IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 3, 2021, accepted August 22, 2021, date of publication August 24, 2021, date of current version September 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107732

Diversity-Aware Entity Exploration on Knowledge Graph

LIANG ZHENG, SHUO LIU, ZHUOFEI SONG, AND FANGTONG DOU
School of Information Management, Shanghai Lixin University of Accounting and Finance, Shanghai 201620, China
Corresponding author: Liang Zheng (liangzheng@lixin.edu.cn)

This work was supported by Shanghai Young College Teachers Training Foundation of China under Grant ZZLX21044.

ABSTRACT Knowledge graphs are graph-structured knowledge bases containing abundant entities and relations among the entities. Entity exploration can help users understand the overall structure of knowledge graphs, as well as find the entities of interest in an exploratory manner. Being different from typical entity-centric search whose goal is to retrieve the most related entities for a user's specific need, the related entities that present diverse aspects are preferred for user's ambiguous needs in entity exploration. In this paper, we propose a novel diversity-aware entity exploration approach based on random walk model, which naturally mimics human conceptual exploration by surfing a class association graph. This model leverages the diversity, representativeness and relatedness of entity classes to rank these classes in a unified way. Furthermore, for each top ranked class, the associated entities are ranked by combining their diversity and popularity. We compare our algorithm with four baseline algorithms and the experimental results indicate that it outperforms baselines. Furthermore, we conduct a task-based user study to evaluate our approach and the experimental results show that our work provides effective support for entity exploration.

INDEX TERMS Knowledge graph, diversified entity exploration, random walk model.

I. INTRODUCTION

In recent years, many knowledge graphs (KGs) have been created and reused to facilitate the real-world applications such as Web search and business intelligence. These KGs (e.g., DBpedia [1], YAGO [2], and Freebase [3]) contain abundant entities and relations among the entities. When exploiting them, users need to first explore the data to investigate whether there is useful information. Entity exploration provides an intuitive way to understand the overall structure of KGs, as well as retrieve useful information in an exploratory manner. It closely reflects information needs of end users in the real world [4].

Entity exploration is related to the related entity finding and recommendation. In information retrieval (IR), the related entity finding (REF) task [5] is to find the most related entities, and has specific constraints that the type of the target entity and the type of relation to the target entity are both given. E.g., for a source entity (''Michael Schumacher''), a relation (''His teammates while he was racing in Formula 1'') and a target type (''people''), the target entities (e.g., ''Eddie Irvine'' and ''Felipe Massa'') are returned. As to related entity recommendation, it has been defined as recommending the entities related to the entity appearing in a Web search query [6]. For instance, given an entity query (e.g., Steven Spielberg), the most related entities are recommended in search engine result pages. As an important entity facet displayed, traditional search engines (e.g., Google) leverage entity classes (types) to label and categorize the related entities. The tailored information of these entities is organized and displayed via their classes (e.g., the related person and movie of Steven Spielberg).

As mentioned above, these approaches could help users search and explore useful information. However, the above efforts have the following limitations.

First, typical search engines and REF task are expected to satisfy those users who have a specific need in mind. In contrast, entity exploration has an unclear information need, and is always used to conduct learning and investigative tasks [7]–[9]. In the process of entity exploration, human cognitive structures are constructed step by step, especially for layman users (i.e., novices in the domain). For instance, when a user is browsing an entity (''Steven Spielberg''), the concept (''Film director'') may be built in user's mind. Then, the user tends to understand and select the related concepts of interest (e.g., related actor) for further exploration. The exploration

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan Abid.

processes can be regarded as random walk in a concept graph. Hence, we propose a novel approach based on random walk model to facilitate entity exploration. Specifically, our model naturally mimics human conceptual exploration by surfing a class (concept) association graph.

Second, typical recommendation approaches adopt various techniques (e.g, structural similarity based, meta-path based, and co-occurrence based) to recommend ''similar'' related entities (i.e., the retrieval of too homogeneous results). The results that are relevant only to a single meaning may leave the user unsatisfied (e.g., if the user's intent corresponded to another meaning). Thus, in entity exploration, presenting more diverse entities of different classes may be preferred, especially for users having ambiguous needs [10]. For instance, querying a major search engine about ''Steven Spielberg'' returns the related ''Movie'' and ''Person''. Presenting more different and representative aspects (e.g., related actor, company, and award) may be beneficial for users' knowledge understanding and expansion.

Third, although entity classes can categorize related entities into meaningful groups, the large number of entities associated with the same class often make it difficult for users to efficiently explore. For instance, in typical search engine result pages (e.g., Google), ''Steven Spielberg'' has ''45+ more'' related ''Movie'' and ''15+ more'' related ''Person''. To reduce users' cognitive load further, we combine two dimensions (i.e., diversity and popularity) to rank those entities associated with the same class.

Our contributions can be summarized as follows:

- We propose a novel diversity-aware approach based on random walk model to facilitate entity exploration. It naturally mimics human conceptual exploration by surfing a class association graph.
- We leverage the diversity, representativeness and relatedness of entity classes to rank them in a unified way. We adopt the difference between two classes to measure the diversity of them. We measure the relatedness between two classes based on their distance. We introduce the representativeness by combining three metrics (i.e., frequency, conciseness, and specificness). Furthermore, for each top ranked entity class, the associated entities are ranked by combining their diversity and popularity.
- We compare our algorithm with four baseline algorithms and the experimental results indicate that it outperforms baselines. We also conduct a task-based user study to evaluate our approach and the experimental results show that our approach provides useful support for entity exploration.

The remainder of this paper is organized as follows. Related works are discussed in Section II. The problem is stated in Section III, as well as an overview of our approach. We discuss our approach in detail in Section IV. The evaluations are given in Section V. We conclude our approach in Section VI.

II. RELATED WORKS

In general, our study is related to three fields in the literature, namely, related entity finding and recommendation, knowledge graph exploration, and diversification problem.

A. RELATED ENTITY FINDING AND RECOMMENDATION

Related entity finding (REF) task [5] is to retrieve and rank related entities given a structured XML query specifying an input entity, the type of related entities and the relation between the input and related entities in the context of a given document collection. Bron *et al.* [11] propose a framework for addressing REF task and perform a detailed analysis of four core components: co-occurrence models, type filtering, context modeling and homepage finding. Fang *et al.* [12] propose unified probabilistic models to formalize the process of REF. The proposed methods incorporate entity relevance, type estimation, type matching, entity prior and entity co-occurrence into a holistic probabilistic framework.

Related entity recommendation is often applied in the major Web search engines. Spark [6] at Yahoo! extracts several features from a variety of data sources and uses a machine learning model to produce a recommendation of entities for a Web search query, where neither the relation type nor the type of the target entity are specified. Microsoft [13] has also developed a similar system that performs personalized entity recommendation by analyzing user click logs and entity pane logs. Torzec *et al.* [14] propose a layered-graph based embedding approach for entity recommendations based on Wikipedia in the yahoo! knowledge graph. The entity candidates are generated based on topological and semantic similarities.

In fact, these works have inspired the initial idea of our work, but there are some differences. First, in general, the related entity finding and recommendation are expected to satisfy those users who have specific information needs. In contrast, entity exploration has an unclear information need. Second, these approaches are agnostic to entity types and they recommend similar related entities (i.e., the retrieval of too homogeneous results). The results that are relevant only to a single meaning may leave the user unsatisfied (e.g., if the user's intent corresponded to another meaning). In contrast, our approach leverages the diverse and representative entity classes to categorize the related entities, and selects diverse entities of different entity classes (types). Third, these commercial Web search engines utilize their own usage data, such as knowledge graph, query terms, search sessions, and user click logs. Our approach is user-independent and we resorts to data sources publicly available on the Web.

