Knowledge Enhanced Search Result Diversification

Zhan Su
Zhicheng Dou
suzhan@ruc.edu.cn
dou@ruc.edu.cn
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China

Yutao Zhu
yutao.zhu@umontreal.ca
Université de Montréal, Québec
Canada

Ji-Rong Wen
jirong.wen@gmail.com
Beijing Key Laboratory of Big Data Management and Analysis Methods
Beijing, China
Key Laboratory of Data Engineering and Knowledge Engineering, MOE
Beijing, China

ABSTRACT

Search result diversification focuses on reducing redundancy and improving subtopic richness in the results for a given query. Most existing approaches measure document diversity mainly based on text or pre-trained representations. However, some underlying relationships between the query and documents are difficult for the model to capture only from the content. Given that the knowledge base can offer well-defined entities and explicit relationships between entities, we exploit knowledge to model the relationship between documents and the query and propose a knowledge-enhanced search result diversification approach KEDIV. Concretely, we build a query-specific relation graph to model the complicated query-document relationship from an entity view. Then a graph neural network and node weight adjust algorithm are applied to the relation graph to obtain context-aware entity representations and document representations at each selection step. The diversity features are derived from the updated node representations of the relation graph. In this way, we can take advantage of entities' abundant information to model document's diversity in search result diversification. Experimental results on commonly used datasets show that our proposed approach can outperform the state-of-the-art methods.

CCS CONCEPTS

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

KEYWORDS

Search Result Diversification, Knowledge Base, Relation Graph

ACM Reference Format:
Zhan Su, Zhicheng Dou, Yutao Zhu, Ji-Rong Wen. 2022. Knowledge Enhanced Search Result Diversification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539459

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539459

1 INTRODUCTION

Search result diversification aims to improve the quality of the search results by adjusting the ranking of documents that cover different subtopics. Different from Ad-hoc retrieval, search result diversification approaches are expected to offer relevant but distinct documents to the given query. Therefore, how to mine the exact query intents and remove redundant documents is an essential part of the search result diversification task. Existing approaches could be roughly categorized into explicit and implicit approaches. Explicit methods model a document's diversity based on its subtopic coverage [9, 17, 19, 25, 27], while implicit methods measure a document's novelty via modeling its dissimilarity with selected documents and candidate documents [4, 35, 50]. To reduce the computational complexity of sampling, most approaches adopt the greedy selection strategy.

However, most existing approaches model document diversity mainly based on the text-similarity of the documents, some subtle but essential relationships between the query and documents may be hard for the model to learn only based on document content.

From our perspective, it is not enough to model document diversity only depending on the text since the query is often ambiguous. Given that entities often appear in the documents and the explicit relationship of the entities in the knowledge base is well-defined and meaningful. It is a natural motivation to leverage the entities and their relations to bridge the gap between the query and the documents. There are at least two benefits of utilizing a knowledge base in search result diversification: (1) The explicit meanings contained by the entities can help the word disambiguation and infer the topics of the documents. (2) The abundant information and well-organized relations brought by the knowledge base can reflect the potential relations of the documents, which is hard to learn only from the text. Previous studies [14, 16] have demonstrated that knowledge bases are useful resources to help understand the meanings of the query. However, knowledge can be further leveraged more than that. Concretely, we can exploit the entities extracted from the query and the documents to establish the relations between the documents and the query in the search result diversification. For example, as shown in Figure 1, document d_1 is more likely to cover the astronomy topic for containing the entity "Titan (Saturn's moon)", while document d_2 is more possible to deal with sports for mentioning the entity "Titan (football team)". From the query's aspect, it is hard to determine the exact query intent only using the term "titan". However, with the help of entity linking, it is easier to discover the relations of the query "titan" to the different "Titan"

Query Titan

	Document Content	Entity Linking
d_1	Scientists have determined that Saturn's moon Titan has hundreds of times more liquid hydrocarbons than all the known oil and natural gas reserves of Earth.	Titan (Saturn's moon) Category: Astronomy
d_2	Save money with discount tickets to 2009 Tennessee Titans games. Buy cheap season tickets to Titans football games…	Titan (football team) Category: Sports
d_3	Shop online from the latest Titan Watches for men & women in India. Choose from casual & formal Titan Watches in various.	Titan (watches) Category: Business

Figure 1: An example of using entity linking in the documents for search result diversification.

entities that occur in different documents. More importantly, the abundant information of the entity (such as categories) can be used for measuring the document's diversity.

Since the knowledge is powerful and suitable for search result diversification, we propose a **K**nowledge **E**nhanced search result **DIV**ersification approach **KEDIV** to leverage entities and their relations to model the document's diversity. Specifically, we can extract the essential entities that have been commonly mentioned by the query and the documents. Moreover, the entities that frequently occur in most documents are also taken into consideration. To fully exploit the abundant information contained in the knowledge base, we extract four essential relations (co-occurrence, category, knowledge, and document-entity relations) related to the task of the search result diversification from the entities. Then we build a query-specific relation graph to comprehensively model the relations of the query, entities, and documents. On the relation graph, the query, entities, and documents are presented as nodes, while four relations are converted to the edges between them. The relation graph is both effective and explainable. For example, the document nodes on the relation graph are connected by the entity nodes that belong to them. Then we could measure the document's novelty based on the entity set contained by them. To further aggregate the information on the relation graph, we apply a graph neural network to update the representations of the nodes on the graph. Considering that the novelty of the entities and documents changes during the document selection procedure, we dynamically adjust the node weight of the entity nodes and derive the diversity features from the context-aware entity and document representations. Experimental results on commonly used datasets demonstrate the superiority of our proposed method KEDIV over existing state-of-the-art models. A series of further studies validate the effectiveness of our proposed relation graph and its adjusting algorithm.

Our contributions are mainly in these aspects:

(1) We propose exploiting entities and their relationship to measure the document novelty in search result diversification, which has been neglected by most diversification methods.

(2) We extract four essential relations from the knowledge base to model the relations of the documents and the query. Furthermore, we build a query-specific relation graph to comprehensively model the relations of the query, entities, and documents.

(3) We dynamically adjust the node weights according to the document selection and derive context-aware document representations for diversified scoring.

2 RELATED WORK

2.1 Search Result Diversification

Search result diversification needs to measure the document's novelty and redundancy. Explicit approaches leverage document's coverage of different query aspects to figure out the documents that cover the novel subtopics, while implicit methods focus on modeling documents' relationship (*e.g.*, dissimilarity) to reduce redundancy.

