SEED: Entity Oriented Information Search and Exploration

Chen, Jun

2017

Chen , J , Jacucci , G , Chen , Y & Ruotsalo , T 2017 , ' SEED: Entity Oriented Information Search and Exploration ' , pp. 137-140 . https://doi.org/10.1145/3030024.3040978

http://hdl.handle.net/10138/308687
https://doi.org/10.1145/3030024.3040978

unspecified
acceptedVersion

SEED: Entity Oriented Information Search and Exploration

Jun Chen
School of Information, Renmin University of China, China
Key Laboratory of Data Engineering and Knowledge Engineering, MOE, China
chenjun2013@ruc.edu.cn

Yueguo Chen
School of Information, Renmin University of China, China
Key Laboratory of Data Engineering and Knowledge Engineering, MOE, China
chenyueguo@ruc.edu.cn

Giulio Jacucci
Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland
giulio.jacucci@helsinki.fi

Tuukka Ruotsalo
Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland
tuukka.ruotsalo@helsinki.fi

IUI'17 Companion, March 13–16, 2017, Limassol, Cyprus.
ACM ISBN 978-1-4503-4893-5/17/03.
http://dx.doi.org/10.1145/3030024.3040978

Abstract
Entity search and exploration can enrich search user interfaces by presenting relevant information instantly and offering relevant exploration pointers to users. Previous research has demonstrated that large Knowledge Graphs allow exploitation and recommendation of explicit links between the entities and other information to improve information access and ranking. However, less attention has been devoted to user interfaces for effectively presenting results, recommending related entities and explaining relations between entities. We introduce a system called SEED which is designed to support entity search and exploration in large Knowledge Graphs. We demonstrate SEED using a dataset of hundreds of thousands of movie related entities from the DBpedia Knowledge Graph. The system utilizes a graph embedding model for ranking entities and their relations, recommending related entities, and explaining their interrelations.

Author Keywords
Exploratory Search, Knowledge Graph, Knowledge Representation, Embedding Model

ACM Classification Keywords
H.3.3 [Information Search and Retrieval]: Miscellaneous; H.5.2 [User Interfaces]: Miscellaneous

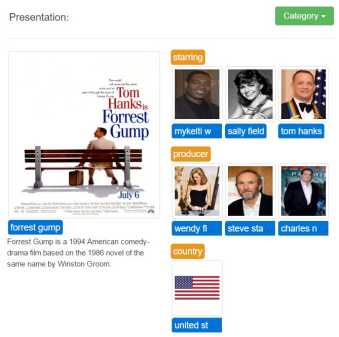

Figure 1: The entity presentation is based on ranking the informative relations and their target entities of the source entity.

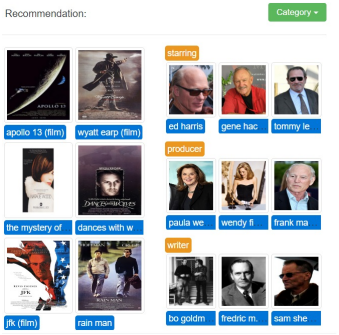

Figure 2: The entity recommendation is based on ranking similar source entities, the informative relations and their target entities of the source entity.

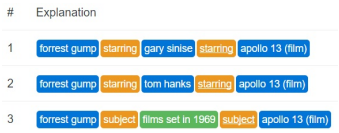

Figure 3: The explanation is based on ranking and visualizing the paths between selected entities.

Introduction

Entity oriented information exploration, as opposed to conventional document search, is becoming a key enabler for next generation search and information exploration user interfaces [5, 6]. It retrieves entities (e.g. the movie *Forrest Gump*) and presented them instantly with their relevant information (e.g. the actor *Tom Hanks*), which fulfills users' information needs. In large Knowledge Graphs (KGs), entities are linked to each other, which allows recommendation of entities to enable direct exploration points to novel and serendipitous results (e.g. the movie *Apollo 13*, which is similar to the movie *Forrest Gump*) [2].

KGs specify a rich set of relations that are meaningful to humans and foster methods for explaining and enhancing understanding of the presented information (e.g. *Apollo 13* is similar to *Forrest Gump* because *Tom Hanks* and *Gary Sinise* are starring both movies, both are American movies, and both depict historical events). In this way, entity oriented search and exploration can bridge the gap between unstructured keyword-search and structured knowledge [5].

Designing intelligent search user interfaces for entity exploration involves recommending entities for further exploration. Users can evaluate recommended entities by considering the entity information presented and the relations between entities [3].

