PivotE: Revealing and Visualizing the Underlying Entity Structures for Exploration

Xueran Han
Renmin University of China
hanxueran@ruc.edu.cn

Jun Chen
Renmin University of China
chenjun2013@ruc.edu.cn

Jiaheng Lu
University of Helsinki
jiaheng.lu@helsinki.fi

Yueguo Chen
Renmin University of China
chenyueguo@ruc.edu.cn

Xiaoyong Du
Renmin University of China
duyong@ruc.edu.cn

ABSTRACT

A Web-scale knowledge graph (KG) typically contains millions of entities and thousands of entity types. Due to the lack of a pre-defined data schema such as the ER model, entities in KGs are loosely coupled based on their relationships, which brings challenges for effective accesses of the KGs in a structured manner like SPARQL. This demonstration presents an entity-oriented exploratory search prototype system that is able to support search and explore KGs in an exploratory search manner, where local structures of KGs can be dynamically discovered and utilized for guiding users. The system applies a path-based ranking method for recommending similar entities and their relevant information as exploration pointers. The interface is designed to assist users to investigate a domain (particular type) of entities, as well as to explore the knowledge graphs in various relevant domains. The queries are dynamically formulated by tracing the users' dynamic clicking (exploration) behaviors.

In this demonstration, we will show how our system visualizes the underlying entity structures, as well as explain the semantic correlations among them in a unified interface, which not only assist users to learn about the properties of entities in many aspects but also guide them to further explore the information space.

PVLDB Reference Format:
Xueran Han, Jun Chen, Jiaheng Lu,Yueguo Chen, Xiaoyong Du. PivotE: Revealing and Visualizing The Underlying Entity Structures for Exploration. *PVLDB*, 12(xxx): xxxx-yyyy, 2019.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx

1. INTRODUCTION

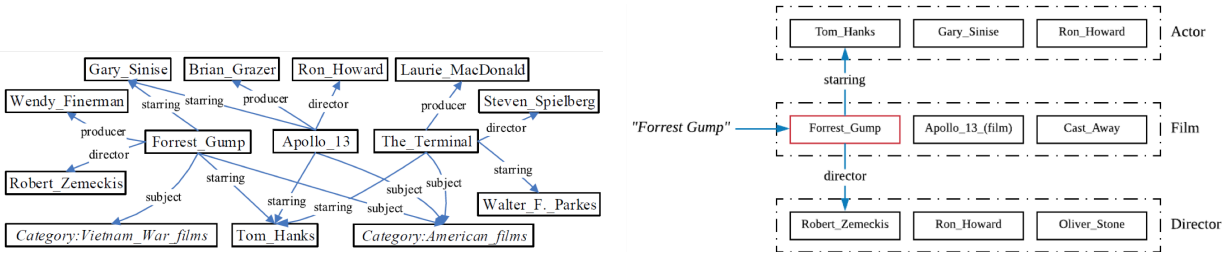
Web-scale open domain knowledge graphs such as DBpedia and Freebase contain a huge amount of entities and their relationships. We observe that entities in KGs are typically labeled with types. Entities of two types are usually (or statistically) coupled with specific relations. For example, *films* and *actors* are likely to be coupled via a relation of *starring*. Such statistically coupled relations allow us to explore KGs from entities of one type to entities of other types that are coupled to the current entity type. This motivates us to develop a system called PivotE, to enable users to explore an open domain KG with a matrix-based visual interface. There are three main challenges to achieve this goal: 1) conventional user interactions are constrained by the keyword-based input, which limits users to express their information needs clearly and reformulate queries rapidly in the unfamiliar information space; 2) millions of entities are connected by thousands of relations in knowledge graphs, which limits systems to recommend relevant entities and semantic features effectively and efficiently; 3) conventional search systems often force users to narrow the information space continually, which limits users to switch across the multi-domains freely in the information space.