Implicit Methods. MMR [4] is an early unsupervised method that leverages a parameter λ to balance the document's relevance and novelty. Since the diversification task is an NP-hard problem, MMR made a greedy selection strategy that chooses the most novel document at each step. Recently, several supervised methods have been proposed. For example, SVM-DIV [45] learned to predict diverse subsets of the documents with a structural SVM framework. R-LTR [50] formulated the diversification as a special relational learning-to-rank task based on the various handcrafted novelty features. PAMM [35] maximized the distance between the positive and negative rankings. Based on R-LTR and PAMM, NTN [36] automatically generated document's diversity features via neural tensor network. Yan et al. [43] proposed an approximate loss to directly measure the ranking quality of the entire document list. Graph4DIV [28] treated documents' similarity as intent coverage similarity. Compared with these methods, though our approach is also a supervised approach, we model the relationship between the query and documents based on entities and their relationships. Besides, our approach does not depend on an external classifier like that used in Graph4DIV.

Explicit Methods. Explicit approaches aim to maximize documents' coverage of different subtopics of the given query as possible. Santos et al. [27] leveraged probabilistic methods to measure documents' coverage of various aspects and proposed xQuAD. Dang and Croft [9] proposed PM2 that exploits the topic's popularity to adjust the proportion of different subtopics covered by the documents. Afterwards, these methods were enhanced by introducing the term information [10] and the hierarchical structure of the subtopics [15]. Recently, many explicit supervised approaches are proposed for search result diversification. For instance, DSSA [17] measured the document's subtopic coverage with RNNs and the attention mechanism. Then Qin et al. [25] leveraged the Transformer [32] to model the global relationship of all the documents and subtopics, which avoids the greedy selection process. Liu et al. [19] adapted the GAN framework to alleviate the insufficiency of training samples. Different from explicit methods, we use entities and their relations to model the document's diversity rather than subtopic coverage.

Other Methods. Apart from the two categories mentioned above, there are also several diversification approaches [44, 48, 49] designed for different purposes. For example, to avoid the greedy selection strategy, several approaches are proposed to compute the approximate optimal ranking results [11, 37]. MDP-DIV [37] leveraged the Markov decision process to select the documents imitating users' browsing behavior. Based on the MDP-DIV, M^{2}Div [11] exploited Monte Carlo tree search to explore the possible rankings during the MDP procedure. Xu et al. [41] proposed a pairwise policy gradient strategy to compare two document lists within the same query and achieved good performance. Different from these methods, KEDIV is an implicit approach that leverages knowledge to model the document's relationship in search result diversification.

2.2 Knowledge Base for IR

Early researches [5, 12, 13, 26, 31] have explored the application of knowledge base (*e.g.*, Wikipedia) in the information retrieval. Recently, knowledge continues to make breakthroughs in IR [16, 20, 22, 38, 39] and NLP tasks [2, 23, 24, 29, 30, 47].

It turns out that the abundant multidimensional information brought by knowledge is useful to capture the underlying relationship of text. For instance, Xiong et al. [40] exploited entity to improve the matching accuracy in academic search. Xiong et al. [39] introduced knowledge graphs to neural ranking models and significantly improved the generalization ability. Xiong et al. [38] also combined the representation of word and entity to improve the ranking model with an attention mechanism. Moreover, knowledge is also a reliable resource that helps to understand the query and documents. For example, Jiang et al. [16] generated queries' various facets based on the knowledge base, which demonstrates that knowledge could be the complement explanation to the ambiguous queries. Lu et al. [22] leveraged knowledge to capture the relationship between the user history and current query in personalized search. Different from these methods, we focus on search result diversification and leverage knowledge to model the relationship of the documents.

3 DIVERSIFICATION FRAMEWORK OF KEDIV

Search result diversification aims to reduce redundancy at the top of the ranking list. However, some underlying query-document and document-document relations are difficult for the model to learn only from the text content.

In this paper, we propose to leverage the knowledge base to enhance the modeling of document's relationship in search result diversification. Concretely, we build a query-specific relation graph to present the complicated relationship of the query, the documents, and the entities contained in them. Then, we apply a graph convolutional network to aggregate information of the relation graph, and the document representations will be updated by their entity nodes. Furthermore, we dynamically adjust the node weights of the graph based on the document selection.

3.1 Problem Formulation

Given a query q and its retrieved document set $\mathcal{D}$, a search result diversification model is expected to provide the re-ranking document list $\mathcal{R}$ that considers document diversity apart from relevance. Given that enumerating all the possible permutations of the candidate documents for the models are unacceptable, most existing approaches follow the greedy selection strategy: iteratively selecting the most novel (and relevant) document d^* from the candidate document set C and adding it to the selected ranking list $\mathcal{S}$. Specifically, $C = \mathcal{D}$, and $\mathcal{S} = \emptyset$ at the initial state. Therefore, the diversified ranking list $\mathcal{R}$ can be generated based on the diversified scoring function $f(q, d_i, \mathcal{S})$ at each step. From the entity perspective, the function f can be described as $f(q, d_i, \mathcal{E}_{d_i}, \mathcal{E}_s)$, where $\mathcal{E}_{d_i}$ is the entity set covered by the document d_i and $\mathcal{E}_s$ is the entity set covered by the selected document sequence $\mathcal{S}$.

3.2 Architecture of KEDIV

The overall structure of our approach KEDIV is shown in Figure 2. To consider the complicated relationship of the query q, document set $\mathcal{D}$, and their entity set $\mathcal{E}$, we leverage four types of their relationship to construct a *relation graph* $\mathcal{G}_e$. A graph neural network is further applied on the relation graph to produce diversity features $\mathbf{H}_i$ for each candidate document d_i. The final ranking score $f(q, d_i, \mathcal{E}_{d_i}, \mathcal{E}_s)$ at each step consists of relevance part $S^{\mathrm{rel}}(q, d_i)$ and diversity part $S^{\mathrm{div}}(q, d_i, \mathcal{E}_{d_i}, \mathcal{E}_s)$ with a weighted parameter λ:

$$f(q, d_i, \mathcal{E}_{d_i}, \mathcal{E}_s) = (1 - \lambda) S^{\mathrm{rel}}(q, d_i) + \lambda S^{\mathrm{div}}(q, d_i, \mathcal{E}_{d_i}, \mathcal{E}_s). \quad (1)$$