B. KNOWLEDGE GRAPH EXPLORATION

Most existing studies have focused on improving the efficiency of exploration over knowledge graph. These approaches include visual exploration and identifying key entities in KG.

Visualisation provides an important way for exploration on KG. It leverages the human perception and analytical abilities to offer exploration trajectories. There are many visualisation tools available [15]–[18]. However, the layman users (i.e., novices in the domain) may struggle to grasp the complex knowledge graph presented in the visualization. Many path-based approaches are introduced in order to reduce users' exploration burden further.

SPHINX [16] is a system for metapath-based entity exploration, which allows users to define different views based on both automatically selected and user-defined meta-paths. SEED [17] is designed to support entity-oriented exploration in large-scale KGs, and retrieve similar entities of some seed entities by semantic patterns mining. Han *et al.* [18] present an entity-oriented exploratory search prototype system (called PivotE) that is able to support search and explore KGs in an exploratory search manner. The system applies a path-based ranking method to recommend similar entities and their relevant information as exploration pointers.

Identifying key entities can help users understand the knowledge better and judge the suitability of an entity quickly in KG. Troullinou *et al.* [19] exploit the structure and the semantic relationships of a data graph to identify the most important entities using the relevance and in/out degree centrality of an entity. Lee *et al.* [20] adapt personalized PageRank algorithm to rank entities according to a given query that is represented as a set of entities in the graph.

The above exploration systems provide useful supports for entity exploration over KGs. However, the above efforts ignore that: in essence, the exploration process is building users' cognitive structures, especially for layman users. Our approach stresses the importance of entity class, which is a basic concept of entity and used to label entities. Furthermore, we mimic human conceptual exploration based on random walk model in a concept (class) association graph.

C. DIVERSIFICATION PROBLEM

The importance of diversity has been recognized in various contexts, such as diversifying search results [21]–[25], entity exploration [26], summarization [27], and recommendation system [28].

Zhou *et al.* [21] introduce a search result diversification algorithm by adopting the Simpson's Diversity Index from biology. The diversity index is characterized by two aspects: richness and evenness. Richness quantifies the number of different classes of elements in a set, while evenness considers the uniformity of the distribution of these classes. Arnaout *et al.* [22] diversify search results of RDF knowledge graphs by using a maximal marginal relevance algorithm. It can trade off the relevance of results in the top-k set and their diversity (as measured by their average distance). Rafie *et al.* [23] model the search result diversity as an expectation maximization problem and estimate the model parameters. Kennedy and Naaman [24] leverage the community-contributed collections of rich media on the web to automatically generate representative and diverse views of the world's landmarks. DivRank [25] balances the centrality and the diversity of the vertices for ranking them based on a vertex-reinforced random walk in an information network. Metaexp [26] introduces a set of diverse meta-paths to assist the user during the exploration of large knowledge graphs. The diversity of k meta-paths is measured by combining coverage and diversity. Coverage is the number of meta-paths that are covered by the chosen k meta-paths. Diversity quantifies the number of unique node- or edge-labels present in the k meta-paths. FACES [27] highlights the importance of diversified entity summaries by combining three dimensions: diversity, uniqueness, and popularity. Ziegler *et al.* [28] propose an approach towards balancing top-N recommendation lists according to the topic diversification.

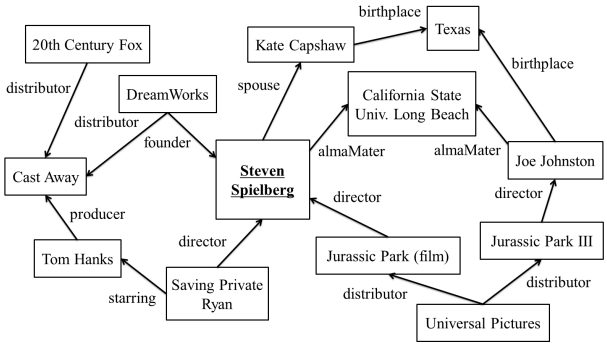


FIGURE 1. An excerpt of knowledge graph with Steven Spielberg in DBpedia.

These approaches define the diversification in various ways, namely in terms of content (i.e., similarity), novelty, coverage, or hybrid [29]. Some of them formalize the diversification as an optimization problem, and handle it with non-optimized heuristics based on greedy vertex selection. Inspired by these works, our approach highlights the importance of diversified entity classes by combining the representativeness, difference and relatedness in a unified way.

III. PROBLEM STATEMENT

In this section, we introduce basic concepts of the problem we investigate, and then explain the flow of our approach.

Definition 1 (Knowledge Graph):[1]: A knowledge graph is a finite directed labeled graph denoted by $G = (V_G, A_G, L, l_G)$, where:

- V_G is a finite set of entities as vertices;
- A_G is a finite set of edges, and each edge connects an entity to another entity in V_G;
- L is the set of edge types;
- $l_G : A_G \rightarrow L$ is a function that labels each edge $a \in A_G$ with a type $l(a) \in L$.

Fig.1 shows an excerpt of knowledge graph with Steven Spielberg.

[1]In this study, we focus on the relations between instance-level entities and hence we ignore the literal edges. The knowledge graph can be regarded as a kind of entity-relation graph.

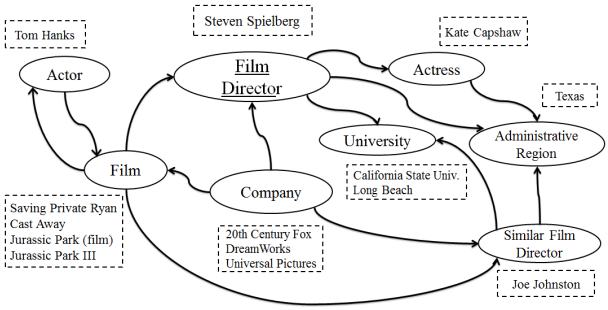


FIGURE 2. An example of class association graph.

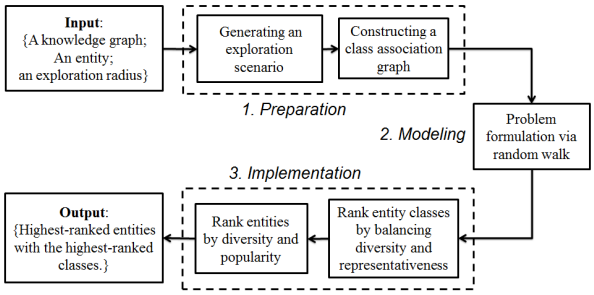


FIGURE 3. The work flow of our approach.

Definition 2 (Exploration Trajectory [30]): An exploration trajectory T in a knowledge graph $G = (V_G, A_G, L, l_G)$ is defined as a sequence of entities and edge labels in the form of $T =< e_1, l_1, e_2, \ldots, e_n, l_n, e_{n+1} >$, where:

- $e_i \in V_G, i = 1, \ldots, n+1$;
- $l_i \in L, i = 1, \ldots, n$;
- e_1 and e_{n+1} are the first and last entities of the trajectory T, respectively;
- n is the length of the exploration trajectory T.

The distance between two entities, denoted by $dist(\cdot, \cdot)$, is the shortest length of the exploration trajectories between them, or 0 if no such trajectory exists.

Note that the trajectory T is similar to the notion of path in an undirected graph. For instance, <Steven Spielberg, director, Saving Private Ryan, starring, Tom Hanks> is an exploration trajectory in Fig.1, and its length is two.

Definition 3 (Exploration Scenario): Given an entity e and a nonnegative integer r, an exploration scenario $S = (V_S, A_S)$ of entity e is a subgraph of knowledge graph G, where:

- $V_S \in V_G$;
- $A_S \in A_G$;
- $\forall e' \in V_S, dist(e, e') \leq r$ or $dist(e', e) \leq r$.