Our solution is based on a concept called semantic feature which is introduced in our previous work [6]. It is composed of a predicate and an entity. For instance, as illustrated in Fig. 1-a, an examplar semantic feature has an entity *Tom_Hanks* and a directional predicate *starring* (i.e., denoted as *Tom_Hanks:starring*). The PivotE system utilizes entities and their semantic features to form a matrix (see Fig. 3) that plots the relationships between entities (x-axis, of mostly the same type) and their semantic features (y-axis). Moreover, entities (whose types are likely to be different with those in x-axis) embedded in semantic features are used as pointers to guide users for exploring other information domains.

Investigation and browse are two core operations of exploratory search [5]. In PivotE[1], for addressing challenge (2) and (3), we allow users to conduct these two operations simply based on click operations. By clicking entities in the x-axis, users provide seeds of a particular type of entity. This is called an investigation process that expands entities of the same type in the x-axis. Techniques proposed in [1] are applied as the model of entity set expansion. In addition to the investigation process, as a by-product of entity set expansion, the ranked semantic features in the y-axis, provide pointers to other entity types so that a user can apply the browse operation by pivoting the x-axis into entities of an-

[1]Our demo video and code are available at `https://github.com/lemontreehxr/KGexplore`

Proceedings of the VLDB Endowment, Vol. 12, No. xxx
ISSN 2150-8097.
DOI: https://doi.org/10.14778/xxxxxxx.xxxxxxx



(a) An example of a KG **(b)** A view of entity types

Figure 1: A knowledge graph contains a huge number of semantic features (e.g., *Tom_Hanks:starring*), which can be exploited on the fly to learn about the properties of an entity (e.g., *Forrest_Gump*) in many aspects and identify the possible search directions (e.g., *Actor* and *Director*) for further exploration.

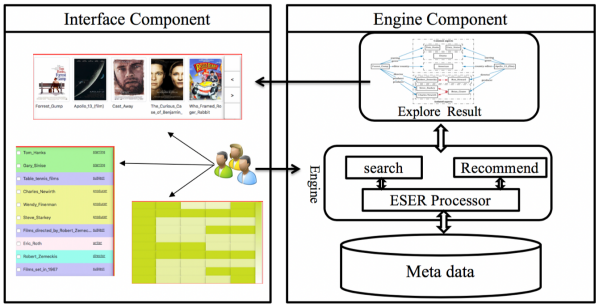


Figure 2: The architecture of the SEED system.

other type. Users can further apply an investigation process on the new x-axis to achieve a continuous data exploration process over the KGs.

Less attention has been devoted to the user interfaces for supporting users to efficiently explore the abundance of knowledge graphs in different aspects. A salient feature of the PivotE interface is that, as users explore the KGs (by clicking entities of interest), the query results are dynamically formulated and updated. Such an interactive model is beyond the traditional way to information access of KGs such as SPARQL, keyword search, or natural language questions. It achieves the goal of "learn-as-you-go" which is the target of a typical exploratory search system.

2. THE SEED SYSTEM

Figure 2 illustrates the architecture of our system, which consists of three main components: a user interface, a search engine, and a recommendation engine. For the interface, after users submit a query, the system will present relevant entities and semantic features as responses, as well as show the correlation of entities and semantic features in the form of a heat map. In the investigation stage of PivotE, we supply the exploration points for users to effectively and efficiently identify the possible search directions in the complicated information space. Users can add query conditions by selecting the semantic features of the entity or update the query by double-clicking on the image of entities. In the recommendation stage, users can flexibly switch to the relevant entity domains (e.g., *Actor* and *Director*) for exploration via the semantic features. In this way, users can gradually acquire knowledge from one data domain to others in a continuously perceptive manner, rather than blindly leap to irrelevant ones.

2.1 The User Interface

The main workspace of our prototype system PivotE is divided into five areas: the query area (see Fig. 3-a, b and g), the entity recommendation area (see Fig. 3-c), the entity presentation area (see Fig. 3-d), the semantic feature recommendation area (see Fig. 3-e), and the explanation area (see Fig. 3-f).