The diversity score function $S^{\mathrm{div}}(d_i, \mathcal{E}_{d_i}, \mathcal{E}_s)$ is our focus in this paper, which considers the document diversity from an entity view.[1] In another word, KEDIV not only uses the document representation for diversification but also considers the entities contained in the candidate documents, selected documents, and query. For the relevance part, KEDIV adopts the same relevance features $\mathbf{R}_i$ as the previous work [17, 19, 25, 28]. The relevance score $S^{\mathrm{rel}}(d_i)$ is then calculated from the relevance feature $\mathbf{R}_i$ with an MLP layer:

$$S^{\mathrm{rel}}(d_i) = \mathrm{MLP}(\mathbf{R}_i). \quad (2)$$

The diversity score $S^{\mathrm{div}}(d_i, \mathcal{E}_{d_i}, \mathcal{E}_s)$ is derived from diversity feature $\mathbf{H}_i$ with another MLP layer. In this paper, we build a relation graph to present the relationship of entities contained in the query and documents, then exploit the graph structure to update the representation of the entities and the documents. With the updated representations, we generate the diversity feature $\mathbf{H}_i$ from the relation graph as follows:

$$S^{\mathrm{div}}(d_i, \mathcal{E}_{d_i}, \mathcal{E}_s) = \mathrm{MLP}(\mathbf{H}_i), \quad \mathbf{H}_i = \mathcal{F}(d_i, \mathcal{E}_{d_i}, \mathcal{E}_s, \mathcal{G}_e), \quad (3)$$

where $\mathcal{G}_e$ is the corresponding relation graph for the query q. Note that at each time step t, the diversity feature $\mathbf{H}_i$ of the document d_i is dynamically generated according to the entity set $\mathcal{E}_s$ that has been covered by the selected documents. Note that the notation t is omitted for brevity. The function $\mathcal{F}$ describes how KEDIV produces the diversity features when given the relation graph $\mathcal{G}_e$, the entity set $\mathcal{E}_s$ covered by selected document sequence $\mathcal{S}$.

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

(1) **Relation Graph (Section 3.3).** We build a relation graph $\mathcal{G}_e$ for each query q. The relation graph $\mathcal{G}_e$ could be divided into two parts: the query part and the document part. In the query part, query terms and entities extracted from the query are presented as nodes in the relation graph. As for the document part, the entities that frequently appear in most documents of the document set $\mathcal{D}$ are used. The relation graph is designed to represent the relationship of query terms, entities, and documents. For example, query term nodes are connected to the document entity nodes if they appear close in the documents (*i.e.*, the distance of their positions

[1] We omit the query q in the latter equations for notation convenience.

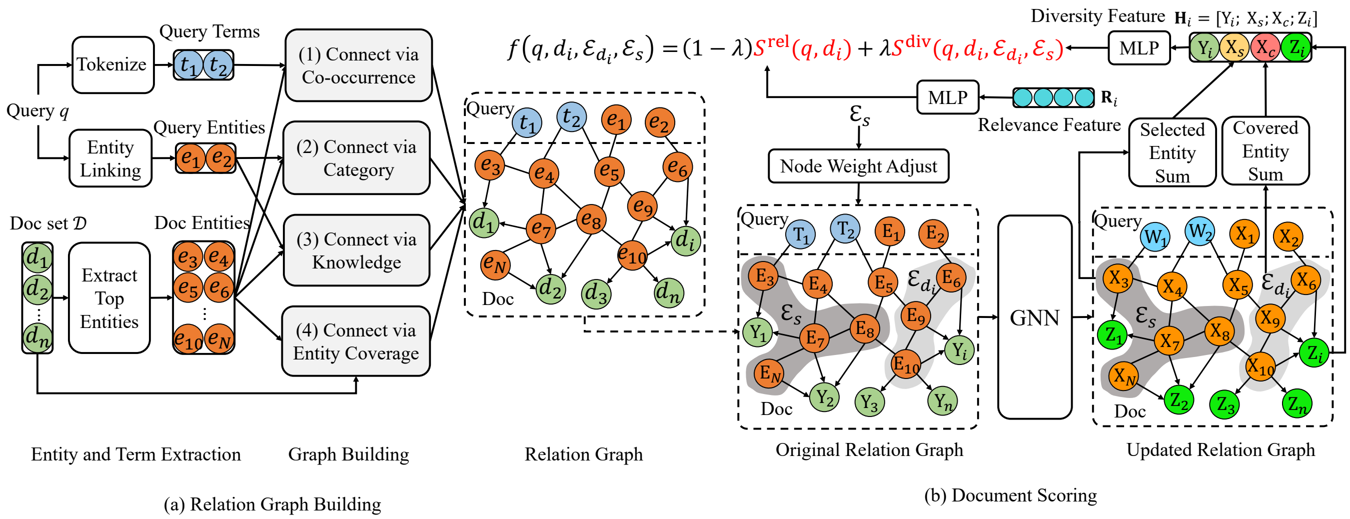

Figure 2: Overview of KEDIV. (a) illustrates the relation graph building process. There are query term nodes, entity nodes and document nodes on the relation graph. (b) demonstrates the document scoring process. Supposing d_1 and d_2 are selected documents at the current step, node weights of the relation graph are adjusted based on the entity set $\mathcal{E}_s$ covered by the selected document sequence $\mathcal{S}$. The color changes of the nodes reflects the representation update via GNN.

is less than a predefined threshold), while document nodes are connected by entity nodes that are contained in them with directional edges. Besides, entity nodes are connected if they are related in the knowledge base.

(2) **Entity-based Diversity Features (Section 3.4).** We generate the diversity feature $\mathbf{H}_i$ of document d_i based on the given relation graph $\mathcal{G}_e$. Supposing the initial entity representations $\mathrm{E} = [\mathrm{E}_1, \cdots, \mathrm{E}_N]$ and document representations $\mathrm{Y} = [\mathrm{Y}_1, \cdots, \mathrm{Y}_n]$, their representations updated via graph neural network are $\mathbf{X} = [\mathrm{X}_1, \cdots, \mathrm{X}_N]$ and $\mathrm{Z} = [\mathrm{Z}_1, \cdots, \mathrm{Z}_n]$, respectively. For diversity features, we consider the entity features X_c of the entity set $\mathcal{E}_{d_i}$, the entity features X_s of the selected entity set $\mathcal{E}_s$, the initial and updated document representation Y_i and Z_i of the candidate document d_i. The diversity feature $\mathrm{H}_i = [\mathrm{Y}_i; \mathrm{X}_s; \mathrm{X}_c; \mathrm{Z}_i]$ is the concatenation of these features.

3.3 Relation Graph

Modeling the relation between documents and queries is the foundation of search result diversification. In this paper, we leverage the relationship of entities from documents and queries to model diversity. Since the knowledge base is usually organized as a knowledge graph, we present the relationship of entities with graph structure naturally. To present a global view of the relation and get the query-specific entity representations, we build a *relation graph* $\mathcal{G}_e$ to present the relation of all the document $d_i \in \mathcal{D}$ and query q.