Also, r is called exploration radius of scenario S.

Fig.1 shows an exploration scenario of Steven Spielberg in DBpedia, and the exploration radius of this scenario is three.

Definition 4 (Class Association Graph [31]): Let $E(c)$ be the set of entities of class c, and let C be the set of classes subject to that every $c \in C$ is with $E(c) \neq \varnothing$. Then a class association graph is an edge-weighted directed graph denoted by $CAG = (C, A, W_A)$, where:

- C is a finite set of classes as vertices;
- A is the edge set, and an directed edge (c_1, c_2) between $c_1, c_2 \in C$, called an association, is in A iff a path exists from $e_1 \in E(c_1)$ to $e_2 \in E(c_2)$;
- $W_A : A \rightarrow N$ is a weighting function that maps edges to natural numbers.

Fig.2 shows a class association graph derived from exploration scenario of Steven Spielberg in DBpedia. There is an association from "Film Director" to "Administrative Region" because a path exists from Steven Spielberg, through Kate Capshaw, to Texas.

Note that the related entities have the same classes (types) with current browsing entity. In order to distinguish them, we introduce "similar" classes. For instance, Joe Johnston is also a "Film Director". We add a new class "similar Film Director" in Fig.2.

The process of our approach is illustrated in Fig.3. Firstly, given a KG, a browsing entity, and an exploration radius as input, an exploration scenario is automatically generated. Based on the exploration scenario, we construct a class association graph. Then, we propose a variant of the random walk model that mimics conceptual exploration by surfing the class association graph. Furthermore, we present an implementation of this model, which picks the diverse and representative entity classes based on semantic and information theory concepts. Also, within each entity class, we rank the related entities by their diversity and popularity. Finally, we extract a set of entities by selecting the highest-ranked entities, which are associated with the highest-ranked entity classes.

IV. APPROACH

We propose a novel approach DivRW to facilitate entity exploration, which not only naturally mimics human conceptual exploration by surfing a class association graph, but also takes into account the diversity of exploration in a unified way. It is motivated from random walk model, which is used in many real world tasks, such as Web page ranking [32], entity ranking [25], and entity summarization [33].

In this section, we first discuss how we construct a class association graph. Then, we formalize the problems via random walk modeling. Finally, we present an implementation of our model.

A. CLASS ASSOCIATION GRAPH CONSTRUCTION

The construction process includes two steps: generating an exploration scenario $S = (V_S, A_S)$ and constructing a class association graph $CAG = (C, A, W_A)$ based on S.

1) GENERATING AN EXPLORATION SCENARIO

We adopt a straightforward way to generate an exploration scenario by the use of breath-first search. We choose the browsing entity e to be the root. Then we add edges incident with all vertices adjacent to e. These vertices are at level 1 in

the scenario S. Next, add the edges that connect these vertices at level 1 to adjacent vertices not already in S. This produces the vertices at level 2. Follow the same procedure until the number of level is equal to the exploration radius r. In this way, we fetch the vertices V_S and a part of edges A_S.

Yet, this generates a spanning tree and may result in the loss of edges between the vertices. So, we add the procedure of the edge completion. For each vertex, we add the edges that connect this vertex to the rest of vertices, and these edges are not already in the scenario S.

Algorithm 1 depicts the process of generating an exploration scenario S of entity e with exploration radius r in a knowledge graph KG. The vertices V_S and a part of edge A_S of S are generated in line 1-13. The procedure of the edge completion is in line 14-17.

Algorithm 1 Generating an Exploration Scenario

Input: KG: a knowledge graph; e_0: an entity; r: an nonnegative integer
Output: S: an exploration scenario

1: Initialize $S = (V_S, A_S)$, $V_S \leftarrow \varnothing, A_S \leftarrow \varnothing$;
2: $r' = 0$; $E \leftarrow \varnothing$; $E \leftarrow E \cup \{e_0\}$; $V_S \leftarrow V_S \cup \{e_0\}$;
3: **while** $r' \leq r$ **and** $E \neq \varnothing$ **do**
4: **for all** $e \in E$ **do**
5: remove e from E;
6: find each neighbor e' of e;
7: **if** $e' \notin E$ **and** $e' \notin V_S$ **then**
8: $E \leftarrow E \cup \{e'\}$; $V_S \leftarrow V_S \cup \{e'\}$;
9: $A_S \leftarrow A_S \cup \{(e, e')\}$;
10: **end if**
11: **end for**
12: $r' = r' + 1$;
13: **end while**
14: **for all** $e \in V_S$ **do**
15: find each neighbor e' of e;
16: $A_S \leftarrow A_S \cup \{(e, e')\}$;
17: **end for**
18: **return** S;

The time complexity of generation algorithm is $O(r * |V_S| * |V_S|)$, where r is the exploration radius, and $|V_S|$ is the number of vertices in the exploration scenario S.

2) CONSTRUCTING A CLASS ASSOCIATION GRAPH

Given an exploration scenario S, we can construct a class association graph $CAG = (C, A, W_A)$ of S. There are two major parts: capturing the set of classes C and the set of edges A. The weighting function W_A is discussed in the next section.

Firstly, we generate the set of classes C. Many existing knowledge graphs (e.g., DBpedia, Yago, and Freebase) are either available as Linked Open Data [34], or they can be exported as RDF datasets [35]. Given an entity e, its class set C_e can be derived from RDF triples in the form (uri, $rdf : type$, c), where uri identifies the entity e and c is an entity class. For example, <dbr: Tom Hanks> is the URI of Tom Hanks in DBpedia. A triple <dbr: Tom Hanks, $rdf : type$, dbo:Person> can be captured, and the <dbo:Person> is a class of Tom Hanks.[2] For the convenience of query later, we also construct a bidirectional index between entity and its classes.

Then, we adopt a simple way to capture the association set A. For each class $c_i \in C$, we enumerate all the candidate classes. Whether there is an association between c_i and c_j, iff a path exists from $e_i \in E(c_i)$ to $e_j \in E(c_j)$. $c_j \in C$ is the candidate class. $E(c)$ is the set of entities of class c.

Algorithm 2 describes the process of constructing a class association graph CAG derived from an exploration scenario S. The time complexity of construction algorithm is $O(|C| * |C| * |V_S|)$, where $|C|$ is the number of vertices of CAG, and $|V_S|$ is the number of vertices of S.

Algorithm 2 Constructing a Class Association Graph

Input: $S = (V_S, A_S)$: an exploration scenario
Output: CAG: a class association graph

1: Initialize CAG=(C, A, W_A), $C \leftarrow \varnothing, A \leftarrow \varnothing, W_A \leftarrow \varnothing$;
2: **for all** $e \in V_S$ **do**
3: add the class set of e to C;
4: **end for**
5: **for all** $c \in C$ **do**
6: **for all** $c' \in C$ **do**
7: **if** a path from $e \in E(c)$ to $e' \in E(c')$ in scenario S exists **then**
8: $A \leftarrow A \cup \{(c, c')\}$;
9: **end if**
10: **end for**
11: **end for**
12: **return** CAG

B. PROBLEM FORMULATION VIA RANDOM WALK MODELING

Similar to the standard random walk [32], we can model the actions of a generic surfer. At each step of the walk, the surfer can perform two kinds of atomic actions:

- Move following an association. When a surfer has just visited current class, he moves along an edge to the associated class for a more complete understanding of domain. For instance, as shown in Fig.2, the current class "Film director" has established. The surfer then can explore the related classes (e.g., "Actress" and "University").
- Jump between classes. The surfer jumps to an arbitrary class that is not linked with current class. For instance, as shown in Fig.2, the surfer can jump from "Actress" to "University."

