4DIV determines the next document d^* by measuring the ranking score $f(d_i)$ of each remained candidate document d_i and selecting the document with the highest ranking score. $f(d_i)$ is comprised of relevance and novelty of the document given the current query q, document set $\mathcal{D}$, and selected document sequence $\mathcal{S}$:

$$f(d_i, \mathcal{D}, \mathcal{S}) = \lambda f^{\text{rel}}(d_i) + (1 - \lambda) f^{\text{div}}(d_i, \mathcal{D}, \mathcal{S}), \tag{1}$$

where $f(d_i, \mathcal{D}, \mathcal{S})$ denotes document d_i's ranking score that consists of relevance score $f^{\text{rel}}(d_i)$ and diversity score $f^{\text{div}}(d_i, \mathcal{D}, \mathcal{S})$[1]. λ is the parameter to control the balance between relevance and diversity. This is the common format of most search result diversification models. As for the relevance part, Graph4DIV uses the same relevance features $\mathbf{R}_i$ as those used in previous work [16, 20, 46].

[1]To reduce the notation redundancy, we omit the query q in all equations.

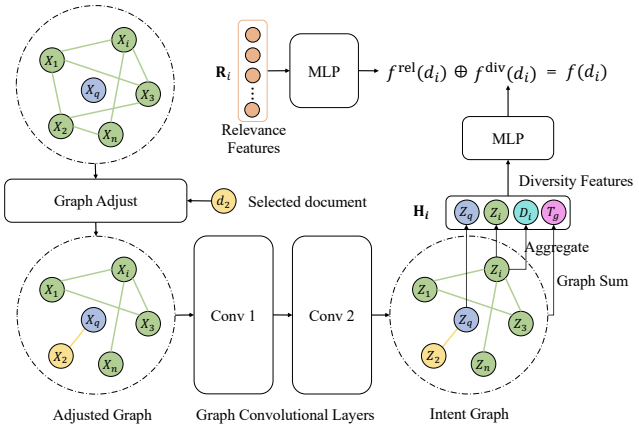

Figure 1: Architecture of Graph4DIV. At step $t = 2$, score $f(d_i)$ of the i-th document d_i is derived from relevance score $f^{\text{rel}}(d_i)$ and diversity score $f^{\text{div}}(d_i)$. Node v_i's degree feature D_i is obtained by Aggregation operation. The feature $\mathbf{T}_g$ of the entire graph is produced by the graph sum operation.

The relevance score $f^{\text{rel}}(d_i)$ is calculated from the relevance feature $\mathbf{R}_i$ with an MLP layer:

$$f^{\text{rel}}(d_i) = \text{MLP}(\mathbf{R}_i). \tag{2}$$

The details will be introduced in Section 3.4.2.

The computation of the diversity score $f^{\text{div}}(d_i, \mathcal{D}, \mathcal{S})$ is the focus of this paper. We propose building an intent graph $\mathcal{G}$ and extract diversity features $\mathbf{H}$ based on the graph. The diversity score is then computed as:

$$f^{\text{div}}(d_i, \mathcal{D}, \mathcal{S}) = \text{MLP}\left(\mathbf{H}\left(d_i, \mathcal{D}, \mathcal{S}\right)\right), \tag{3}$$

$$\mathbf{H}(d_i, \mathcal{D}, \mathcal{S}) = \mathcal{F}(d_i, \mathcal{D}, \mathcal{S}, \mathcal{G}_{D,S}), \tag{4}$$

where $\mathcal{G}_{D,S}$ is the corresponding intent graph for query q that is updated after $\mathcal{S}$ is selected from $\mathcal{D}$. Note that q also belongs to the nodes of this graph but the notation is omitted here for simplification and space saving. The diversity features $\mathbf{H}_i$ of the document d_i is dynamically changing at each step t in the document selection process, and we also omit the notation t for convenience. The function $\mathcal{F}$ describes how our model produces the representation of document d_i and related diversity features when given the intent graph $\mathcal{G}_{D,S}$, the selected documents set $\mathcal{S}$, and document set $\mathcal{D}$.

The key components of our `Graph4DIV` for computing $\mathbf{H}_i$ are briefly introduced as follows:

(1) **Graph Building and Adjustment (Section 3.3).** We build an intent graph for each query q based on the result of the documents relation classifier (introduced in Section 3.3.3). In the intent graph, the query and its all candidate documents are represented as nodes. The query node is only connected to the selected documents in order to obtain a context-aware query representation. For the remaining candidate documents, there will be an edge between two candidate document nodes only when they share the same intent of the query. The graph is dynamically adjusted according to the selection of documents at each step. For example, as shown in Figure 2, at the time step $t = 2$, given the previous selected document d_2, we adjust the graph by dropping the edges between the selected document node v_2 and the remaining candidate document nodes $v_1, v_3, \cdots, v_n$, and connecting the selected document node v_2 to the query node v_q with the wight of the relevance score (elaborated in Section 3.3.2).

Algorithm 1 Diversity Ranking algorithm of Graph4DIV

Procedure Graph4DIV Diversity Ranking
Input: query q, document set $\mathcal{D}$, and initial intent graph $\mathcal{G}_{D,\phi}$.
Output: diversity ranking sequence $\mathcal{R}$ for query q.
$\mathcal{S} \leftarrow \emptyset$
$C \leftarrow \mathcal{D}$ //initial state $C = \mathcal{D} \setminus \mathcal{S} = \mathcal{D}$
$t \leftarrow 0$
while C **do**
  $t \leftarrow t + 1$
  $\mathcal{P}_t \leftarrow \emptyset$ //$\mathcal{P}_t$ is the score set of the candidates at t step
  for document $d_i \in C$ **do**
    $\mathcal{P}_t \leftarrow \mathcal{P}_t \cup \{ f(d_i, \mathcal{D}, \mathcal{S}) \}$ //append the score of d_i
  end for
  $d^* = \text{getbest}(\mathcal{P}_t)$
  $\mathcal{G}_{D,S} \leftarrow \text{GraphAdjust}(q, d^*, \mathcal{G}_{D,S})$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{d^*\}$
  $C \leftarrow C \setminus \{d^*\}$
end while
$\mathcal{R} \leftarrow \mathcal{S}$
return $\mathcal{R}$