In this paper, we describe the design and implementation of the SEED system as illustrated in Figure 4. SEED supports users to search and explore entities in large KGs and demonstrates: 1) *Entity Presentation*: which information related to the entity should be presented to users? 2) *Entity Recommendation*: which entities to recommend for user as exploration points? 3) *Relation Explanation*: how to explain why the entities are recommended? (see Figures 1,2, and 3 for examples)

The SEED system

SEED has a web-based user interface as illustrated in Figure 4, which is composed of four interface components: 1) *Query Construction area*; 2) *Entity Presentation area* ; 3) *Entity Recommendation area*; 4) *Relation Explanation area* (see Figures 4-a, 4-b, 4-c and 4-d for examples respectively).

Query Construction

In order to assist users constructing a query to retrieve relevant entities, SEED has an entity auto-completion mechanism. Auto-completion triggers a ranked dropdown list of matching entities in response to a typed prefix of an entity label as illustrated in Figure 4-a.

Entity Presentation

In order to select information from hundreds of relations attached to a source entity in response to a query, SEED ranks and presents relations and referred target entities that are relevant to users. For instance, in Figure 4 the movie entity *Forrest Gump* is retrieved (Figure 4-b2) and a set of relations (the orange color span, such as *starring*) and their target entities (the blue color span, such as *Tom Hanks*) are ranked and presented for users (Figure 4-b3). Similarly, relevant categories to which the movie *Forrest Gump* belongs to are ranked and presented (Figure 4-b1).

Entity Recommendation

In order to go beyond ranked results and provide users with exploration points to related content, SEED provides entity recommendations. The system uses a graph embedding model [4] to represent entities in a lower dimensional vector space and discovers similar entities by comparing their representations in that lower dimensional space. For example, given the source movie entity *Forrest Gump*, the system finds similar entities, such as the movie *Apollo 13* (Figure 4-c2) that can be used as the starting point for exploration.