3.3.1 Graph Definition. Given a query q, document set $\mathcal{D}$, and entity set $\mathcal{E}_{\mathcal{D}}$ covered by the document set $\mathcal{D}$, we build a query-specific relation graph $\mathcal{G}_e = (N(\mathcal{G}_e), E(\mathcal{G}_e))$ mainly based on the entities and their relation. In the relation graph $\mathcal{G}_e$, query terms, entities and documents are presented as nodes while their relation is converted to the edges. Specifically, $E(\mathcal{G}_e)$ is the edges set of the relation graph, while $N(\mathcal{G}_e) = \{t_1, \cdots, t_m, e_1, \cdots, e_N, d_1, \cdots, d_n\}$ is the nodes set of the relation graph. $\{t_1, \cdots, t_m\}$ stand for the query term nodes, $\{e_1, \cdots, e_N\}$ represent the entity nodes from the query and documents, and $\{d_1, \cdots, d_n\}$ are the document nodes. Hence, the total node number $S = |N(\mathcal{G}_e)|$ of the relation graph is $m + N + n$. The procedure of building the relation graph $\mathcal{G}_e$ is shown in Figure 2 (a).

3.3.2 Relation Types. Instead of directly using the preliminary defined relationship from the knowledge graph, we pay attention to the relationship that is related to the diversification task and comprehensively consider the essential relationship among the document set $\mathcal{D}$.

Formally, given m query terms, N frequent entities and n documents for each query q, we can derive the query-specific adjacent matrix $\mathbf{A} \in \mathbb{R}^{S \times S}$ of the relation graph $\mathcal{G}_e$ ($S = m + N + n$) based on the four relations:

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E(\mathcal{G}_e); \\ 0, & \text{else.} \end{cases} \tag{4}$$

where v_i and v_j are the i-th node and j-th node of $\mathcal{G}_e$, respectively. The element in row i and column j of $\mathbf{A}$ is $\mathbf{A}_{i,j}$, which reflects the relation of v_i and v_j. Note that (v_i, v_j) denotes the directional edge between node v_i and node v_j ($v_j \rightarrow v_i$). For convenience, we use t_i, e_j and d_k to denote the i-th query term node, j-th entity node and k-th document node, respectively.

We consider four relations to construct graph $\mathcal{G}_e$ as follows:

(1) **Entity-term co-occurrence relation.** We assume entities that often appear near the query terms in documents have a closer relationship to the query. Specifically, we can obtain the token sequence s_i of document d_i with the knowledge base annotation, say $s_i = [w_1, e_1, w_2, \cdots, w_M]$ as the sequence of entities and words. If the query term t_j, $j \in [1, \cdots, m]$ appears together with the entity e_k and their distance is less than K tokens, we connect the

(bidirectional) edge between the query term node t_j and the entity node e_k on the relation graph, namely $(t_j, e_k) \in E(\mathcal{G}_e)$ and $(e_k, t_j) \in E(\mathcal{G}_e)$.

(2) **Entity-category relation.** The entities defined in the knowledge base often have lots of attributes (such as category and locations). In search result diversification, we find the category attribute can help distinguish ambiguous entities. For the example shown in Figure 1, it is straightforward to make out the real meaning of different "titan" in the documents with the entity category annotation. Moreover, the category of entities could be used to infer related topics of the document. Therefore, we connect the entity nodes that belong to the same category on the relation graph $\mathcal{G}_e$ to present the underlying relationship of these entities. Specifically, for entity e_i and entity e_j, $(e_i, e_j) \in E(\mathcal{G}_e)$ and $(e_j, e_i) \in E(\mathcal{G}_e)$ if e_i and e_j belong to the same category.

(3) **Entity relations in the knowledge base.** The original relation defined in the knowledge base reflects the relationship of two entities in human knowledge, which is a complement for modeling the relationship between entities and documents. Besides, in the search result diversification task, due to the limitation of the training corpus, it is challenging to directly model such relationships of entities from the document and query. Therefore, we connect the entity nodes that have relations on the relation graph to offer KEDIV a global view. Formally, for entity e_i and entity e_j, $(e_i, e_j) \in E(\mathcal{G}_e)$ and $(e_j, e_i) \in E(\mathcal{G}_e)$ if e_i and e_j have pre-defined relations in the knowledge base.

(4) **Document-entity relation.** To fully leverage the information contained in the entity nodes to enhance the document's representations, we use a directional edge to connect the entity nodes and the document nodes that mention them. In this way, the entity nodes can be used to update the representation of the documents that contain the same entities, but the entity nodes themselves will not be affected by the document nodes. For example, if the entity e_j occurs in the document d_i, then $(d_i, e_j) \in E(\mathcal{G}_e)$ $(e_j \rightarrow d_i)$.

3.3.3 *Entity Selection and Weighting.*

In our preliminary experiments, we find that the number of entities varies in different documents and the entities play different roles. To avoid the noise introduced by infrequent entities, we hypothesize that the crucial entities are frequently mentioned in different documents. So, we extract the top-N frequent entities from the document set $\mathcal{D}$. Formally, we compute the entity weight $w(e_i)$ as $w(e_i) = \max(N_{e_i}, 1)$, where N_{e_i} is the total number of documents in $\mathcal{D}$ that mention entity e_i. By default, the weights of the query term nodes, query entity nodes, and document nodes are set as 1.

Given that an important task of search result diversification is to measure the novelty of candidate documents. However, the novelty of a document is often changed considering the selected document sequence. Since we leverage entities to evaluate the document's diversity, it is necessary to adjust the weights of the entities according to the entity set $\mathcal{E}_s$ covered by the selected document sequence $\mathcal{S}$. Concretely, we lower the weights of the entity nodes on the relation graph $\mathcal{G}_e$. Supposing the entity e_i is covered by M_{e_i} documents in the selected document sequence $\mathcal{S}$, then the weight of the entity e_i is adjusted as $w(e_i|\mathcal{S}) = w(e_i)/(M_{e_i} + 1)$. Particularly, the initial weight of the entity e_i is $w(e_i) = w(e_i|\emptyset)$.

3.4 Entity-based Diversified Scoring

As mentioned in Section 3.2, we obtain the document's final ranking score based on diversity features and relevance features. For we aim to evaluate the document's diversity at the entity level, we extract both the entity representations and document representations from the relation graph to generate the diversity features. The updates of the node representations on the relation graph and the generation of the diversity feature $\mathbf{H}$ will be illustrated as follows.

Query-specific Entity Representation. Based on the query-specific relation graph, we can obtain the entity representations that absorbs the information from their neighboring nodes. Formally, given the relation graph $\mathcal{G}_e$, the initial representation of all query term nodes $\mathbf{T} = [\mathbf{T}_1, \cdots, \mathbf{T}_m]$, the representations of all entity nodes $\mathbf{E} = [\mathbf{E}_1, \cdots \mathbf{E}_N]$, and the representations of the document nodes $\mathbf{Y} = [\mathbf{Y}_1, \cdots, \mathbf{Y}_n]$, we leverage a graph neural network to update the node representations based on graph structure of the relation graph, and get the query-specific representations of query term nodes $\mathbf{W} = [\mathbf{W}_1, \cdots, \mathbf{W}_m]$, entity nodes $\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_N]$, and document nodes $\mathbf{Z} = [\mathbf{Z}_1, \cdots, \mathbf{Z}_n]$.