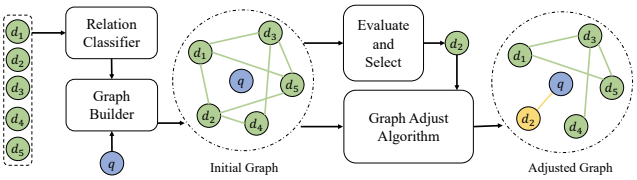

Figure 2: Adjustment of intent graph based on the selection result of candidate documents.

(2) **Graph-based Diversity Features (Section 3.4).** We then compute the diversity features based on the current intent graph. Specifically, considering the initial node representations $\mathbf{X} = [\mathbf{X}_q, \mathbf{X}_1, \cdots, \mathbf{X}_n]$ of all the nodes on the graph, they are updated after a two-layer graph convolutional network. As a result, we can get the new representations $\mathbf{Z} = [\mathbf{Z}_q, \mathbf{Z}_1, \cdots, \mathbf{Z}_n]$ for them. To compute the diversity features $\mathbf{H}_i$, we consider the query's representation $\mathbf{Z}_q$, document d_i's representation $\mathbf{Z}_i$, the degree D_i of the node v_i, and the representation $\mathbf{T}_g$ of the entire intent graph. The diversity features of d_i are calculated as the assemble of these features $\mathbf{H}_i = [\mathbf{Z}_q; \mathbf{Z}_i; D_i; \mathbf{T}_g]$ (illustrated in Section 3.4.2).

The overall process of our proposed GraphDIV for search result diversification is summarized as Algorithm 1.

3.3 Intent Graph

Measuring the similarity of two documents is the foundation of the implicit diversity approaches. In the search result diversification task, we treat the similarity of documents as the similarity of subtopic covering. To model the relationship of multiple document pairs simultaneously and extract more comprehensive diversity features containing both local and global information, we present all the documents $d_i \in \mathcal{D}$ and the query q on the graph, which is called the *intent graph*.

Algorithm 2 Graph Adjustment algorithm used by Graph4DIV

Procedure GraphAdjust
Input: query q, selected document d_k and the intent graph $\mathcal{G}_{D,S}$ for query q at step t.
Output: Adjusted intent graph $\mathcal{G}_{D,S}$.
$\mathcal{N}_k \leftarrow$ getNeighbors$(\mathcal{G}_{D,S}, d_k)$
for document $d_i \in \mathcal{N}_k$ **do**
    $\mathcal{G}_{D,S} \leftarrow$ removeLink$(\mathcal{G}_{D,S}, d_i, d_k)$
end for
$\mathcal{G}_{D,S} \leftarrow$ addLinktoQuery$(\mathcal{G}_{D,S}, q, d_k)$
return $\mathcal{G}_{D,S}$

The intent graph is an essential part of our approach to model the document-document and query-document relationship for diversification. We build one intent graph $\mathcal{G} = (N, E)$ for each query $q, q \in \mathcal{Q}$, where N denotes the nodes, and E denotes the edges. $\mathcal{G}$ is an undirected graph and its nodes N are comprised of the current query q and all documents contained in $\mathcal{D}$. The edges will be dynamically adjusted after a new document is selected and added to $\mathcal{S}$.

The building and adjustment procedure of the intent graph are shown in Figure 2. We build a document relation classifier to judge the subtopic covering relationship of documents. Such a relation is represented as edges between document nodes. Based on the result of the classifier, the graph builder will build the initial intent graph with query node and document nodes. Then the graph adjustment algorithm will refine the intent graph according to the document selection result at each step. Next, we will introduce the critical parts of our workflow in detail.

3.3.1 Graph Builder. First of all, we create an intent graph $\mathcal{G}_0$ with the current query q and all documents contained in $\mathcal{D}$ as the nodes, and an empty edge set, *i.e.*, $N(\mathcal{G}_0) = \{v_q, v_1, \cdots, v_n\}$ and $E(\mathcal{G}_0) = \phi$. Then, we build a document-document relation classifier to predict the relationship between two documents. As the target of search result diversification is to improve result diversity, and the common way to measure diversity is based on intent [1, 4, 7, 44]. Inspired by this, we train a classifier to explicitly judge whether two documents belong to the same intent and we consider this is a simple but effective way to predict the connection between documents. More details will be elaborated in Section 3.3.3. After getting the prediction result of all the pairs of candidate documents, the graph builder will connect the document nodes that are predicted to belong to the same intent and get the initial graph $\mathcal{G}_{\mathcal{D},\mathcal{S}}$ and currently we have $\mathcal{S} = \phi$. In our approach, we treat edge weight between documents as a binary value.

3.3.2 Graph Adjustment after Document Selection. Given the current intent graph $\mathcal{G}_{D,\mathcal{S}}$, we will employ the document scoring algorithm (introduced in Section 3.4) to assess each document in the remaining documents $C = \mathcal{D} \setminus \mathcal{S}$. Consistent with the diversification algorithm, we divide the document nodes in N into two sets: the selected documents $\mathcal{S}$ and the remnant documents C.