Formally, given a class association graph CAG=(C, A, W), the surfer's actions, move (M) and jump (J), can be modeled

[2]Prefix: dbr: http://dbpedia.org/resource/;
dbo: http://dbpedia.org/ontology/;
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#.

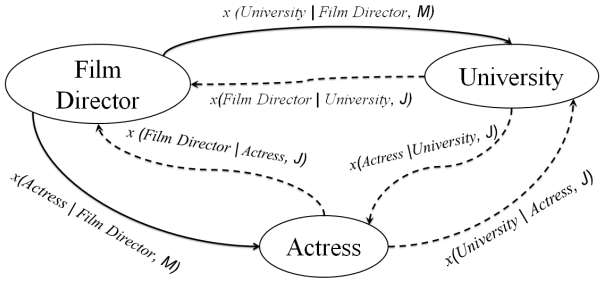


FIGURE 4. An example of two actions (move (*M*) and jump (*J*)).

by a set of conditional probabilities which depend on the current class $u \in C$:

- $x(M|u)$: the probability of moving from u, and
- $x(J|u)$: the probability of jumping from u.

There are only two kinds of action. Thus, these values must satisfy the normalization constraint $x(M|u) + x(J|u) = 1$.

These actions need to specify their targets. We model two actions between current class u and the target class v by using the following parameters:

- $x(v|u, M)$: the probability of moving from class u to class v, and
- $x(v|u, J)$: the probability of jumping from class u to class v.

These sets of probabilities must satisfy the following normalization constraints for each class $u \in C$, $\sum_{v \in C} x(v|u, M) = 1$, and $\sum_{v \in C} x(v|u, J) = 1$.

In Fig.4, nodes represent entity classes; solid lines represent move (M) actions between classes; dashed curves represent jump (J) actions between classes. Each action is associated with a nonuniform probability. For instance, $x(Actress|FilmDirector, M)$ is the probability of moving from class "Film Director" to "Actress." $x(University|Actress, J)$ is the probability of jumping from class "Actress" to "University."

The random walk is defined by a sequence of actions performed by the surfer. The probabilistic model can be used to compute the probability that the surfer is located in each class u at step t, $x_u(t)$. The probability distribution on all the classes is represented by the vector $x(t) = [x_1(t), x_2(t), \ldots, x_N(t)]'$, being N the total number of classes. By taking all the possibilities of the surfer's actions into account, the probability $x_v(t+1)$ is updated as follows:

$$x_v(t+1) = \sum_{u \in C} x_u(t) \cdot (x(v|u, M) \cdot x(M|u) + x(v|u, J) \cdot x(J|u)). \tag{1}$$

The probabilities defining the surfer model can be organized in the following $N \times N$ matrices:

- The move matrix $\mathbf{M}$ whose element is $x(v|u, M)$;
- The jump matrix $\mathbf{J}$ whose element is $x(v|u, J)$;
- A diagonal matrix Δ collecting $x(M|u)$;
- A diagonal matrix Γ collecting $x(J|u)$.

Then (1) can be rewritten as:

$$x(t+1) = x(t) \cdot (\mathbf{M} \cdot \mathbf{\Delta} + \mathbf{J} \cdot \mathbf{\Gamma}). \tag{2}$$

In previous studies [32], it has been proved: If $x(J|u) \neq 0$ and $x(v|u, J) \neq 0$, $\forall u, v \in C$, then there exists x^* such that $\lim_{t \to \infty} x(t) = x^*$ and x^* does not depend on the initial state vector $x(0)$.

Efficient computation is needed for practical applications. In general, the iterative computation of (2) is usually configured to stop after a certain number of iterations. Finally, we can rank the entity classes in the class association graph.

To implement this model, we need to give $\mathbf{M}$, $\mathbf{J}$, $\mathbf{\Delta}$, and $\mathbf{\Gamma}$, i.e., to define $x(v|u, M)$, $x(v|u, J)$, $x(M|u)$, and $x(J|u)$. We will discuss them in the next section.

C. MODEL IMPLEMENTATION

Similar to the computation of the PageRank, we assume that the surfer has a consistent probability of choosing between move action and jump action. Thus, we define $x(M|u) = d$, and $x(J|u) = 1 - d$, where $d \in [0, 1]$ is regarded as a parameter and tested in experiments.

In the process of human conceptual exploration, surfers may prefer to visit the diverse, representative, and relevant classes. Based on this assumption, we define the computation of the probability $x(v|u, M)$ and $x(v|u, J)$.

1) MOVE MATRIX $\mathbf{M}$

When moving via an association, surfer may explore the most relevant target classes associated with current class. Thus, surfer can have a deep understanding of domain. Here, we not only consider the representativeness of target classes, but also balance the relatedness and difference of the classes.

The probability of moving from current class u to the target class v is defined as follows:

$$x(v|u, M) = \tau_v \cdot (\lambda \cdot rel(u, v) + (1 - \lambda) \cdot diff(u, v)) \tag{3}$$

where:

- τ_v: the representativeness of the target class v;
- $rel(u, v)$: the relatedness between class u and class v;
- $diff(u, v)$: the difference between class u and class v.

The three metrics are described in detail below.

1) In practice, the surfer follows this intuition, i.e., the more frequent, concise and specific an entity class is, the more representative it is. Thus, the representativeness of class v can be defined based on the following weighted linear combination:

$$\tau_v = \alpha \cdot freq(v) + \beta \cdot conc(v) + \gamma \cdot spec(v) \tag{4}$$

where:

- $freq(v) = \frac{|E(v)|}{|V_s|}$ is the frequency of class v, $E(v)$ is the set of entities of class v, and V_s is the set of entities in the exploration scenario S. The more associated entities a class has, the higher the coverage of the class is;

- $conc(v) = e^{-(|label(v)|-1)}$ is the conciseness of class v. A concise class having a shorter label is more understandable. For instance, the class "AmericanFilmDirectors" is more understandable than "ProducersWhoWonTheBestPictureAcademyAward". $conc(v)$ is an exponential function, which is a decreasing function and returns the value in the range (0, 1] (for normalization purpose);
- $spec(v) = \frac{dist(\top,v)}{dist(\top,v,\bot)}$ is the specificness of class v. $dist(\top, v)$ is the length of the shortest path from $\top$ (i.e., the greatest element of class hierarchy) to v. $dist(\top, v, \bot)$ is the length of the shortest path from $\top$, through v, to $\bot$ (i.e., the least element of class hierarchy). The deeper the depth of class in hierarchy is, the more specific it is;
- $\alpha + \beta + \gamma = 1$ and $\alpha, \beta, \gamma \in [0, 1]$ indicate the weights for each metric to be tuned empirically.

2) In general, the shorter the distance between two elements is, the higher the relatedness of the two elements is. Therefore, we compute the relatedness between two classes via the distance between their associated entities.

$$rel(u, v) = rel(E(u), E(v)) = \frac{\sum_{sp \in SP} e^{-length(sp)}}{|E(u)| \cdot |E(v)|} \quad (5)$$

where:

- $E(*)$ is the set of entities of class $*$;
- SP is the set of the shortest paths from $e_u \in E(u)$ to $e_v \in E(v)$;
- $length(\cdot)$ is the length of the shortest path.

In fact, $rel(u, v)$ is an exponential function, which is a decreasing function and returns the value in the range (0, 1]. Note that we do not consider a circle (i.e., from an entity to itself).

3) For the diverse and comprehensive understanding of domain, the surfer may investigate those classes that are different from current class. We use the difference between class u and class v to measure the diversity of them.

$$diff(u, v) = 1 - \frac{E(u) \cap E(v)}{E(u) \cup E(v)} \quad (6)$$

and $E(*)$ is the set of entities of class $*$. Here, we leverage the degree of overlap of two entity sets to compute the difference of two classes.