We use the graph convolutional network (GCN) [18] to generate the representations of the entity and document nodes on the relation graph in this paper, which can also be implemented using other GNNs. What we hope to demonstrate is that the information contained in the relation graph is helpful to measure the novelty of the documents. Updated with GCN, the nodes that have direct connections in the relation graph $\mathcal{G}_e$ will have similar representations, while the nodes without edges will have more different representations. Different from treating all entity nodes equally, we apply the node weights from Section 3.3.3 to the aggregation procedure of the graph neural networks. The nodes with large node weights will contain more information from themselves, while the nodes with fewer weights will be influenced more by their neighbors. Concretely, supposing $\mathbf{A}$ is the corresponding adjacent matrix of the relation graph $\mathcal{G}_e$, $\mathbf{W}_g = \mathrm{diag}\{w(t_1), \cdots, w(t_m), w(e_1), \cdots, w(e_N), w(d_1), \cdots, w(d_n)\}$ is the weight of each node, $\mathbf{U}^{(0)} = [\mathbf{T}_1, \cdots, \mathbf{T}_m, \mathbf{E}_1, \cdots, \mathbf{E}_N, \mathbf{Y}_1, \cdots, \mathbf{Y}_n]$ is the initial representation of the nodes on the relation graph, the node representation updates via GCN are calculated as follows:

$$\mathbf{U}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{U}^{(l)}\mathbf{W}^{(l)}), \tag{5}$$

where $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{W}_g + \mathbf{I}_N$ is generated from the adjacency matrix $\mathbf{A}$ of the relation graph in Equation (4), node weights $\mathbf{W}_g$, and the identity matrix $\mathbf{I}_N$. $\tilde{\mathbf{D}}_{i,i} = \sum_j \mathbf{A}_{i,j}$ stands for the degree of the i-th node. $l \in [0, L)$ is the layer number of the graph neural network. $\mathbf{U}^{(l)} \in \mathbb{R}^{S \times D}$ is the node features matrix where $S = m + N + n$ and D is the node feature dimension. $\sigma(\cdot)$ is an activation function, *e.g.*, $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$. $\mathbf{W}^{(l)}$ is a learnable matrix for l-th layer. We set $L = 2$ in this paper according to the experimental results. As shown in Figure 2 (b), the output of the graph convolutional network is $\mathbf{U}^{(L)} = [\mathbf{W}_1, \cdots, \mathbf{W}_m, \mathbf{X}_1, \cdots, \mathbf{X}_N, \mathbf{Z}_1, \cdots, \mathbf{Z}_n]$.

Relevance and Diversity Features. As introduced in Equation (1), the final ranking score of candidate document d_i is derived from relevance features and diversity features. We use the same relevance features as the previous work [17, 19, 25, 28]. The relevance score $S^{\mathrm{rel}}(d_i)$ of d_i is obtained based on the traditional ad-hoc features $\mathbf{R}_i$, such as PageRank, TF-IDF, BM25.

The diversity feature is our focus in this paper. To evaluate the diversity of the documents from both entity and document views, diversity features $\mathbf{H}_i$ of the document d_i are generated from the entity nodes and document nodes on the relation graph $\mathcal{G}_e$. The diversity features generation function $\mathcal{F}$ of Equation (3) can be described as follows:

$$\mathbf{H}_i = [\mathbf{Y}_i; \mathbf{X}_s; \mathbf{X}_c; \mathbf{Z}_i], \tag{6}$$

where $[;]$ means concatenation operation, $\mathbf{H}_i$ is the concatenation of initial document representation $\mathbf{Y}_i$, entity representation $\mathbf{X}_s$ derived from the selected entity set $\mathcal{E}_s$, entity representation $\mathbf{X}_c$ of the entity set $\mathcal{E}_{d_i}$, and the updated document representation $\mathbf{Z}_i$ of the document d_i.

$\mathbf{Y}_i$: The representation of the document node d_i on the relation graph $\mathcal{G}_e$. Apart from the entity features extracted from the relation graph, we use the fundamental representation of the document as a part of the diversity features.

$\mathbf{X}_s$: The representations of the selected entity set $\mathcal{E}_s$. $\mathbf{X}_s$ could be viewed as the representation of the entity coverage status at the current step, which is important in search result diversification. Concretely, $\mathbf{X}_s = \sum_{e_j \in \mathcal{E}_s} \mathbf{X}_j$ is the sum of the selected entity representations.

$\mathbf{X}_c$: The representations of the document d_i's entity set $\mathcal{E}_{d_i}$. Formally, $\mathbf{X}_c = \sum_{e_j \in \mathcal{E}_{d_i}} \mathbf{X}_j$ is the sum of the entity representations covered by document d_i. $\mathbf{X}_c$ is an essential part of document d_i's diversity features, which offers an entity view of document's novelty.

$\mathbf{Z}_i$: The representation of the document node d_i on the relation graph. Based on the relation graph $\mathcal{G}_e$, the document's representation will be updated via the entity nodes that belong to the document. Hence, the information contained in the entity nodes will enhance the representation of the documents.

3.5 Training and Optimization

Our model KEDIV could be trained end to end in the search result diversification. For each query $q \in \mathcal{Q}$, with the input of the corresponding candidate document set $\mathcal{D}_q$ and relation graph $\mathcal{G}_e$, the diversity ranking $\mathcal{R}_q$ is produced by KEDIV:

$$\mathcal{R}_q = \mathrm{KEDIV}(q, \mathcal{D}_q, \mathcal{G}_e), \quad f = \arg\min \sum_{q \in \mathcal{Q}} \sum_{o \in O_q} \mathcal{L}(\mathcal{R}_q, Y_o),$$

where O_q is the training sample set of query q, Y_o is a ideal diversified ranking of document set $\mathcal{D}$ with respect to the query q, and $\mathcal{L}$ is the list-pairwise loss function proposed by Jiang et al. [17]. List-pairwise loss is a context-aware loss for training greedy selection models. At each selection step t, given the selected document sequence $\mathcal{S}$ ($|\mathcal{S}| = t-1$), the training samples could be described as $(\mathcal{S}, d^+, d^-, w)$, where $d^+, d^- \in C$, C is the candidate document set. The training sample $(\mathcal{S}, d^+, d^-, w)$ represents that document d^+ is more diverse than the document d^- with the weight of w considering the selected sequence $\mathcal{S}$, where $w = |M([\mathcal{S}, d^+]) - M([\mathcal{S}, d^-])|$, $M(\cdot)$ is the metric function (*e.g.*, α-nDCG [7]), $[\mathcal{S}, d]$ is the current selected sequence $\mathcal{S}$ appended by the document d. Based on the training samples above, the loss could be calculated as follows:

$$\mathcal{L}_{\text{list-pairwise}} = -w_o \log \left(\frac{1}{1 + e^{-(s_o^+ - s_o^-)}} \right), \tag{7}$$

where w_o is the weight of each sample, ($o \in O_q$). s_o^+ and s_o^- are the ranking scores of the document d_o^+ and d_o^- output by the model.