Assuming that the best document d^* with the highest score is selected and appended to $\mathcal{S}$, we use Algorithm 2 to adjust the intent graph. Considering that some parts of the user's information needs might be met when the document d^* is selected, we hope the model to focus more on the intents that have not been covered by the selected documents set $\mathcal{S}$ yet. Motivated by this, we propose leveraging $\mathcal{S}$ to update the context-aware query representation. We connect the query node with the nodes within $\mathcal{S}$. With these edges, the representation of the current information need in the query can be updated based on the selected documents via a graph neural network (*e.g.*, GCN). Furthermore, we mainly exploit the remnant documents to obtain the dependent representation of remaining documents, hence we drop all the edges between documents in $\mathcal{S}$ and those in C. More specifically, after d^* is selected, we add an edge to connect d^* and q with the relevant score as edge weight in order to help update the context-aware query representation. The relevant score is the normalized form of the initial ranking score without considering diversity. We then remove all the edges connecting d^* and other documents in C.

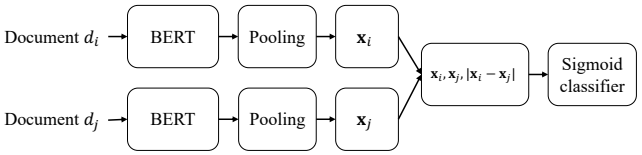

Figure 3: Documents Relation Classifier based on BERT.

3.3.3 Documents Relation Classifier. To convert the complicated and invisible relationship of the query and documents to the edges on the intent graph, we design a classifier to explicitly judge whether two documents cover the same subtopic according to the document's content. Instead of getting the document's relationship from the document's representation, we hope our model can integrate the relation of documents and the query into their representations. Such relation information of the documents comes from the prediction results of the documents relation classifier.

The main structure of the classifier is shown in Figure 3. Given a query q and its document set $\mathcal{D}$, we sample all the document pairs from $\mathcal{D}$ and send them to the relation classifier. Supposing that a pair of documents (d_i, d_j) is given, the documents relation classifier is expected to judge whether d_i and d_j share the same subtopic. To mine the subtopic information from the documents, we leverage BERT [29] to extract the representation of documents d_i and d_j. Here we employ BERT because it is pre-trained on the large corpus and has achieved great performance on several natural language processing tasks [6, 13, 17, 21, 24]. Other advanced text matching models can also be a potential choice for building such a classifier. For the convenience of processing, the two documents are tokenized into a fixed length, say M. Therefore, we can get the token sequences $\mathbf{x}_i = [[\text{CLS}], w_1, w_2, \cdots, w_M]$ and $\mathbf{x}_j = [[\text{CLS}], t_1, t_2, \cdots, t_M]$ standing for documents d_i and d_j, respectively, where "[CLS]" is a special token. Thereafter, we obtain the representation $\mathbf{x}_i$ and $\mathbf{x}_j$ based on the representations of the "[CLS]" tokens computed by BERT. Considering that the difference of two document's representations may contain the useful information for the classifier, we use the feature $\mathbf{x}_{ij} = [\mathbf{x}_i; \mathbf{x}_j; |\mathbf{x}_i - \mathbf{x}_j|]$ as the joint representation of document d_i and d_j. Furthermore, we can derive $c_{ij} = \text{MLP}(\mathbf{x}_{ij})$, which is the judge of d_i and d_j given by the documents relation classifier. $c_{ij} = 1$ denotes that the document

d_i and d_j might cover the same intent, while $c_{ij} = 0$ implies that the document d_i and d_j are less likely to share the joint intent.

Assuming that the number of all documents is $n = |\mathcal{D}|$, the total number of intent graph's nodes is $n + 1$ since we present the query and all documents on the graph. Based on the result of the document relation classifier, we can derive the adjacent matrix $\mathbf{A}$ for the initial intent graph $\mathcal{G}_{D,\phi}$, where $\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)}$. The adjacent matrix $\mathbf{A}$ is defined as:

$$\mathbf{A}[i,j] = \begin{cases} 0, & \text{if } i = 1 \text{ or } j = 1; \\ c_{(i-1)(j-1)}, & \text{else.} \end{cases} \tag{5}$$

Here $\mathbf{A}[i,j]$ is the i-th row and j-th column element of $\mathbf{A}$, which stands for the relation of document d_{i-1} and d_{j-1} ($i \geq 1$ and $j \geq 1$).

According to the Algorithm 2, the adjacent matrix $\mathbf{A}$ dynamically changes in the document selection process. Given the selected document d_k at step t, we set $\mathbf{A}[i,k] = \mathbf{A}[k,i] = 0$ for $i \in [2, n+1]$ to drop all the edges between d_k and other documents. Then we set $\mathbf{A}[1,k] = \mathbf{A}[k,1] = r_k$ to connect the query node and the document node v_k, where r_k is the relevance score of initial ranking without considering diversity.

It is worth noting that our classifier is only trained to predict whether two documents belong to the same intent. We do not predict whether a single document contains a specific intent, which is still a challenging task. This is also why our method is considered as an implicit method.

3.4 Diversified Scoring based on Graph

As introduced in Section 3.1, in order to make a better document selection, we take both the relevance feature of the document and the diversity feature extracted from the intent graph into consideration. As we want to take the global document's relationship into account and represent the dynamic information need of the query, we propose leveraging the dynamic intent graph in the duration of document selection.

3.4.1 Representation Learning via GCN. Given the initial representation $\mathbf{X} = [\mathbf{X}_q, \mathbf{X}_1, \cdots, \mathbf{X}_n]$ of the query and document nodes, $\mathbf{X}_q$ is the distributed representation of query q, while $\mathbf{X}_i$ is the initial representation of the document d_i. Then we can update the representation using the information presented on the intent graph and get the new feature vectors $[\mathbf{Z}_q, \mathbf{Z}_1, \cdots, \mathbf{Z}_n]$ of each node with local and global information. Instead of using document representations to calculate similarity, we hope to use similarity to generate document representations. Specifically, we leverage graph convolutional network (GCN) to aggregate neighbor's intent information to produce new document representation. With the help of GCN, the representations of documents will be enhanced by their neighbors with similar intents. The diversity features extracted by the GCN will be used to produce the diversity score of the documents.