Finally, to satisfy the probability normalization constraints in (1), we normalize the move matrix $\mathbf{M} = [m_{ij}]$ element as follows:

$$m_{ij} = \frac{x(i|j, M)}{\sum_{i \in C} x(i|j, M)} \quad (7)$$

and C is the class set in the class association graph.

2) JUMP MATRIX J

When understanding current class enough, the surfer may jump to arbitrary classes that are not linked with current class. These classes, which should be representative and diverse, can expand surfer's domain knowledge.

The probability of jumping from current class u to the target class v is defined as follows:

$$x(v|u, J) = \tau_v \cdot diff(u, v) \quad (8)$$

where:

- τ_v: the representativeness of the target class v, and
- $diff(u, v)$: the difference between class u and class v.

τ_v is defined in (4), and $diff(u, v)$ is defined in (6).

Finally, to satisfy the probability normalization constraints in (1), we normalize the jump matrix $\mathbf{J} = [J_{ij}]$ element as follows:

$$J_{ij} = \frac{x(i|j, J)}{\sum_{i \in C} x(i|j, J)} \quad (9)$$

and C is the class set in the class association graph.

3) ENTITY RANKING

The representative and diverse entity classes provide multi-granular and progressive exploration assistances. In some cases, the class may have many associated entities, so that users feel disoriented and require some understanding to pick the target entities.

To lighten users' burden further, we rank the entities associated with the same class in terms of their "goodness". We define the "goodness" of an entity e by combining two dimensions (i.e., diversity and popularity):

$$good(e) = \phi \cdot div(e) + \varphi \cdot pop(e) \quad (10)$$

where:

- $div(e) = \frac{|type(e)|}{|C|}$ is the diversity of entity e, $type(e)$ is all the classes (types) of e, and C is the class set in the class association graph. The more associated classes an entity has, the higher the diversity of the entity is;
- $pop(e) = \frac{|neighbor(e)|}{|V_s|}$ is the popularity of entity e, $neighbor(e)$ is all the neighbors of e in the exploration scenario S, and V_s is the set of entities in S. The more neighbors an entity has, the higher the popularity of the entity is;
- $\phi + \varphi = 1$ and $\phi, \varphi \in [0, 1]$ indicate the weights for each metric to be tuned empirically.

In the end, we select a set of entities by picking the highest-ranked entities, which are associated with the highest-ranked entity classes.

V. EXPERIMENTS

In this section, we compare the performance of our approach with that of four baseline algorithms. Also, we conduct a task-based user study to evaluate the effectiveness of our approach.

A. PERFORMANCE EVALUATION

In this experiment, our approach called DivRW (abbreviation for diversity based on random walk) was compared with four

TABLE 1. Statistics of experimental datasets.

	DBpedia	YAGO
Entities	14,178	11,519
Average number of neighbors per entity	223.1	105.8
Average number of classes per entity	9.8	40.2

baseline algorithms by four evaluation metrics (i.e., diversity, comprehensiveness, and goodness).

1) DATASETS

We performed experiments on two real-world datasets.

- DBpedia[3] is a central interlinking hub of the encyclopedic dataset on the Web. Specifically, the candidate entity collection was obtained from the *Cleaned object properties extracted with mappings* dataset. Classes of entities were obtained from the *DBpedia instance types* dataset. Class hierarchy was obtained from the *DBpedia ontology*.
- YAGO[4] is a knowledge base which is extracted from Wikipedia and other sources. The candidate entity collection was obtained from the *yagoFacts* dataset. Classes of entities were obtained from the *yagoSimpleTypes* dataset. Class hierarchy was obtained from the *yagoTaxonomy*.

We collected four "big" classes (i.e., person, place, organisation, work), whose instance number was larger than 500K. For each class, we collected those entities that have more than 20 different edges each. The detailed distribution of the two datasets is listed in Table 1. We can observe that the entities have a larger number of neighbors in DBpedia, and the entities have more classes in YAGO.

2) EVALUATION BASELINES AND PARAMETER SETTINGS

Inspired by previous related works, we introduce four representative algorithms (i.e., TF-IDF, PageRank, VRRW, and Diversity-based reranking) to compare with our approach DivRW .[5]

- TF-IDF. Inspired by [27], FACES ranks features by using TF-IDF from each facet to form the diversified summary, and thus we adopt TF-IDF to rank the importance of an entity associated with a class. We take an entity as a word and a class as a document, respectively. Term frequency, $tf(e, c)$, is the frequency of entity e,

$$tf(e, c) = \frac{1}{|E(c)|} \tag{11}$$

  where $E(c)$ is the set of entities of class c, and entity e occurs in its class c just once.
  The inverse document frequency, $idf(e, C)$, is the logarithmically scaled inverse fraction of the classes that contain the entity e,

$$idf(e, C) = \log \frac{|C|}{|type(e)|} \tag{12}$$

  where $type(e)$ is all the classes (types) of e, and C is the class set in the class association graph.
  TF-IDF is used to measure the importance of an entity e, which is associated with a class $c \in C$.

$$tfidf(e, c, C) = tf(e, c) \cdot idf(e, C). \tag{13}$$

- PageRank. Inspired by [32], PageRank is used to measure the importance of Web pages, and thus we use PageRank algorithm to measure the importance of entities in a knowledge graph.
- VRRW. Inspired by [25], DivRank balances the centrality and the diversity of the vertices to rank them based on a vertex-reinforced random walk (VRRW), and thus we leverage the class coverage of entity to reinforce vertex (i.e., entity) based on VRRW. Let $p_t(e, e')$ be the transition probability from any entity e to any entity e' at iteration t.

$$p_t(e, e') = (1 - d) \cdot \frac{1}{N} + d \cdot \frac{p_0(e, e') \cdot \eta_t(e')}{\sum_{e_i \in V} p_0(e, e_i) \cdot \eta_t(e_i)} \tag{14}$$

  where:
  - d is a damping factor. Following [25], we set $d = 0.9$;
  - N is the number of entities in KG;
  - V is the set of entities in KG;
  - $\eta_t(e') = \frac{|type(e')|}{|C|}$ where $type(e')$ is all the classes (types) of e', and C is the class set in KG. The more classes an entity has, the more value the entity has;
  - $p_0(e, e')$ is the regular transition probability prior to any reinforcement.

  $p_0(e, e')$ is defined as follows:

$$p_0(e, e') = \begin{cases} \pi \cdot \dfrac{w(e, e')}{deg(e)}, & e \neq e'. \\ 1 - \pi, & otherwise. \end{cases} \tag{15}$$

  where:
  - $w(e, e')$ is equal to 1 for an edge (e, e') exists in KG, and 0 otherwise;
  - $deg(e)$ is the out-degree of vertex e;
  - π is a tuned parameter and we set $\pi = 0.25$ by following [25].
- Diversity-based reranking. Inspired by content-based diversity (i.e., items that are dissimilar to each other) [29], we apply a reranking method to select diverse and popular entities based on average pair-wise difference. Firstly, we rank all the entities using their popularity (defined in (10)). Then, based on these ranked entities, we adopt a heuristic method to select diverse entities. We pick the most popular entity e_1 to be the first one of

[3]https://databus.dbpedia.org/dbpedia/collections/latest-core

[4]http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/

[5]These algorithms are implemented in Python and the Python development tool is PyCharm 4.0. The codes of our project are uploaded and can be downloaded freely on https://github.com/waynezheng/DivRW

TABLE 2. Comparison of the quality of top-k entities selected by experimental algorithms.