Users are allowed to type keywords to formulate an initial query (see Fig. 3-a). After submitting a query, relevant entities with respect to the query and their relevant semantic features are returned in the entity and semantic feature recommendation areas respectively (see Fig. 3-c and d). Users can look up the profile of a particular entity by clicking it (see Fig. 3-d). Besides, users can discover the semantic correlation among entities and semantic features via the explanation area (see Fig. 3-f), which is illustrated as a heat map for an overview (i.e., the darker the color, the stronger the semantic correlation between an entity and a semantic feature, and vice versa). In this way, users can have a better understanding of the search context and understand the recommendation of the system and then identify a more reasonable search direction for further exploration.

To support query reformulation rapidly, users are allowed to manipulate entities and semantic features (i.e., from Fig. 3-c and e) directly to facilitate all fundamental tasks like selection, duplication, deletion, etc. An existing query could be easily reformulated by the addition or removal of such entities and semantic features (see Fig. 3-b), and then the results will be updated accordingly. Moreover, users can revisit the queries in the timeline (see Fig. 3-g), as well as view the visualization of their search behaviors such as submitting a query, looking up an entity and horizontally exploring by specifying an entity (see Fig. 4). This assists them to clarify the search context and their positions during long-term search sessions and supports them to compare the information by conveniently revisiting historical queries.

2.2 The Search Engine

The search engine is designed to retrieve entities matching to the given keywords. Since multi-fielded entity representation has been proved to be beneficial for entity search based on knowledge graphs, we apply a five-field entity representation scheme for describing an entity as illustrated in Tab. 1, including names (i.e., its labels), attributes (i.e., its literals), categories (i.e., the labels of its categories), similar entity names (i.e., the labels of the redirected and disambiguated entities) and related entity names (i.e., the labels of the connected entities). The mixture of language models (i.e., a multi-fielded extension of the query likelihood

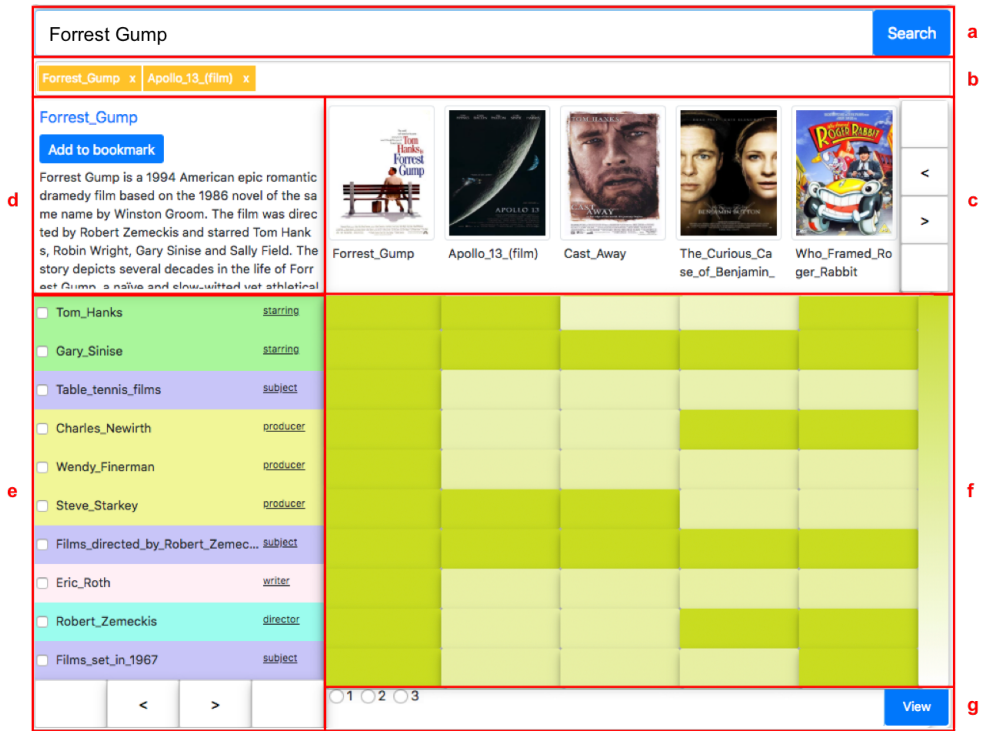


Figure 3: User interface of PivotE, the main work space is divided in four areas: the query area (see Fig. 3-a, b and g), the entity recommendation area (see Fig. 3-c), the entity presentation area (see Fig. 3-d), the semantic feature recommendation area (see Fig. 3-e), the explanation area (see Fig. 3-f).

retrieval model, where the retrieval score of a structured document is a linear combination of probabilities of query terms in the language models calculated for each document field) [4], is applied in our prototype system for returning top-k relevant entities with respect to the query.