In the first stage, documents nodes on the graph aggregate all the neighbors' feature vectors within a predefined scope K. Then the document nodes update their representation by the information collected from their neighbors. The procedure is conducted layer by layer. In this work, the scope K is determined by the layer num L of the GCN, namely, $K = L$. According to our experiment, we set $L = 2$. Concretely, supposing $\mathbf{A}$ is the corresponding adjacent matrix for the current intent graph $\mathcal{G}_{D,S}$, $\mathbf{Z}^{(0)} = \mathbf{X}$ is the initial representation of the nodes on the graph, we use GCN to calculate the features of the current nodes as follows:

$$\mathbf{Z}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(l)}\mathbf{W}^{(l)}), \tag{6}$$

$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N, \tag{7}$$

where $l \in [0, L)$ is the identify of each layer in the GCN; $\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)}$ is the current adjacency matrix of the undirected intent graph and $\tilde{\mathbf{D}}[i,i] = \sum_j \tilde{\mathbf{A}}[i,j]$; n is the number of all candidate documents of query q; $\mathbf{I}_N$ is the identity matrix; $\mathbf{Z}^{(l)} \in \mathbb{R}^{(n+1)\times D}$ is the matrix of node features where D is the dimension size of the node features; $\mathbf{W}^{(l)}$ is a layer-specific trainable weight matrix for l-th layer; and $\sigma(\cdot)$ is an activation function, *e.g.*, $\text{ReLU}(\cdot) = \max(0, \cdot)$ or $\tanh(\cdot)$.

Table 3: Relevance features used by previous methods

Name	Description	# Features
TF-IDF	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

3.4.2 Relevance and Diversity Features. As shown in Equation (1), we score each candidate document d_i based on relevance and diversity. Following previous work [16, 20, 28, 46], the relevance score $f^{\text{rel}}(d_i)$ is produced (illustrated in Equation (2)) by the traditional relevance features $\mathbf{R}_i$, including BM25, TF-IDF, PageRank, etc. The whole list of features is shown in Table 3 and is consistent with that in [16, 20, 28].

The diversity score $f^{\text{div}}(d_i, \mathcal{D}, \mathcal{S})$ is calculated (illustrated in Equation (3)) based on the diversity features $\mathbf{H}(d_i, \mathcal{D}, \mathcal{S})$ extracted from the current intent graph $\mathcal{G}_{D,S}$:

$$\mathbf{H}_i = [\mathbf{Z}_q; \mathbf{Z}_i; D_i; \mathbf{T}_g], \tag{8}$$

where $\mathbf{H}_i$ consists of the current query embedding $\mathbf{Z}_q$, document embedding $\mathbf{Z}_i$, degree feature D_i, and the whole graph representation $\mathbf{T}_g$. $[;]$ means the concatenation operation. We have:

$$\mathbf{Z}_q = \mathbf{Z}^{(L)}[1], \quad \mathbf{Z}_i = \mathbf{Z}^{(L)}[i+1], \tag{9}$$

$\mathbf{Z}_q$: The representation of query q. To make a fair comparison with the previous work [16, 20, 28, 38, 46], we use doc2vec embedding as the initial query and the documents representation. Based on the intent graph, the representation of the query contains the information of selected documents, which can dynamically change when the graph is adjusted. With the dynamic representation of query q, our model can model the information needs of the query precisely and timely.

$\mathbf{Z}_i$: The representation $\mathbf{Z}_i$ of document d_i, which contains the local information by aggregating the neighbor's features of document node v_i. We have $\mathbf{Z}_i = \mathbf{Z}^{(L)}[i+1]$ from the GCN.

D_i: The degree of document d_i on the intent graph. For the diverse documents may share more edges with other documents, the degree of the node v_i in the intent graph is an essential measure to evaluate the diversity of document d_i. We have $D_i = \sum_{j=2}^{n+1} \mathbf{A}[i,j]$. Since we only use the diversity features of the remaining candidate

documents, we omit the edges connecting the query node and selected document nodes when generating D_i.

$\mathbf{T}_g$: The representation of the whole graph obtained by summing the representations of all documents. We have $\mathbf{T}_g = \sum_{i=1}^{n+1} \mathbf{Z}^{(L)}[i]$. Derived from all the nodes vectors, the feature $\mathbf{T}_g$ is the global feature of the entire intent graph. Combined with local and global features, our approach can consider the comprehensive information in the diversification task.

3.5 Training and Optimization

The training for our model can be divided into two phases, the training for document relation classifier and the training for diversity ranking. They will be introduced as follows.

(1) **Classifier Training.** We use the diversity judgements to generate the training data for document relation classifier. For example, if d_1 and d_2 cover one intent, and d_2 and d_3 cover another intent, we can obtain $(d_1, d_2, 1)$ and $(d_2, d_3, 1)$ as positive samples, while $(d_1, d_3, 0)$ as a negative sample. With the generated data, we use a binary cross-entropy to train the classifier. In our experiment, the positive samples are much less than the negative samples (about 1:8). To accelerate the training process and avoid bias, we randomly discard negative samples to keep the ratio as 1:1.

(2) **Diversity Ranking.** Given a query set $\mathcal{Q}$, the diversity ranking $\mathcal{R}_q$ is produced based on Algorithm 1 (Graph4DIV):

$$\mathcal{R}_q = \text{Graph4DIV}(q, \mathcal{D}_q, \mathcal{G}_{D,\phi}), \tag{10}$$

$$f = \arg\min \sum_{q \in \mathcal{Q}} \sum_{o \in O} \mathcal{L}(\mathcal{R}_q, Y_o), \tag{11}$$

where $q \in \mathcal{Q}$, Y_o is a ground-truth ranking of training sample of query q, and $\mathcal{L}$ is the loss function of the model.