4 EXPERIMENTS

4.1 Dataset and Evaluation

We conduct our experiments on the ClueWeb dataset [3] that has been commonly adopted by most existing work [9, 17, 19, 25, 27, 28, 43, 50]. It contains the corpus from TREC Web Tracks 2009 to 2012, which has approximate 50 million documents. There are 200 human-labeled topics that need to be diversified in the ClueWeb dataset totally. Since the topic #95 and topic #100 lack document diversity judgments, the left 198 queries are adopted to conduct five-fold cross-validation. The subtopic number of each topic varies from three to eight. The initial document rankings are provided by the Lemur Service.[2]. The search result diversification is considered on the top 50 relevant web pages from Lemur. The diversity evaluation metrics are calculated on the top 20 results of the diversified rankings outputted by the models. The knowledge base used in this work is Freebase [1]. The query entities are annotated via CMNS [14] and document entities are obtained from FACC.[3]

The metrics we adopt to evaluate our model and baselines are the widely used diversified ranking metrics, such as α-nDCG [7], ERR-IA [6], NRBP [8], and S-rec [46].

4.2 Baseline Methods

The baseline methods can be categorized into four types.

(1) **Non-diversified Approaches.** Since the diversified methods should consider both relevance and diversity, it is necessary to compare the performance of ad-hoc retrieval approaches with diversified ones when demonstrating the effects of diversification. Lemur denotes the ranking results from the Indri search engine, while ListMLE [34] is a supervised method that only considers document relevance.

(2) **Explicit Approaches.** We also compare our approach with classical heuristic explicit methods, such as xQuAD [27] and PM2 [9]. Some variants of these approaches are also proposed. For example, HxQuAD and HPM2 [15] adopt a hierarchical structure to model diversity, while TxQuAD, TPM2 [10] focus on term-level diversification without grouping. As for supervised methods, Jiang et al. [17] proposed a subtopic attention framework DSSA with a list-pairwise loss to train supervised diversified models.

(3) **Implicit Approaches.** KEDIV is an implicit method that does not depend on the subtopic information. We also compare KEDIV with supervised implicit methods including R-LTR [50], PAMM [35], NTN [36], DALETOR [43], and Graph4DIV [28]. Graph4DIV exploits graph to model the relation of documents and query during the document sequence selection. For DALETOR, we follow the parameters in the paper [43] and use α-nDCG loss to tune the model based on the same initial document rankings with Graph4DIV.

(4) **Ensemble Approaches.** We compare KEDIV with the ensemble methods that exploit both implicit and explicit features. DESA [25] is a non-greedy framework that leverages an encoder to fuse the features from subtopics and documents, while DVGAN [19]

[2]Lemur service: http://boston.lti.cs.cmu.edu/Services/clueweb09_batch/
[3]FACC1: http://lemurproject.org/clueweb09/FACC1/

Table 1: Overall performances. The symbol ⋆ stands for significant improvements obtained by KEDIV compared with Graph4DIV in two-tailed t-test with p-value< 0.05.

Category	Method	α-nDCG	ERR-IA	NRBP	S-rec
Ad-hoc	Lemur	.369	.271	.232	.621
	ListMLE	.387	.287	.249	.619
Explicit	xQuAD	.413	.317	.284	.622
	TxQuAD	.410	.308	.272	.634
	HxQuAD	.421	.326	.294	.629
	PM2	.411	.306	.267	.643
	TPM2	.399	.291	.250	.639
	HPM2	.420	.317	.279	.645
	DSSA	.456	.356	.326	.649
Implicit	R-LTR	.403	.303	.267	.631
	PAMM	.411	.309	.271	.643
	R-LTR-NTN	.415	.312	.275	.644
	PAMM-NTN	.417	.311	.272	.648
	DALETOR	.397	.305	.271	.607
	Graph4DIV	.468	.370	.338	.666
	KEDIV (Ours)	**.485⋆**	**.390⋆**	**.362⋆**	**.671**
Ensemble	DESA	.464	.363	.332	.653
	DVGAN	.465	.367	.334	-

uses a generator and a discriminator to learn subtopic signals and document similarity signals with an adversarial framework.

4.3 Implementation Details

For the relation graph $\mathcal{G}_e$, the numbers of the query nodes, entity nodes, and document nodes are 10, 50, and 50. We adopt Glove from NLTK [33] for the word embedding and transE [2] for the entity embedding. The dimensions of original entities and words are both 50. After the concatenation, the input dimension of word and entity is 100. The initial document representation is doc2vec with the dimension of 100. The layer number of the graph convolutional networks is tuned from one to three. The optimizer in our experiments is AdamW [21] with a learning rate of 9e-4. We tune the learning rate from 1e-6 to 1e-3 and select the models according to α-nDCG@20 with five-fold cross-validation. The parameter λ in Equation (1) is set as 0.5 based on the experiment.

4.4 Overall Performance

The overall performance of our approach KEDIV and other baseline models are shown in Table 1. KEDIV achieves better results compared with these methods in terms of the diversity evaluation metrics α-nDCG, ERR-IA, NRBP, and S-rec.

(1) KEDIV outperforms all implicit methods. Specifically, the graph-based methods, such as Graph4DIV and KEDIV, can achieve better performance than the traditional approaches, which demonstrates the advantages of leveraging graph structure to model the complicated query-document relationship in the search result diversification. Compared with Graph4DIV, KEDIV models document diversity through entities and their relationship, which improves a lot in terms of α-nDCG (3.63%), ERR-IA (5.41%), and NRBP (7.10%). This indicates that using knowledge information is helpful to model underlying document relationships that are hard for the model to learn automatically from the corpus. Moreover, the relationship provided by the knowledge base is more intuitive and explainable than the features learned from deep neural networks. Besides, Graph4DIV depends on an extra document relation classifier to explicitly judge documents similarity, while the relation graph of KEDIV is constructed based on the well-defined entity relations of the knowledge base. It also implies that knowledge-enhanced methods can achieve better performance even without pre-trained models. The improvement gained by KEDIV demonstrates the possibility of exploiting knowledge to figure out the potential relationship between the query and the documents in search result diversification. Additionally, KEDIV could be trained end-to-end while Graph4DIV is a pipeline method.