	DBpedia									YAGO								
	Diversity			Coverage			Goodness			Diversity			Coverage			Goodness		
	k=20	k=50	k=100	k=20	k=50	k=100	k=20	k=50	k=100	k=20	k=50	k=100	k=20	k=50	k=100	k=20	k=50	k=100
TF-IDF	0.23	0.32	0.53	0.27	0.33	0.13	0.27	0.43	0.45	0.63	0.68	0.71	0.13	0.28	0.32	0.17	0.24	0.25
PageRank	0.57	0.63	0.71	0.58	0.42	0.36	0.21	0.25	0.27	0.69	0.66	0.63	0.21	0.41	0.61	0.36	0.39	0.37
VRRW	0.61	0.65	0.77	0.57	0.45	0.36	0.32	0.29	0.33	0.71	0.69	0.67	0.19	0.43	0.62	0.38	**0.41**	0.39
Reranking	0.71	0.69	0.81	**0.69**	0.75	0.84	0.38	0.48	0.57	0.74	0.73	0.72	0.27	0.42	0.56	0.24	0.26	0.38
DivRW,with t_1	**0.74**	0.78	**0.87**	0.62	0.78	**0.89**	**0.52**	0.65	**0.67**	0.74	**0.79**	**0.78**	0.37	**0.56**	**0.81**	0.21	0.31	0.42
DivRW,with t_2	0.68	0.76	0.81	0.65	0.76	0.86	0.45	0.59	0.63	0.71	0.76	0.77	0.28	0.46	0.71	0.18	0.19	0.21
DivRW,with t_3	0.67	0.73	0.76	0.56	0.67	0.81	0.32	0.55	0.61	0.69	0.72	0.73	0.29	0.49	0.66	0.21	0.23	0.22
DivRW,with t_4	0.71	0.76	0.82	0.56	0.76	0.86	0.49	0.58	0.63	**0.75**	0.76	0.78	**0.39**	0.53	0.67	0.25	0.31	0.42
DivRW,with t_5	0.73	**0.81**	0.85	0.59	**0.79**	0.89	0.51	**0.67**	0.67	0.74	0.79	0.77	0.31	0.55	0.66	**0.26**	0.36	**0.46**
DivRW,with t_6	0.69	0.73	0.79	0.53	0.65	0.85	0.42	0.55	0.61	0.72	0.75	0.75	0.31	0.54	0.69	0.22	0.25	0.34

The best quality values are highlighted.

a result queue. From the rest of ranked entities, we select an entity e_2 which is the most different from e_1, and put e_2 onto the end of queue. In this way, we can find a better diversity solution, where a new added entity is the most different from the entities in the queue. In order to capture the difference between an entity e and the k entities $Q_k = \{e_1, \ldots, e_k\}$, we define a metric based on their average pair-wise difference as follows:

$$DIFF(e, Q_k) = \frac{\sum_{1 \le i \le k} diff(e, e_i)}{k} \tag{16}$$

$$diff(e, e_i) = 1 - \frac{|type(e) \cap type(e_i)|}{|type(e) \cup type(e_i)|} \tag{17}$$

where $type(e_i)$ is the classes (types) of entity e_i.

Our approach DivRW has seven parameters (i.e., d, α, β, γ, λ, ϕ, and φ). The parameter settings are given below.

- We set $d = 0.85$, which is similar to the damping factor setting of PageRank. Various studies in PageRank have tested different damping factors, and the damping factor is generally set around 0.85.
- For the weighting parameter λ in (3), we set $\lambda = 0.5$ based on equity.
- For the parameters (α, β, γ) in (4), we empirically set $t_1 = (1, 0, 0)$, $t_2 = (0, 1, 0)$, $t_3 = (0, 0, 1)$, $t_4 = (0.33, 0.33, 0.33)$, $t_5 = (0.6, 0.3, 0.1)$, and $t_6 = (0.1, 0.3, 0.6)$.
- As to the balance parameters (ϕ, φ) in (10), we set $\phi = \varphi = 0.5$ based on equity.

More parameters settings will be experimented in future work.

For each experimental dataset, we randomly selected 100 entities. As to each selected entity, we constructed its exploration scenario satisfying the exploration radius $r = 2$, and selected its top-k (=20, 50, 100) related entities using our approach (DivRW) and four baseline algorithms (i.e., TF-IDF, PageRank, VRRW, and Diversity-based reranking). We empirically conducted 10 runs using these four algorithms and assessed the results with three evaluation metrics (i.e., diversity, comprehensiveness, and goodness).

Specially, DivRW selected the highest-ranked entities that associated with the highest-ranked entity classes. For each highest-ranked class, we selected top-5 entities at most, and these selected entities were different from each other.

3) EVALUATION METRICS

To measure the quality of top-k entities, we utilized three criteria: diversity, comprehensiveness, and goodness.

Given k entities $Q_k = \{e_1, \ldots, e_k\}$, the three criteria are defined as follows.

- Diversity measures the difference between entities.

$$diversity(Q_k) = \frac{\sum_{1 \le i \le j \le k} diff(e_i, e_j)}{k(k-1)/2} \tag{18}$$

  where $diff(e_i, e_j)$ is defined in (17).
- Comprehensiveness measures the class (type) coverage of entities.

$$coverage(Q_k) = \frac{|\bigcup_{i=1}^{k} type(e_i)|}{|C|} \tag{19}$$

  where $type(e_i)$ is the classes (types) of entity e_i, and C is the class set in the class association graph.
- Goodness refers to the average goodness of entities.

$$goodness(Q_k) = \frac{1}{k} \sum_{i=1}^{k} good(e_i) \tag{20}$$

  where $good(e_i)$ is defined in (10).

4) RESULTS

Table 2 shows the comparisons of the quality of top-k entities selected by experimental algorithms in datasets DBpedia and YAGO respectively. A comparative analysis based on these results is given below.

In most cases, our approach DivRW performed better than other algorithms. Reranking outperformed TF-IDF, PageRank, and VRRW. PageRank and VRRW had a better performance than TF-IDF. VRRW behaved slightly better than PageRank in some cases.

The scores of diversity and coverage in DBpedia were higher than those in YAGO. This was because there were plenty of fine-grained entity classes in YAGO. Although the selected entities had rich classes, there was overlap among these classes. With the increase of the number of selected entities (i.e., k = 20, 50, 100), these algorithms had a better performance.

As to the influence of the three parameters (α, β, γ) in representativeness computing, we investigated how the three

parameters influenced the scores under the same k and algorithm. Higher scores of the metrics were obtained when the frequency or conciseness had higher weight (t_1 and t_5). The metrics of frequency and conciseness provided a better fit for measuring representativeness in our experimental datasets. It indicated that an entity class which was more representative should have more associated entities and a shorter label. More datasets and parameter settings will be experimented in future work.

B. USER STUDY

We conducted a task-based user study to investigate how our approach helps user's exploration in practice. By analyzing subjects' responses to questionnaire and their behaviors during the experiment, we mainly aimed to test the following three goals.

- Verifying that entity classes label and group the entities, and thus exploiting them can effectively help human understanding in entity exploration.
- Assessing that the selected classes are diverse, but still representative and comprehensive.
- Confirming that our approach improves the efficiency of entity exploration compared with baselines.

1) EXPERIMENTAL SETTING

a: DATASET AND TASKS

We used DBpedia covering a large amount of broad-ranging entities. It allowed us to design exploration tasks that were not targeted at a particular domain. Our entity exploration tasks were derived from 120 popular search terms in the Google trends 2020.[6] From these search terms, we chose 10 queried entities. These 10 entities were found in DBpedia and had more than 20 different edges each.