We follow the previous studies [16, 28] and utilize the list-pairwise loss function for optimization. The generation of ground-truth ranking Y_o follows the procedure of list-pairwise training [16]. This loss function is computed based on the same sampling strategy proposed by Jiang et al. [16]. The sample pair (r_1, r_2) could be presented as (C_{t-1}, d_1, d_2), where C_{t-1} is the same previous document sequence shared by r_1 and r_2. Ranking r_1 and r_2 are different only at t-th position. The list-pairwise loss can be defined as:

$$\mathcal{L} = \sum_{q \in \mathcal{Q}} \sum_{o=1}^{|O_q|} w^o \left(y^o \log(P(r^o_{1,2}) + (1 - y^o) \log\left(1 - P(r^o_{1,2})\right)\right),$$

where O_q is the pair samples set of query q. $P(r^o_{1,2})$ is the probability of pair (r_1, r_2) to be positive, and w^o is the weight of pair sample (r_1, r_2) based on the metric function M as:

$$P(r^o_{1,2}) = \frac{1}{1 + \exp(s^o_{r_2} - s^o_{r_1})}, \tag{12}$$

$$w^o = |M(r^o_1) - M(r^o_2)|, \tag{13}$$

where M is the metric function that evaluates the quality of model's diversity ranking. $M(r_1) > M(r_2)$ shows that (r_1, r_2) is a positive pair, while $M(r_1) < M(r_2)$ implies (r_1, r_2) to be negative.

4 EXPERIMENTS

4.1 Data Collections

We use the Web Track dataset [2] from 2009 to 2012, which is the same as previous work [16, 20, 28]. There are 200 queries in the dataset in total. However, since query #95 and #100 have no diversity judgements, only 198 queries are used in our experiment. There are 3 to 8 subtopics for each query. In the experiment, the subtopic features are only used to train explicit baseline methods and they are not used in our model.

4.2 Evaluation Metrics

We use various metrics that are used by lots of previous research, including α-nDCG [7], ERR-IA [4], and NRBP [8], which are official diversity evaluation metrics used in Web Track. They measure the diversity by explicitly rewarding novelty and penalizing redundancy. Besides, we also use the diversity measure Subtopic Recall (denoted as S-rec, a.k.a. I-rec) [44]. Consistent with previous diversification models [16, 28, 37, 38, 46] and TREC Web Track, we adopt the top 50 results of Lemur[2] for diversity re-ranking, and all evaluation metrics are computed on the top 20 results of a document ranking list. Two-tailed paired t-test is used to conduct significance testing with p-value < 0.05.

4.3 Baseline Models

We compare our `Graph4DIV` with various methods including:

(1) **Non-diversified methods.** `Lemur`: We use the same adhoc results as [14, 16] for fair comparison. Results of Lemur are produced by Indri engine using language model. `ListMLE` [36] is a learning-to-ranking method without considering diversity.

(2) **Explicit methods.** `xQuAD` [31], `PM2` [9], `TxQuAD`, `TPM2` [10], `HxQuAD`, and `HPM2` [14] are some representative unsupervised explicit baseline methods. Similar to our method, all these methods balance the importance of relevance and diversity by a parameter λ. Based on the hierarchical structure, `HxQuAD` and `HPM2` adopt an additional parameter α to control the weight of subtopic layers. `DSSA` [16] is a supervised explicit diversification method, which models the diversity of the documents with subtopic attention at each step in the document selection process using RNNs. Explicit methods are shown to be more effective than the implicit methods in existing studies [9, 14, 16, 20, 28]. Note that our proposed `Graph4DIV` does not use subtopics and it is an implicit method.

(3) **Implicit methods.** `R-LTR` [46], `PAMM` [37], and `NTN` [38] are the representative supervised implicit methods. For `PAMM`, we use α-nDCG@20 as the optimization metrics and tune the number of positive rankings l^+ and negative rankings l^- per query. The neural tensor network (`NTN`) is used on both `R-LTR` and `PAMM`, denoted as `R-LTR-NTN` and `PAMM-NTN`, respectively.

(4) **Ensemble methods.** `DESA` [28] and `DVGAN` [20] are two ensemble methods that use both explicit (subtopic) features and implicit features. `DESA` leverages self-attention based encoder-decoder structure to model the interactions between documents and subtopics. With the framework of the generative adversarial network, `DVGAN` is able to generate training data that combine both explicit and implicit features.

[2]Lemur service: http://boston.lti.cs.cmu.edu/Services/clueweb09 batch/

Table 4: Performance comparison of all methods. The baselines include: (1) non-diversed methods; (2) explicit methods; (3) implicit methods; and (4) ensemble methods. The best result is in bold. † indicates significant improvements obtained by GraphDIV in t-test with p-value< 0.05.

	ERR-IA	α-nDCG	NRBP	S-rec
(1) Lemur	.271†	.369†	.232†	.621†
(1) ListMLE	.287†	.387†	.249†	.619†
(2) xQuAD	.317†	.413†	.284†	.622†
(2) TxQuAD	.308†	.410†	.272†	.634
(2) HxQuAD	.326†	.421†	.294†	.629
(2) PM2	.306†	.411†	.267†	.643
(2) TPM2	.291†	.399†	.250†	.639
(2) HPM2	.317†	.420†	.279†	.645
(2) DSSA	.356	.456	.326	.649
(3) R-LTR	.303†	.403†	.267†	.631
(3) PAMM	.309†	.411†	.271†	.643
(3) R-LTR-NTN	.312†	.415†	.275†	.644
(3) PAMM-NTN	.311†	.417†	.272†	.648
(3) Graph4DIV (Ours)	**.370**	**.468**	**.338**	**.666**
(4) DESA	.363	.464	.332	.653
(4) DVGAN	.367	.465	.334	-