(2) KEDIV outperforms all explicit methods, including the state-of-the-art explicit method DSSA. The better performance achieved by DSSA and KEDIV demonstrates the capability of supervised method compared with unsupervised ones (*i.e.*, xQuAD, PM2). Compared with the explicit supervised method DSSA that leverages subtopic features from commercial search engines (*e.g.*, Google suggestions), KEDIV could still acquire large improvement, which reflects the effectiveness of the knowledge base in search result diversification. The possible reason is that the well-defined relationship brought by the entities is suitable to model the relationship of the documents. On the other hand, the improvement gained by KEDIV also demonstrates the need for knowledge-enhanced methods in search result diversification, especially for small datasets like ClueWeb09.

(3) KEDIV outperforms ensemble methods, *i.e.*, DESA and DVGAN. DESA combines implicit and explicit features and applies an encoder-decoder framework to produce the ranking result directly, while DVGAN leverages both explicit and implicit diversity features from the generator and discriminator. Without utilizing explicit subtopics, KEDIV can achieve better performance than these ensemble models, which shows the advantages of leveraging knowledge to measure the underlying relationship between query and documents. Besides, the diversity features extracted from the relation graph could be further exploited by these ensemble methods to offer an entity view of the document's diversity.

4.5 Ablation Study

To further investigate the performance of KEDIV with different settings, we apply an ablation study and different settings to it to figure out four questions: (1) What is the effect of each component in diversity features $\mathbf{H}$? (2) What is the effect of node weights in KEDIV? (3) How does KEDIV work under different graph settings? (4) What is the effect of each type of the relationship on the relation graph?

(1) **Ablation of diversity features.** As introduced in Equation (6), we extract the entity representations and document representations from the relation graph $\mathcal{G}_e$ to generate the diversity features $\mathbf{H}_i$ of document d_i. To investigate the effect of each representations, we remove the entity and document representations $\mathbf{X}_s, \mathbf{X}_c, \mathbf{Y}_i$ and $\mathbf{Z}_i$ from $\mathbf{H}_i$ and the results are shown in Table 2. The removal of any component of diversity features $\mathbf{H}_i$ leads to the decline in terms of all the metrics, which demonstrates the

Table 2: Performance of KEDIV with different settings.

	α-nDCG	ERR-IA	NRBP	S-rec
KEDIV	**.485**	**.390**	**.362**	**.671**
w/o $\mathbf{Y}_i$	.478	.385	.358	.660
w/o $\mathbf{X}_s$	.469	.373	.342	.658
w/o $\mathbf{X}_c$	.476	.379	.349	.669
w/o $\mathbf{Z}_i$	.479	.381	.351	.670
w/o Node Weights Adjust	.474	.379	.350	.662
w/o Node Weights	.462	.363	.330	.662
w/ GIN	.477	.384	.355	.661
w/ 1-layer-GCN.	.472	.378	.348	.660
w/ 3-layer-GCN.	.478	.379	.349	.671
w/ undirected edges	.475	.380	.351	.662

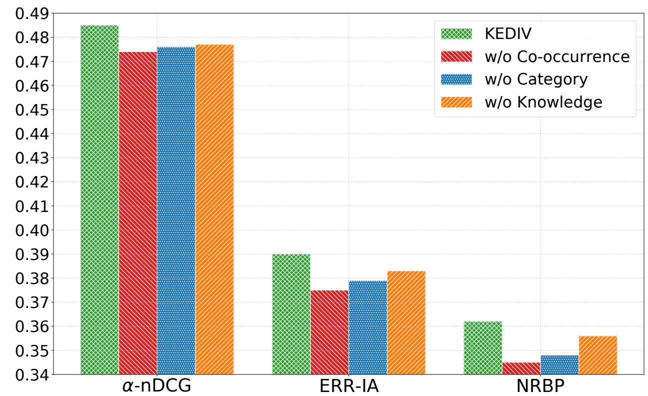

Figure 3: Effects of the different relations.

effectiveness of these features. Among all the representations, the performance of KEDIV without $\mathbf{X}_s$ declines the most. Different from the others, $\mathbf{X}_s$ is the entity representation reflecting the selection status during the greedy procedure. $\mathbf{X}_s$ presents an entity view of the selected document sequence $\mathcal{S}$, which is the foundation of the search result diversification. Since $\mathbf{X}_s$ contains the information from the entity set $\mathcal{E}_s$, KEDIV can sense the entities that have been mentioned by the selected documents, and hence, pay more attention to the uncovered entities and documents. Besides, the entity representation $\mathbf{X}_c$ of the candidate document d_i also plays an essential part. The potential reason is that leveraging well-defined entities and their relations helps to figure out the underlying relations of the documents and queries, which is a complement to the traditional approaches.

(2) **Effects of the node weight.** We leverage node weights on the relation graph to measure the importance of different entities. Considering that the novelty of entities and documents are changed during the document selection procedure, we lower the weights of the entities that have been covered by the selected documents based on the entity set $\mathcal{E}_s$ covered by the selected document sequence $\mathcal{S}$. Given that the document representations are dynamically updated via entity nodes, we can obtain the context-aware document representations at each step. As shown in Table 2, the performance of KEDIV without node weights adjustment and node weights drops a lot in terms of all the metrics. Compared with KEDIV without node weights adjustment, KEDIV without node weights performs worse, which indicates the need of evaluating the novelty of entities in search result diversification. Consistent with the target of search result diversification, the node weight adjustment strategy punishes the redundancy at the entity level and makes the relation graph dynamically present the situation of the current step, which helps KEDIV focus more on the uncovered entities.

(3) **Different graph settings.** As shown in Figure 2 (b), the node representations of the relation graph can be updated via different graph neural networks. According to the experimental results, we leverage GCN in KEDIV. The results of KEDIV with different GNNs (*e.g.*, GIN [42]) and layers are shown in Table 2. The performance of GIN is not as good as GCN, which may imply that GCN is more suitable to aggregate the information of entity nodes on the relation graph. Besides, KEDIV with a 2-layer-GCN gets higher scores considering the general metrics compared with 1-layer-GCN and 3-layer-GCN. For the layer number of the graph neural networks influence the information aggregation on the relation graph, it is not enough to use only 1-layer-GCN, while 3-layer-GCN also leads to the decline of the metrics. Because we leverage directed edges to connect the entity nodes and document nodes, we also demonstrate the performance of KEDIV with all undirected edges (denoted as w/ undirected edges) on the relation graph. Since the document representations will influence the representations of the entity node with undirected edges, the selected document nodes on the relation graph may bring the noise to the entity nodes, which will further influence other candidate document nodes via entity nodes.