For each entity, we established the knowledge exploration tasks including entity summarization and related entity finding. For instance, the exploration tasks about Tom Hanks were designed as follows: ''Suppose you will write a summary about Tom Hanks including at least three main aspects, and then find at least five most related entities.'' Finally, the 10 tasks were to be used in the user study.[7]

b: PARTICIPANT APPROACHES

To evaluate the performance of our approach, we compared three different organizations of result:

- *Entity-Only*. The selected entities were displayed as a flat list. This method was served as a baseline performance. We used PageRank to select top-20 entities directly.
- *Entity-Class*. In addition to displaying entities as a flat list, the second baseline added the entity class tags to describe the result entities. We adopted TF-IDF to pick top-20 entities with the corresponding tag (i.e., class). The implementation of TF-IDF is described in (13).
- *Entity-Class-Group*. This mode grouped and labeled the entities via their classes. The user traversed these tags in a top-down mode. Our approach DivRW selected top-20 entities that associated with the highest-ranked entity classes. For each highest-ranked class, we selected top-5 entities at most, and these selected entities were different from each other.

c: PROCEDURE

We invited 20 student subjects majoring in computer science. They were familiar with the Web, but with no knowledge of our project. The evaluation procedure was performed as follows.

- Before the evaluation session, the subjects learned how to perform the tasks through a 5 mins tutorial. In the tutorial, we asked the subjects to read a short abstract of entity first, and then carry out the task. The subject was given one task as a warmup. To avoid bias as far as possible, we used the same example to illustrate and the subjects took the same task as a warmup. In addition, there was 5 minutes for free use and questions.
- Then, the subjects used each of the given approaches arranged in random order. For each approach, the subjects were randomly assigned to perform the exploration tasks. Meanwhile, these tasks among the given approaches were different. The subjects were asked to complete all the tasks in 30 minutes. We recorded their answers and the time they spent on each task until the subject concluded the experiment with an explicit action of termination.
- Finally, with regard to each approach, the subjects responded to the post-task questionnaire on a 5-point Likert scale, and provided feedback on the quality of exploration.

d: EVALUATION METRICS

Two metrics were measured in the experiments: task time and user feedback. Task time referred to the average time used by a subject for carrying out one task based on a given system. User feedback was measured after performing the test using a post-task questionnaire. The subject had to answer four evaluation questions (appearing in a slightly abbreviated form here):

- *Comprehensive*. Does this set of results offer a comprehensive overview of the related entities (0-5 scale)?
- *Diverse*. How diverse are the content you have explored (0-5 scale)?
- *Representative*. How many entity classes are representative of the related entities (0-5 scale)?
- *Satisfying*. How satisfied are you with this set of results (0-5 scale)?

For the purpose of evaluation, the subjects viewed these questions in random order and without repetition.

Fig.5 shows the exploration tasks about Tom Hanks. There are post-task questionnaires and three results generated by three approaches (i.e., PageRank, TF-IDF, and DivRW).

[6]https://www.google.com/trends/topcharts

[7]The tasks are uploaded on https://github.com/waynezheng/DivRW

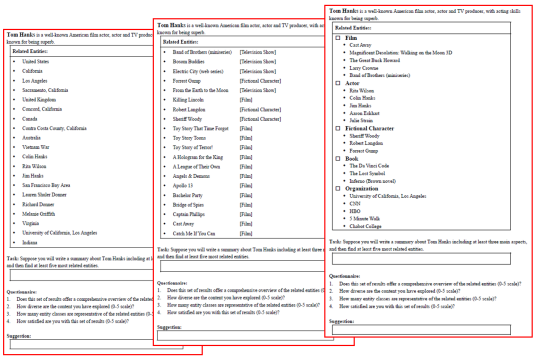


FIGURE 5. The exploration tasks about Tom Hanks including post-task questionnaires and three results generated by three approaches (i.e., PageRank, TF-IDF, and DivRW).

TABLE 3. Average time spent in exploration tasks of the three approaches.

	Entity-Only (PageRank)	Entity-Class (TF-IDF)	Entity-Class-Group (DivRW)
Avg. time in minutes	9.5	6.2	3.8

2) RESULTS

Table 3 shows the average time spent in the tasks of the three approaches respectively. Since our approach DivRW provided an overview using entity classes as tags, subjects took far less time (3.8 minutes) to complete these tasks. Compared with Entity-Only (PageRank) and Entity-Class (TF-IDF), Entity-Class-Group (DivRW) was informative and distinguishing, which remarkably helped the subjects to retrieve comprehensive information quickly.

Table 4 shows a summary of the results of post-task questionnaire. The results of each question were averaged over all users. Repeated measures ANOVA revealed that the differences in subjects' mean ratings were all statistically significant ($p < 0.1$). LSD post-hoc tests ($p < 0.05$) revealed that, according to the *comprehensive*, DivRW provided the best comprehensive overview of all the related entities due to the label of entity classes. Besides, TF-IDF had a better overview than PageRank. Also, as to the *diverse*, DivRW clearly improved the diversity of the related entities compared with TF-IDF and PageRank.

DivRW provided significant gains in *comprehensive* and *diverse*, but it yielded little difference in *representative*. The results indicated that the diverse and comprehensive entities increased the total number of related entities. These redundant entities may be considered less representative. The results of the *satisfying* showed that the mode of Entity-Class-Group was preferable to Entity-Only and Entity-Class. It provided the subjects with more helpful support for exploration tasks compared with baselines.

We summarized all the major comments of the subjects. As to Entity-Class-Group (DivRW), 16 subjects (80%) said that it provided a comprehensive overview of information and enabled users to explore domain knowledge quickly. 13 subjects (65%) said that it offered diverse results to help users to expand domain knowledge, but 4 subjects (20%) said that it may lead to redundant class tags. As to Entity-Only (PageRank) and Entity-Class (TF-IDF), 18 subjects (90%) said that the large number of entities ranked as a flat list often made it difficult for users to understand the overall domain. These comments were consistent with subjects' experience and behavior reported previously. All of these collectively supported the goals of our evaluations. Entity-Class-Group (DivRW) selected the diverse and representative entity class tags to describe the result entities. It provided a comprehensive overview of the entities, and helped users to understand and expand the domain knowledge quickly.

TABLE 4. Results of post-task questionnaire.

	Entity-Only (PageRank)	Entity-Class (TF-IDF)	Entity-Class-Group (DivRW)
Comprehensive	2.5 (0.95)	**3.8** (0.92)	**4.2** (0.85)
Diverse	3.7 (0.52)	3.9 (0.71)	**4.1** (0.63)
Representative	3.3 (1.31)	3.2 (0.82)	3.6 (0.94)
Satisfying	2.5 (0.47)	3.2 (1.03)	**3.9** (0.59)

The standard deviations are shown in brackets. Statistically significant changes ($p < 0.1$) are shown in boldface.

VI. CONCLUSION

In this paper, we have presented a novel approach based on random walk model to facilitate entity exploration over knowledge graph. In our approach, it not only naturally mimics human conceptual exploration by surfing a class association graph, but also takes into account three perspectives (i.e., diversity, relatedness, and representativeness) in a unified way. Extensive evaluations have confirmed the effectiveness of our approach. The results showed that our approach performed better than other three baselines and provided useful supports for entity exploration.

Our approach can be applied in many applications. It can be used for the entity-oriented exploratory KG exploration systems, which provides diverse entities of different types as exploration pointers. Our approach can also be used in other applications such as related entity recommendation in Web search. For instance, Google's and Yahoo's "People also search for" can recommend diverse entities for a given query entity, which help users better specify their information needs. Our approach is also useful in recommendation systems, which can be used as a complementary feature. It can provide more diverse goods of different categories to enhance user satisfaction.

There are some directions for future work. We will study "human factors" in the context of entity exploration. We can collect users' preference on class tags and entities and then leverage them to measure the class tags and entities.