4.4 Implementation Details

For building the relation classifier in our model, we use the pre-trained BERT provided by HuggingFace [35] and fine tune it for our relation classification. The maximum token sequence length M is 512 in BERT. The batch size is 16. We use AdamW [23] as the optimizer with the learning rate of 3e-5. The number of epochs is set as 3. As we use 5-fold cross-validation for training and testing our Graph4DIV, we also train 5 classifiers correspondingly. For training Graph4DIV model, we adopt doc2vec embeddings with the dimension of 100 as the initial document representations on the intent graph, which is the same as previous work [16, 28, 46]. The number of GCN layers is tuned in {1,2,3}, and the learning rate is tuned from 1e-10 to 1e-2 and set as 8e-4. For balancing the weights of relevance score and diversity score in Equation (1), we tune the λ in $\{0.1, 0.2, \cdots, 0.9\}$ and finally set $\lambda = 0.5$. All hyper-parameters are selected by 5-fold cross-validation based on the result of α-nDCG@20. Our code is is available at https://github.com/su-zhan/Graph4DIV.git.

4.5 Experimental Results

The overall results are shown in Table 4. We find that GraphDIV outperforms all explicit, implicit, and ensemble methods. This result clearly demonstrates the superiority of our method.

(1) Graph4DIV significantly outperforms all implicit methods (R-LTR, PAMM, R-LTR-NTN, and PAMM-NTN) in terms of ERR-IA, α-nDCG, and NRBP (t-test with p-value < 0.05). This result proves the effectiveness of our proposed Graph4DIV on search result diversification. Specifically, R-LTR-NTN and PAMM-NTN are two state-of-the-art implicit supervised methods. They calculate the document's novelty by using a neural tensor network based on the document representations. Compared with these two methods, Graph4DIV improves the absolute value of α-nDCG by more than 5%. This indicates that our proposed intent graph can better represent the complicated relationship among several documents than only considering the pairwise similarity between the single candidate document and each of the selected documents. Besides, our GraphDIV does not rely on many artificial features used in R-LTR, which improves its applicability in real scenarios.

(2) Graph4DIV also outperforms explicit methods by a large margin, including DSSA, which is the state-of-the-art explicit method. Indeed, either the methods based on xQuAD or those based on PM2 are unsupervised approaches. The better performance obtained by DSSA and Graph4DIV proves the great advantage of using the supervised method for learning the ranking function in search result diversification. Furthermore, compared with DSSA which explicitly models subtopic coverage, our Graph4DIV only measures the novelty of each document implicitly based on our proposed intent graph through a GCN. Surprisingly, Graph4DIV can still achieve better performance than DSSA regarding all metrics. This demonstrates that the intent graph can well reflect the relation between documents by considering their underlying intents, which is extremely helpful for diversifying the search results.

(3) Interestingly, we also find Graph4DIV can perform slightly better than ensemble methods, *i.e.*, DESA and DVGAN. DESA leverages the self-attention mechanism to measure the similarity between documents and enhances the document representations. The similarity computed by the self-attention mechanism is only tuned by the final diversification loss. On the contrary, the similarity of documents used in Graph4DIV is computed by a fine-tuned BERT with supervision signals of shared intents. With such separated and clear supervision, the similarity between documents can be better captured. On the other hand, we employ a GCN to integrate the similarity information into document representations. Only the documents within a predefined scale can be aggregated, thus avoiding the noise in irrelevant documents. As future work, we plan to equip our Graph4DIV with explicit subtopic features and make it as an ensemble model, which may bring further improvements.

5 DISCUSSIONS

To better analyze our model, we further investigate two research questions: (1) How is the performance of the relation classifier and what is the influence on the final results? (2) What is the effect of each component in Graph4DIV and how is the performance of it with other settings?

5.1 Performance and Influence of Classifier

In our model, the intent graph is built based on the predicted relations of documents. Therefore, the performance of this classifier may influence the final diversification results. Since the whole experiment is conducted with the 5-fold cross-validation, we build five document relation classifiers for the corresponding folds, respectively. The average Accuracy of five fold-classifiers is 0.792 and the average F1 is 0.884. Due to the difficulty of judging two documents whether belong to the same subtopic using only the content, we believe it is a good performance for the classifier and it is also a meaningful exploration for a better utility of document's relation in diversification task. Compared with the excellent performance

Table 5: Comparison between Graph4DIV and Ground-truth.

	ERR-IA	α-nDCG	NRBP	S-rec
Graph4DIV	.370	.468	.338	.666
Ground-truth Intent Graph	.477	.563	.461	.673
Ground-truth Ranking	**.574**	**.657**	**.568**	**.706**

Table 6: Performance of Graph4DIV with different settings.

	ERR-IA	α-nDCG	NRBP	S-rec
Graph4DIV	**.370**	**.468**	**.338**	**.666**
w/o Query Rep.	.355	.455	.322	.653
w/o Graph Features	.347	.446	.315	.650
w/o Doc. Degree	.352	.452	.319	.657
w/o Graph Adjustment	.361	.461	.328	.666
w/ GIN	.361	.458	.330	.655
1-layer-GCN	.353	.453	.318	.653
3-layer-GCN	.355	.458	.320	.666

of Graph4DIV, it implies the robustness and generalization of our approach.

To further investigate the upper bound of using the intent graph in the search result diversification task, we use the ground-truth label for building the graph, where the edge between two documents certainly indicates they belong to the same subtopic. The result is shown in Table 5, which is denoted as Ground-truth Intent Graph. According to the results, we can observe that there is still a large gap between Graph4DIV and that using the ground-truth intent graph. This demonstrates the potential of using the intent graph for diversification. As the average accuracy of our document relation classifier is around 0.8, we speculate that building a better classifier may bring further improvements for Graph4DIV. As a reference, we also provide the result of the Ground-truth Ranking. It is clear to see that only leveraging the intent graph is not enough for the diversification task. The potential reasons include: (1) The intent graph only reflects whether two documents belong to the same intent, but does not contain the information of the specific subtopics that the document belongs to. Missing such information, it is still hard to select the optimal document at each step. (2) The GCN can aggregate documents that may share the same user intents and update their representations. However, it is difficult for GCN to infer the subtopic coverage since the novelty of each document can only be measured implicitly by its similarity with other documents.