Table 1: The multi-fielded entity representation for *Forrest_Gump*

Field	Content
names	Forrest Gump
attributes	"142 minutes", "55 million dollars", etc.
categories	American films, etc.
similar entities names	Geenbow, Gumpian, etc.
related entity names	Tom Hanks, Robert Zemeckis, etc.

2.3 The Recommendation Engine

The recommendation engine is designed to recommend similar entities and their relevant semantic features. Given several examplar entities, we apply a state-of-the-art method [6] to return a ranked list of entities, which applies the highly relevant semantic features of entities in an error-tolerant manner. For a semantic feature, an entity is tackled by estimating the semantic correlation between them, so that more semantic features can be applied for estimating the relevance of the candidate entities to the given example entities. Following this method, we can return the highly relevant semantic features along with these similar entities as the recommendation, as well as explain the semantic correlation among them.

In our system, we represent the KG as a set of triples such as $< s, p, o >$, and use κ to represent the RDF KGs. There are two types of SPs: $< e, p, x >$ and $< x, p, e >$, where x is an entity variable, e is called an anchor entity. The former SP $< e, p, x >$ (shorted as $e : p$) represents a triple pattern having e as the subject, p as the predicate, and the latter one $< x, p, e >$ (shorted as $e : \underline{p}$) represents a triple pattern having e as the object and p as the predicate. For instance, SF π = *Tom_Hanks:starring* means the triple pattern of the entities that have *Tom_Hanks* as a star. When an entity e has a path of SF π, then we denote the entity e as $e \models \pi$ and $E(\pi) = \{e | e \models \pi\}$ means the number of target entities. The process of ranking model can be divided into entities ranking and SFs ranking [6].

2.3.1 The ranking model of SFs

In order to evaluate the correlation between an SF π and a query Q effectively, we use the multiplication of the discrimainability and commonality of an SF π in the κ to represent the similarity of an SF π and a query Q.

$$r(\pi, Q) = d(\pi) \times c(\pi, Q)$$

Given a query Q containing m seeds, these seeds and the $E(\pi)$ likely discover many entities. In order to weaken the similarity of frequent SFs which is widely shared by many entities, we denote the discriminability of an SF π based on the idea of IDF(Inverse Document Frequency). Therefore, the discriminability of an SF π is defined as:

$$d(\pi) = \frac{1}{\|E(\pi)\|}$$

Before computing the commonality of SFs, we first compute $p(\pi|e)$ which is the probability of an entity e having a SF π:

$$p(\pi|e) = \begin{cases} 1 & , \text{ if } e \models \pi \\ p(\pi|c^*) = \dfrac{\|E(\pi) \bigcap E(c*)\|}{\|E(c^*)\|} & \text{otherwise} \end{cases}$$

We then compute the commonality of a SF to a query as:

$$c(\pi, Q) = \prod_{e \in Q} p(\pi|e)$$

2.3.2 *The ranking model of entities*

Inspired by the mechanism of the ranking model of SFs, the similarity of entities and queries consists of two components which are the probability of an entity e having an SF π and the relevance of an SF π to a query Q. Therefore we formalize the relevance of an entity e and a query Q as:

$$r(e, Q) = \sum_{\pi \in \Phi(Q)} p(\pi|e) \times r(\pi, Q)$$

We divide the correlation of entities and semantic features into seven levels, and visualize them with a heat-map, which explores a number of semantic features in KGs and reveals the potential association between entities and semantics.

3. DEMO SCENARIOS

As shown in Figure 3, our demo mainly consists of two scenarios with respect to *entity investigation* and *search domain exploration*.