(4) **Effects of the relations.** As introduced in Section 3.3.2, we leverage four types of relations to build the relation graph. To demonstrate the effect of each relation type, we record the metrics of KEDIV without these relations. Because the document-entity relation is the fundamental relationship to connect the entity nodes and the document nodes on the relation graph, we remove the other three relations from the relation graph one by one. The results of KEDIV without these three relations are shown in Figure 3. Compared with KEDIV, the removal of any relationship leads to the descent of all the metrics. It validates the effectiveness of all the relations leveraged to build the relation graph. The performance of KEDIV declines the most without the co-occurrence relation. Considering that the entities that often appear near the query terms may contain the information that answers the potential query intents, it is helpful and meaningful to capture this position information of the query terms and document entities. This demonstrates the importance of modeling query and document relationships. Credit to the explicit relations brought by the entities, KEDIV can discover the underlying relationship of different documents from both entity view and text view.

5 CONCLUSIONS

In this paper, we propose a knowledge enhanced search result diversification approach KEDIV, which leverages the entities and their relationship to help model the document's diversity. To aggregate multiple dimensional information brought by the knowledge base, we build a query-specific relation graph to present the complicated relationship of the query, the entities, and the documents.

Furthermore, the node weights adjust strategy is adopted to obtain the dynamic relation graph that reflects the current selection situation. Then we leverage a graph neural network to update the representations of all the nodes on the relation graph. With the graph structure, the document representations are enhanced by the entities that belong to them. As for the diversified scoring, representations of the entity nodes and the document nodes are used to generate the diversity features. The experimental results demonstrate that our knowledge-enhanced model KEDIV is both effective and intuitive. In the future, we plan to leverage knowledge to mine the underlying query intents from the query and fuse the knowledge graph with subtopics.

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, China Unicom Innovation Ecological Cooperation Plan, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China NO. 22XNKJ34, and the Outstanding Innovative Talents Cultivation Funded Programs 2021 of Renmin University of China.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proc. of SIGMOD.*

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proc. of NeurIPS.*

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

[4] Jaime G. Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proc. of SIGIR.*

[5] David Carmel, Haggai Roitman, and Naama Zwerdling. 2009. Enhancing Cluster Labeling Using Wikipedia. In *Proc. of SIGIR.*

[6] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proc. of CIKM.*

[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 *Proc. of SIGIR.*

[8] Charles L. A. Clarke, Maheedhar Kolla, and Olga Vechtomova. 2009. An Effectiveness Measure for Ambiguous and Underspecified Queries. In *Proc. of ICTIR.*

[9] Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proc. of SIGIR.*

[10] Van Dang and W. Bruce Croft. 2013. Term level search result diversification. In *Proc. of SIGIR.*

[11] Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2018. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. In *Proc. of SIGIR.*

[12] Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proc. of IJCAI.*

[13] Sreenivas Gollapudi and Aneesh Sharma. 2009. An Axiomatic Approach for Result Diversification. In *Proc. of WWW.*

[14] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. 2015. Entity Linking in Queries: Tasks and Evaluation. In *Proc. of ICTIR.*

[15] Sha Hu, Zhicheng Dou, Xiao-Jie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search Result Diversification Based on Hierarchical Intents. In *Proc. of CIKM.*

[16] Zhengbao Jiang, Zhicheng Dou, and Ji-Rong Wen. 2017. Generating Query Facets Using Knowledge Bases. *IEEE Trans. Knowl. Data Eng.* (2017).

[17] 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 *Proc. of SIGIR.*

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. of ICLR.*

[19] 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 *Proc. of SIGIR.*

[20] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proc. of ACL.*

[21] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proc. of ICLR.*

[22] Shuqi Lu, Zhicheng Dou, Chenyan Xiong, Xiaojie Wang, and Ji-Rong Wen. 2020. Knowledge Enhanced Personalized Search.

[23] Rada Mihalcea and Andras Csomai. 2007. Wikify! Linking Documents to Encyclopedic Knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management.*

[24] Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-Rich Word Sense Disambiguation Rivaling Supervised Systems. In *Proc. of ACL.*

[25] Xubo Qin, Zhicheng Dou, and Ji-Rong Wen. 2020. Diversifying Search Results using Self-Attention Network. In *Proc. of CIKM.*

[26] Celina Santamaría, Julio Gonzalo, and Javier Artiles. 2010. Wikipedia as Sense Inventory to Improve Diversity in Web Search Results. In *Proc. of ACL.*

[27] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proc. of WWW.*

[28] Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. 2021. Modeling Intent Graph for Search Result Diversification. In *Proc. of SIGIR.*

[29] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized Language and Knowledge Embedding. In *Proc. of COLING.*

[30] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *CoRR* (2019).

[31] Zareen Saba Syed, Tim Finin, and Anupam Joshi. 2008. Wikipedia as an Ontology for Describing Documents. In *Proc. of ICWSM.*

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proc. of NeurIPS.*

[33] Wiebke Wagner. 2010. Steven Bird, Ewan Klein and Edward Loper: Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit - O'Reilly Media, Beijing, 2009, ISBN 978-0-596-51649-9. *Lang. Resour. Evaluation* (2010).

[34] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proc. of ICML.*

[35] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. In *Proc. of SIGIR.*

[36] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2016. Modeling Document Novelty with Neural Tensor Network for Search Result Diversification. In *Proc. of SIGIR.*

[37] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2017. Adapting Markov Decision Process for Search Result Diversification. In *Proc. of SIGIR.*

[38] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-Entity Duet Representations for Document Ranking. In *Proc. of SIGIR.*

[39] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding.

[40] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding. In *Proc. of WWW.*

[41] Jun Xu, Zeng Wei, Long Xia, Yanyan Lan, Dawei Yin, Xueqi Cheng, and Ji-Rong Wen. 2020. *Reinforcement Learning to Rank with Pairwise Policy Gradient.*

[42] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *Proc. of ICLR.*

[43] Le Yan, Zhen Qin, Rama Kumar Pasumarthi, Xuanhui Wang, and Michael Bendersky. 2021. *Diversification-Aware Learning to Rank Using Distributed Representation.*

[44] Jun Yu, Sunil Mohan, Duangmanee Putthividhya, and Weng-Keen Wong. 2014. Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Proc. of WSDM.*

[45] Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural SVMs. In *Proc. of ICML.*

[46] ChengXiang Zhai, William W. Cohen, and John D. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proc. of SIGIR.*

[47] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proc. of ACL.*

[48] Jianghong Zhou, Eugene Agichtein, and Surya Kallumadi. 2020. *Diversifying Multi-Aspect Search Results Using Simpson's Diversity Index.*

[49] Jianghong Zhou, Eugene Agichtein, and Surya Kallumadi. 2020. *Diversifying Multi-Aspect Search Results Using Simpson's Diversity Index.*

[50] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *Proc. of SIGIR.*