5.2 Effects of Different Settings

We also investigate the influence of different settings on the performance of Graph4DIV. The results are shown in Table 6.

(1) **Ablation of diversity features.** Since the diversity features contain various features extracted from the intent graph, we explore the effectiveness of them by removing them one by one from the full model. The document representations are basic features, we only remove the other three ones, namely, the query representation $\mathbf{Z}_q$ (w/o Query Rep.), the graph features $\mathbf{T}_g$ (w/o Graph Features), and the document degrees D_i (w/o Doc. Degree). In general, removing any one of them leads to performance degradation. This demonstrates all three kinds of features are effective in our method. Specifically, the performance drops most when removing the graph features. The potential reason is that the graph features can provide a global view of all remnant documents, which helps to determine the next document.

(2) **Graph Adjustment.** To validate the effect of our proposed graph adjustment strategy, we replace it by only using the initial intent graph for training. This variant is denoted as w/o Graph Adjustment. In this model, the document diversity ranking is directly obtained based on the score derived from the initial graph. It is clear to see that the performance degrades when the graph is static. In our graph adjustment algorithm, the selected document is connected with the query to update its representation. The updated query representation can reflect how many subtopics have already been covered. On the other hand, we break the edge between the selected document and the remaining ones, so that the candidate document cannot affect the representation of the query, which further reduce the noise.

(3) **Different GNNs.** As there are several graph neural networks, we also try other ones, such as Graph Isomorphism Network (GIN) [39], in Graph4DIV. All hyper-parameters are kept the same without careful tuning. We can observe the results with GINs (w/ GIN) are worse compared to the model with GCNs but still competitive. This shows that other GNN models could also be used on our intent graphs and demonstrates the scalability of our method.

(4) **Different number of layers in GNN.** The number of layers in GCNs is also important in our method. It determines how many neighbors are considered when updating the representation of a specific document. Based on the result, we can observe that it is not enough to aggregate documents within only one hop (1-layer-GCN). On the contrary, introducing more layers also hurts the performance. This may stem from the over-smoothing problem, which is often observed in multi-layer GCNs [5, 25, 40]. According to the experimental results, it is suitable for Graph4DIV to use the 2-layer-GCN structure.

6 CONCLUSIONS

In this paper, we propose an implicit supervised approach that models the relationship of multiple document pairs simultaneously with graph structure for search result diversification. We further use the graph convolutional network to extract the diversity features that contain both local and global information. To capture the dynamic information needs of the query, we design a graph adjust algorithm for the intent graph to timely present the situation during the document selection process. The experimental results also confirm that our dynamic intent graph is beneficial and meaningful to generate diversity features for the documents in the diversification task.

In the future, we plan to improve the accuracy of the classifier by combining more information and apply the intent graph to the explicit search result diversification methods.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and No. 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Shandong Provincial Natural Science Foundation under Grant ZR2019ZD06.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, Ricardo Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu (Eds.). ACM, 5–14. https://doi.org/10.1145/1498759.1498766

[2] Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao". 2009. *Clueweb09 data set.* https://boston.lti.cs.cmu.edu/Data/clueweb09/

[3] Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel (Eds.). ACM, 335–336. https://doi.org/10.1145/290941.291025

[4] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin (Eds.). ACM, 621–630. https://doi.org/10.1145/1645953.1646033

[5] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 3438–3445. https://aaai.org/ojs/index.php/AAAI/article/view/5747

[6] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! Born-Again Multi-Task Networks for Natural Language Understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 5931–5937. https://doi.org/10.18653/v1/p19-1595

[7] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong (Eds.). ACM, 659–666. https://doi.org/10.1145/1390334.1390446

[8] Charles L. A. Clarke, Maheedhar Kolla, and Olga Vechtomova. 2009. An Effectiveness Measure for Ambiguous and Underspecified Queries. In *Advances in Information Retrieval Theory, Second International Conference on the Theory of Information Retrieval, ICTIR 2009, Cambridge, UK, September 10-12, 2009, Proceedings (Lecture Notes in Computer Science, Vol. 5766)*, Leif Azzopardi, Gabriella Kazai, Stephen E. Robertson, Stefan M. Rüger, Milad Shokouhi, Dawei Song, and Emine Yilmaz (Eds.). Springer, 188–199. https://doi.org/10.1007/978-3-642-04417-5_17

[9] Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson (Eds.). ACM, 65–74. https://doi.org/10.1145/2348283.2348296

[10] Van Dang and W. Bruce Croft. 2013. Term level search result diversification. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, Gareth J. F. Jones, Paraic Sheridan, Diane Kelly, Maarten de Rijke, and Tetsuya Sakai (Eds.). ACM, 603–612. https://doi.org/10.1145/2484028.2484095

[11] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking Graph Neural Networks. *CoRR* abs/2003.00982 (2020). arXiv:2003.00982 https://arxiv.org/abs/2003.00982

[12] Anjan Goswami, Chengxiang Zhai, and Prasant Mohapatra. 2019. Learning to Diversify for E-commerce Search with Multi-Armed Bandit. In *Proceedings of the SIGIR 2019 Workshop on eCommerce, co-located with the 42st International ACM SIGIR Conference on Research and Development in Information Retrieval, eCom@SIGIR 2019, Paris, France, July 25, 2019 (CEUR Workshop Proceedings, Vol. 2410)*, Jon Degenhardt, Surya Kallumadi, Utkarsh Porwal, and Andrew Trotman (Eds.). CEUR-WS.org. http://ceur-ws.org/Vol-2410/paper18.pdf