3.1 Entity investigation

First, to start a search session, users input keywords to obtain a set of entities and their relevant semantic features as search results (see Fig. 3-a). We also support users to express their information needs by using entities and semantic features (see Fig. 3-c and e). For instance, users can express the query intention *"Find films starring Tom Hanks"* by specifying the semantic feature *Tom_Hanks:starring*, as well as express the query intention *"Find films similar to Forrest Gump"* by simply specifying the entity *Forrest_Gump*. Users can click the entity name in Fig. 3-d, which can be redirected to Wikipedia to learn more information in detail. In such a way, users can not only narrow the information space in different aspects but also deeply investigate similar entities in the same data domain.

3.2 Search Domain Exploration

In the process of recommendation, our system can dynamically recommend relevant semantic features of entities (see Fig.3-e) and supply the semantic correlations between them as explanations (see Fig.3-f). The relevant semantic features can be recommended as the properties of entities for iterative exploration. Users can change the search domain and filter entity types by double-clicking the image of the entities (see Fig. 3-c) or the name of entities (see Fig. 3-e). In order to help the reasoning process of the recommendation, we plot the heat map, which can reveal the correlations between entities and features (see Fig. 3-f). For instance, if the system explains the semantic correlation between *Forrest_Gump* and *Apollo_13_(film)* is that both of them are performed by *Tom_Hanks* and *Gary_Sinise*, users may have a better understanding about the search context, and then identify a more reasonable search direction for further exploration (e.g., further exploring the films performed by *Tom_Hanks* by specifying the semantic feature *Tom_Hanks:starring*).

In order to illustrate the recommendation process in detail, we organize queries in a timeline for tracebacks (see Fig. 3-g). As shown in Fig. 4, users can click the "view" bottom if they want to view the exploratory search path and search content.

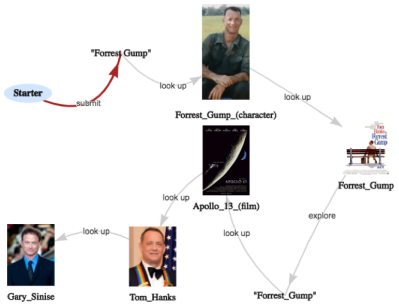


Figure 4: An example of the exploratory path

4. RELATED WORK AND CONCLUSION

Traditional search systems based on KGs focus on supporting users to iteratively formulate queries for better addressing semantic search which is focus on entity search tasks, such as Pilot[2] and PandaSearch[3] which search a particular set of entities matching to the SPARQL or well-defined keywords. In this demo, we design and implement a novel prototype system called PivotE for entity-oriented exploratory search in KGs. It applies a path-based ranking method for recommending entities and their relevant information as exploration pointers, which assists users to learn about the properties of entities and guides them to explore the KGs in different aspects. Our demo is also able to manipulate entities and explore search domain flexibly to express their information needs beyond the keyword-based search.

5. ACKNOWLEDGMENTS

Yueguo Chen is the corresponding author. This work is supported by National Key Research and Development Program (No. 2018YFB1004401) and the National Science Foundation of China under grants U1711261, No. 61472426.

6. REFERENCES

[1] J. Chen, Y. Chen, X. Zhang, X. Du, K. Wang, and J.-R. Wen. Entity set expansion with semantic features of knowledge graphs. *Journal of Web Semantics*, 52:33–44, 2018.

[2] T. Cheng, K. C.-C. Chang, et al. *Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web*. PhD thesis, University of Illinois at Urbana-Champaign, 2007.

[3] F. Huang, J. Li, J. Lu, T. Ling, and Z. Dong. Pandasearch: A fine-grained academic search engine for research documents. *ICDE*, pages 1408–1411, 05 2015.

[4] J. M. Ponte and W. B. Croft. *A language modeling approach to information retrieval*. PhD thesis, University of Massachusetts at Amherst, 1998.

[5] R. W. White and R. A. Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–98, 2009.

[6] X. Zhang, Y. Chen, J. Chen, X. Du, K. Wang, and J.-R. Wen. Entity set expansion via knowledge graphs. In *SIGIR*, pages 1101–1104, 2017.