REFERENCES

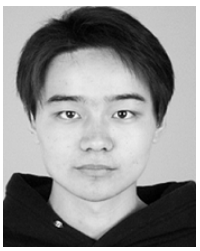
[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia–a crystallization point for the web of data," *J. Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[2] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artif. Intell.*, vol. 194, pp. 28–61, Jan. 2013.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.
[4] R. Kumar and A. Tomkins, "A characterization of online browsing behavior," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 561–570.
[5] K. Balog, P. Serdyukov, and A. P. de Vries, "Overview of the TREC 2011 entity track," in *Proc. 20th Text REtr. Conf.*, 2011, p. 11.
[6] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec, "Entity recommendations in web search," in *Proc. 12th Int. Semantic Web Conf.*, 2013, pp. 33–48.
[7] G. Marchionini, "Exploratory search: From finding to understanding," *Commun. ACM*, vol. 49, no. 4, pp. 41–46, 2006.
[8] R. W. White and R. A. Roth, "Exploratory search: Beyond the query-response paradigm," *Synth. Lectures Inf. Concepts, Retr., Services*, vol. 1, no. 1, pp. 1–98, Jan. 2009.
[9] P. Vakkari, "Exploratory searching as conceptual exploration," in *Proc. 4th Workshop Hum.-Comput. Interact. Inf. Retr.*, 2010, pp. 24–27.
[10] A. Angel and N. Koudas, "Efficient diversity-aware search," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2011, pp. 781–792.
[11] M. Bron, K. Balog, and M. De Rijke, "Ranking related entities: Components and analyses," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1079–1088.
[12] Y. Fang and L. Si, "Related entity finding by unified probabilistic models," *World Wide Web*, vol. 18, no. 3, pp. 521–543, May 2015.
[13] B. Bi, H. Ma, B.-J. Hsu, W. Chu, K. Wang, and J. Cho, "Learning to recommend related entities to search users," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, Feb. 2015, pp. 139–148.
[14] C.-C. Ni, K. Sum Liu, and N. Torzec, "Layered graph embedding for entity recommendation using Wikipedia in the Yahoo! knowledge graph," in *Proc. Companion Proc. Web Conf.*, Apr. 2020, pp. 811–818.
[15] A. S. Dadzie and E. Pietriga, "Visualisation of linked data–reprise," *Semantic Web*, vol. 8, no. 1, pp. 1–21, 2017.
[16] S. Chatzopoulos, K. Patroumpas, A. Zeakis, T. Vergoulis, and D. Skoutas, "SPHINX: A system for metapath-based entity exploration in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2913–2916, 2020.
[17] J. Chen, J. Giulio, C. Yueguo, and R. Tuukka, "SEED: Entity oriented information search and exploration," in *Proc. 22nd Int. Conf. Intell. User Interfaces Companion*, 2017, pp. 137–140.
[18] X. Han, J. Chen, J. Lu, Y. Chen, and X. Du, "PivotE: Revealing and visualizing the underlying entity structures for exploration," *Proc. VLDB Endowment*, vol. 12, no. 12, pp. 1966–1969, 2019.
[19] G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis, "Ontology understanding without tears: The summarization approach," *Semantic Web*, vol. 8, no. 6, pp. 797–815, Aug. 2017.
[20] S. Lee, S.-I. Song, M. Kahng, D. Lee, and S.-G. Lee, "Random walk based entity ranking on graph for multidimensional recommendation," in *Proc. 5th ACM Conf. Recommender Syst. (RecSys)*, 2011, pp. 93–100.
[21] J. Zhou, E. Agichtein, and S. Kallumadi, "Diversifying multi-aspect search results using Simpson's diversity index," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2345–2348.
[22] H. Arnaout and S. Elbassuoni, "Effective searching of RDF knowledge graphs," *J. Web Semantics*, vol. 48, pp. 66–84, Jan. 2018.
[23] D. Rafiei, K. Bharat, and A. Shukla, "Diversifying web search results," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 781–790.
[24] L. S. Kennedy and M. Naaman, "Generating diverse and representative image search results for landmarks," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, pp. 297–306.
[25] M. Qiaozhu, J. Guo, and R. Dragomir, "DivRank: The interplay of prestige and diversity in information networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1009–1018.
[26] F. Behrens, S. Bischoff, P. Ladenburger, J. Rückin, L. Seidel, F. Stolp, M. Vaichenker, A. Ziegler, D. Mottin, F. Aghaei, E. Müller, M. Preusse, N. Müller, and M. Hunger, "MetaExp: Interactive explanation and exploration of large knowledge graphs," in *Proc. Companion Web Conf.*, 2018, pp. 199–202.
[27] K. Gunaratna, K. Thirunarayan, and A. Sheth, "FACES: Diversity-aware entity summarization using incremental hierarchical conceptual clustering," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 116–122.
[28] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proc. 14th Int. Conf. World Wide Web (WWW)*, 2005, pp. 22–32.
[29] M. Drosou and E. Pitoura, "Search result diversification," *ACM SIGMOD Rec.*, vol. 39, no. 1, pp. 41–47, 2010.
[30] M. Al-Tawil, V. Dimitrova, and D. Thakker, "Using knowledge anchors to facilitate user exploration of data graphs," *Semantic Web*, vol. 11, no. 2, pp. 205–234, Feb. 2020.
[31] Y. Qu, G. Weiyi, C. Gong, and G. Zhiqiang, "Class association structure derived from linked objects," in *Proc. WebSci*, 2009, pp. 18–20.
[32] M. Diligenti, M. Gori, and M. Maggini, "A unified probabilistic framework for web page scoring systems," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 4–16, Jan. 2004.
[33] G. Cheng, T. Tran, and Y. Qu, "RELIN: Relatedness and informativeness-based centrality for entity summarization," in *Proc. 10th Int. Semantic Web Conf.*, 2011, pp. 114–129.
[34] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," in *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. Hershey, PA, USA: IGI Global, 2011, pp. 205–227.
[35] *W3C: Resource Description Framework (RDF)*. World Wide Web Consortium (W3C). [Online]. Available: http://www.w3.org/RDF/

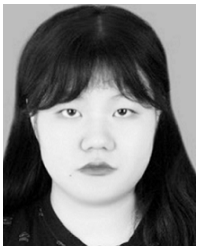
LIANG ZHENG was born in Nanyang, Henan, China, in 1980. He received the M.S. degree in computer software and theory from Chongqing University, in 2008, and the Ph.D. degree in computer science and technology from Nanjing University, in 2017.

Since 2018, he has been a Lecturer with the School of Information Management, Shanghai Lixin University of Accounting and Finance. He has authored several articles in international journals and top-level conferences, such as *Knowledge-Based Systems* (KBS), *World Wide Web Journal* (WWWJ), *Journal of Information Systems and Telecommunication* (JIST), ISWC, ESWC, and APWeb. His research interests include semantic web, knowledge graph, semantic search, and data mining.



SHUO LIU was born in Panjin, Liaoning, China, in September 2000. He is currently pursuing the B.S. degree in computer science and technology with Shanghai Lixin University of Accounting and Finance. His current research interests include artificial intelligence, data mining, and knowledge graph.

ZHUOFEI SONG was born in Baishan, Jilin, China, in December 2000. She is currently pursuing the B.S. degree in computer science and technology with Shanghai Lixin University of Accounting and Finance. Her current research interests include data analysis and knowledge graph.



FANGTONG DOU was born in Weifang, Shandong, China, in November 1999. She is currently pursuing the B.S. degree in computer science and technology with Shanghai Lixin University of Accounting and Finance. Her current research interests include data science and machine learning.