[13] Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-Aware BERT for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2041–2044. https://doi.org/10.1145/3340531.3412330

[14] Sha Hu, Zhicheng Dou, Xiao-Jie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search Result Diversification Based on Hierarchical Intents. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, James Bailey, Alistair Moffat, Charu C. Aggarwal, Maarten de Rijke, Ravi Kumar, Vanessa Murdock, Timos K. Sellis, and Jeffrey Xu Yu (Eds.). ACM, 63–72. https://doi.org/10.1145/2806416.2806455

[15] Shan Jiang, Yuening Hu, Changsung Kang, Tim Daly Jr., Dawei Yin, Yi Chang, and ChengXiang Zhai. 2016. Learning Query and Document Relevance from a Web-scale Click Graph. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (Eds.). ACM, 185–194. https://doi.org/10.1145/2911451.2911531

[16] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. 2017. Learning to Diversify Search Results via Subtopic Attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 545–554. https://doi.org/10.1145/3077136.3080805

[17] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing Format Boundaries With a Single QA System. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 1896–1907. https://www.aclweb.org/anthology/2020.findings-emnlp.171/

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[19] Xiangsheng Li, Maarten de Rijke, Yiqun Liu, Jiaxin Mao, Weizhi Ma, Min Zhang, and Shaoping Ma. 2020. Learning Better Representations for Neural Information Retrieval with Graph Information. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 795–804. https://doi.org/10.1145/3340531.3411957

[20] Jiongnan Liu, Zhicheng Dou, Xiao-Jie Wang, Shuqi Lu, and Ji-Rong Wen. 2020. DVGAN: A Minimax Game for Search Result Diversification Combining Explicit and Implicit Features. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 479–488. https://doi.org/10.1145/3397271.3401084

[21] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 4487–4496. https://doi.org/10.18653/v1/p19-1441

[22] Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 1247–1256. https://doi.org/10.18653/v1/d18-1156

[23] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7

[24] Junyu Lu, Xiancong Ren, Yazhou Ren, Ao Liu, and Zenglin Xu. 2020. Improving Contextual Language Models for Response Retrieval in Multi-Turn Conversation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1805–1808. https://doi.org/10.1145/3397271.3401255

[25] Yimeng Min, Frederik Wenkel, and Guy Wolf. 2020. Scattering GCN: Overcoming Oversmoothness in Graph Convolutional Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/a6b964c0bb675116a15ef1325b01ff45-Abstract.html

[26] Thien Huu Nguyen and Ralph Grishman. 2018. Graph Convolutional Networks With Argument-Aware Pooling for Event Detection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.).

AAAI Press, 5900–5907. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16329

[27] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[28] Xubo Qin, Zhicheng Dou, and Ji-Rong Wen. 2020. Diversifying Search Results using Self-Attention Network. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1265–1274. https://doi.org/10.1145/3340531.3411914

[29] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[30] Rodrygo L. T. Santos. 2013. Explicit web search result diversification. *SIGIR Forum* 47, 1 (2013), 67–68. https://doi.org/10.1145/2492189.2492205

[31] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti (Eds.). ACM, 881–890. https://doi.org/10.1145/1772690.1772780

[32] Atish Das Sarma, Nish Parikh, and Neel Sundaresan. 2014. E-commerce product search: personalization, diversification, and beyond. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel (Eds.). ACM, 189–190. https://doi.org/10.1145/2567948.2577272

[33] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A Graph-to-Sequence Model for AMR-to-Text Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, 1616–1626. https://doi.org/10.18653/v1/P18-1150

[34] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017). arXiv:1710.10903 http://arxiv.org/abs/1710.10903

[35] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[36] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008 (ACM International Conference Proceeding Series, Vol. 307)*, William W. Cohen, Andrew McCallum, and Sam T. Roweis (Eds.). ACM, 1192–1199. https://doi.org/10.1145/1390156.1390306

[37] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto (Eds.). ACM, 113–122. https://doi.org/10.1145/2766462.2767710

[38] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2016. Modeling Document Novelty with Neural Tensor Network for Search Result Diversification. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (Eds.). ACM, 395–404. https://doi.org/10.1145/2911451.2911498

[39] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=ryGs6iA5Km

[40] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2020. What Can Neural Networks Reason About?. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rJxbJeHFPS

[41] Jun Yu, Sunil Mohan, Duangmanee Putthividhya, and Weng-Keen Wong. 2014. Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, Ben Carterette, Fernando Diaz, Carlos Castillo, and Donald Metzler (Eds.). ACM, 463–472. https://doi.org/10.1145/2556195.2556215

[42] Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural SVMs. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008 (ACM International Conference Proceeding Series, Vol. 307)*, William W. Cohen, Andrew McCallum, and Sam T. Roweis (Eds.). ACM, 1224–1231. https://doi.org/10.1145/1390156.1390310

[43] Victoria Zayats and Mari Ostendorf. 2018. Conversation Modeling on Reddit Using a Graph-Structured LSTM. *Trans. Assoc. Comput. Linguistics* 6 (2018), 121–132. https://transacl.org/ojs/index.php/tacl/article/view/1083

[44] ChengXiang Zhai, William W. Cohen, and John D. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, Charles L. A. Clarke, Gordon V. Cormack, Jamie Callan, David Hawking, and Alan F. Smeaton (Eds.). ACM, 10–17. https://doi.org/10.1145/860435.860440

[45] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 2390–2400. https://doi.org/10.1145/3308558.3313468

[46] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 293–302. https://doi.org/10.1145/2600428.2609634

[47] Yutao Zhu, Kun Zhou, Jian-Yun Nie, Shengchao Liu, and Zhicheng Dou. 2021. Neural Sentence Ordering Based on Constraint Graphs. *CoRR* abs/2101.11178 (2021). arXiv:2101.11178 https://arxiv.org/abs/2101.11178