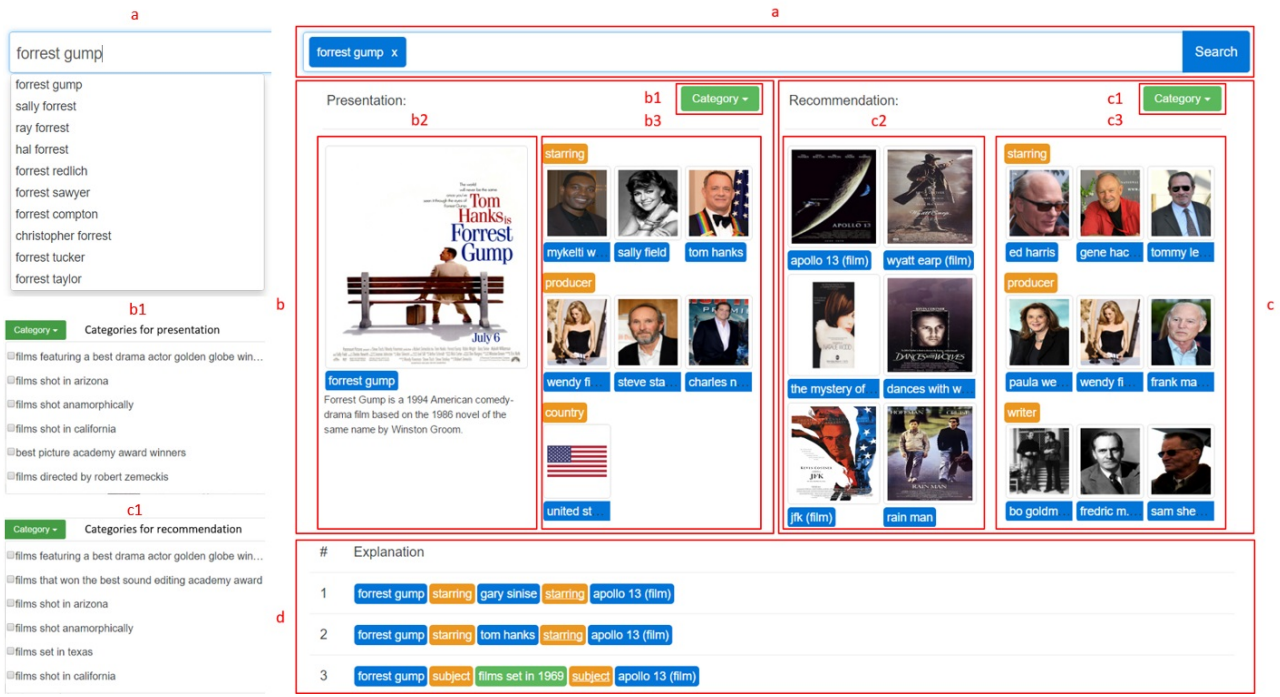

Figure 4: ci

Relation Explanation

The recommendations that result from the similarity computation are ranked, but if the mechanics and reasoning of the recommendation algorithm can be communicated to users in the right way, it can improve acceptance of the prediction and trust towards the system. In order to explain the interrelations between two entities, they can be dragged together to discover the explanation. For instance, when dragging *Forrest Gump* and *Apollo 13* together (see Figure 4-d), SEED will rank the paths with highest explanation ability between them to communicate to users why the entities were recommended.

Entity Representation, Ranking and Indexing

Entity and Relation Ranking

In order to *present* and *recommend* the most informative and relevant information to users, SEED determines the similarity of entities based on their representation in the KG.

In order to determine the similarity between entities, SEED employs the *TransE* graph embedding model [4]. It em-

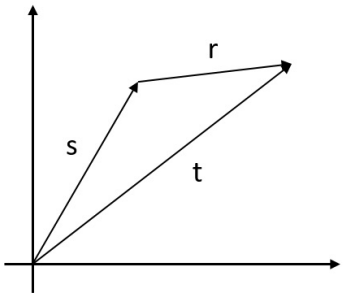

Figure 5: Graph embedding model maps entities and relations into a lower-dimensional space by using local or global connectivity patterns specified by triples of source entity (s), relation (r), and a target entity (t) to generalize the observed relation between a specific entity and all other entities.

beds entities and relations of multi-relational graph data in low-dimensional vector spaces. As a result, SEED ranks the top-k nearest source and target entities in the low-dimensional space by comparing them using simple vector space similarity [4].

Each entity in KGs can have dozens or even hundreds of relations. SEED ranks them in order to return the most informative relations to be presented to the user. We apply information gain [7] to evaluate the informativeness of each relation and relations with the highest information gain are selected. Then SEED ranks target entities for each relation. As a result, SEED presents users with most relevant relations (e.g. starring and producer for the movie *Forrest Gump*) and their target entities (e.g. *Sally Field* and *Tom Hanks* for the starring relation).

Entity and Relation Recommendation
Entity recommendation ranks the relations and entities by considering not only the source entity, but also top-k similar source entities. For example, in the case of the movie *Forrest Gump*, the system not only finds similar movies (e.g. *Apollo 13*), but also relevant actors (e.g. *Ed Harris* or *Gene Hackman*), who are not necessarily performing in *Forrest Gump*, but performing in other similar movies (e.g. *Ed Harris* is performing in *Apollo 13*).

Index and Data Storage
The index of SEED consists of movie related dataset from DBpedia 2014 including 0.33 million entities, 13 relations and 1.9 million triples. Since the efficiency of the query-response is critical for interactive operations, we apply Locality Sensitive Hashing (LSH) [1] as index to support fast retrieval of top-k results. The index is based on an in-memory storage.

Conclusions

We introduced the design and implementation of SEED, a system for entity oriented information search and exploration. The system demonstrates how KGs can be exposed for users through interfaces that allow effective *presentation, recommendation, and explanation* of entity data. Entity and relation ranking, recommendation, and explanations are computed in a lower-dimensional space resulting from a graph embedding model. SEED allows users to turn the tedious typed-query keyword search to entity centric search, which exploits structured Web data for an improved information exploration.

REFERENCES

1. Alexandr Andoni and Piotr Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM* 51, 1 (2008), 117–122.
2. Krisztian Balog, Marc Bron, and Maarten De Rijke. 2011. Query Modeling for Entity Search Based on Terms, Categories, and Examples. *ACM Trans. Inf. Syst.* 29, 4, Article 22 (Dec. 2011), 31 pages.
3. Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. 2013. Entity Recommendations in Web Search. In *ISWC*. 33–48.
4. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.
5. Edgar Meij, Krisztian Balog, and Daan Odijk. 2014. Entity Linking and Retrieval for Semantic Search. In *WSDM*. 683–684.
6. Tuukka Ruotsalo and Eero Hyvönen. 2015. Exploiting Semantic Annotations for Domain-Specific Entity Search. In *ECIR*. 358–369.
7. C. E. Shannon. 2001. A Mathematical Theory of Communication. *SIGMOBILE Mob. Comput. Commun. Rev.* 5, 1 (Jan. 2001), 3–55.