Explass: Exploring Associations between Entities via Top-K Ontological Patterns and Facets

Gong Cheng, Yanan Zhang, and Yuzhong Qu

State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing 210023, P.R. China
{gcheng,yzqu}@nju.edu.cn, ynzhang@smail.nju.edu.cn

Abstract. Searching for associations between entities is needed in many areas. On the Semantic Web, it usually boils down to finding paths that connect two entities in an entity-relation graph. Given the increasing volume of data, apart from the efficiency of path finding, recent research interests have focused on how to help users explore a large set of associations that have been found. To achieve this, we propose an approach to exploratory association search, called Explass, which provides a flat list (top-K) of clusters and facet values for refocusing and refining the search. Each cluster is labeled with an ontological pattern, which gives a conceptual summary of the associations in the cluster. Facet values comprise classes of entities and relations appearing in associations. To recommend frequent, informative, and small-overlapping patterns and facet values, we exploit ontological semantics, query context, and information theory. We compare Explass with two existing approaches by conducting a user study over DBpedia, and test the statistical significance of the results.

Keywords: Association exploration, clustering, exploratory search, faceted search, ontological association pattern.

1 Introduction

Searching for associations (a.k.a. relationships) between two entities is needed in many areas. For instance, a security agent may be interested in the associations between two suspected terrorists. A historian may study the associations between two politicians in history. Researchers may be curious about the associations between each other in an academic network.

Carrying out association search over unstructured data on the Web, e.g. webpage text [15], is not an easy task because direct relations between entities need to be extracted from ambiguous text, and finding indirect associations may even have to integrate information from multiple sources. In recent years, association search has been facilitated by the availability of graph-structured data on the Semantic Web, which exactly describes entities and the relations between them, and can relatively be easily integrated from different sources. In such an entity-relation graph, associations between two entities are explicitly captured by the paths that connect these two vertices. Association search is then transformed into path finding [3], and it faces two challenges when entity-relation graphs

P. Mika et al. (Eds.) ISWC 2014, Part II, LNCS 8797, pp. 422–437, 2014.

become very large: how to efficiently find associations, and *how to help users explore a large set of associations that have been found*. The latter challenge will be addressed in this paper.

We meet this challenge by realizing exploratory search for associations. Exploratory search [16] is designed to serve complex and uncertain information needs, which is often the case in association search. It aims to help users explore, process, and interpret a large set of search results via continuous and exploratory interaction, mainly based on statically defined facets or dynamically generated clusters [9]. Distinguished from existing work on exploratory association search [10,19], our contribution is summarized as follows.

- Our approach to association exploration, called Explass, provides a flat list (top-K, rather than a hierarchy [19]) of clusters for refocusing. Each cluster is labeled with an ontological association pattern (or pattern for short), which makes up of classes and relations and preserves the path structure of association. It can give users a conceptual summary of the associations in the cluster.
- To obtain clusters, i.e. to recommend patterns, we propose to firstly mine all the significant patterns that are highly relevant to the query context by formulating and solving a data mining problem, and then find top-K ones that are as frequent and informative as possible while sharing small overlap between each other by formulating and solving an optimization problem.
- In this novel solution, the frequency of a pattern reflects its relevance to the query context. The informativeness of a pattern is learned from the entity-relation graph according to information theory. The overlap between patterns is identified based on ontological semantics and query context.
- Further, Explass integrates patterns with facet values, which are classes of entities and relations appearing in associations, and can be used to refine the search as filters. Rather than showing all of them [10], we adapt the above solution to recommend top-K ones. We will show that patterns and facets are complementary in terms of usage in association exploration.
- We implement a prototype of Explass based on DBpedia. To investigate how patterns and facets help users explore associations in practice, we compare Explass with two existing approaches by conducting a user study, and test the statistical significance of the results.

The remainder of the paper is structured as follows. Section 2 discusses related work. Section 3 presents an overview of Explass. Section 4 gives some preliminaries. Section 5 and 6 describe the recommendation of patterns and facet values, respectively. Section 7 reports a user study. Section 8 concludes the paper.

2 Related Work

Definitions of Association. Given an entity-relation graph, association between entities has various definitions. Anyanwu and Sheth [3] defined four types of associations, in which path-based association has received the most attention

and is adopted by this paper. Among other definitions, in REX [6], an association conforms to a certain constrained graph pattern, and is obtained by combining paths. In Ming [12], an association between a set of entities is a connected subgraph containing all of them. In this paper, *we will not address these different definitions, and will only deal with path-based association.*

Association Discovery and Ranking. Discovering path-based associations boils down to finding paths in an entity-relation graph, which is a challenge when the graph is large, and has attracted considerable interest [7,11]. However, *what we focus on in this paper is a problem that follows, namely how to help users explore a large set of associations that have been found.* So far, major efforts addressing this issue were made to appropriately rank associations so that more important ones could be shown earlier. Existing ranking methods exploit various structural features of an association [1,2], consider query relevance [20], and produce personalized results [5]. *Complementary to ranking, another line of research builds on exploratory search, and our approach belongs to this category.*

Exploratory Association Search. Exploratory search [16] serves complex and uncertain information needs, and expects search systems to facilitate cognitive processing and interpretation of a large set of search results via continuous and exploratory interaction that goes beyond lookup and ranking. Facets and clustering are two popular methods for realizing this [9]. Facets are usually statically defined, whereas clustering lets search results speak for themselves. Both facets and clustering have been widely adopted in Web search and, in particular, in entity search [14,17]. Recently, they have also been adopted in association search [10,19]. Among existing attempts, RelFinder [10] employs classes of entities and relations appearing in associations as facet values for refining the search and filtering associations. RelClus [19] organizes associations inclusively into a hierarchy of clusters for refocusing, where each cluster is labeled with a pattern. In this paper, *we realize exploratory association search in a new way. Our Explass integrates both clusters (i.e. patterns) and facets (i.e. classes and relations)* and, in particular, *it provides a flat list (top-K) of informative patterns*, thereby avoiding deep and complicated hierarchical organization as well as very general and meaningless high-level patterns met on RelClus. Technically, different from [10,19], *we give our attention to the recommendation of patterns and facet values, and consider their frequency, informativeness, and overlap by exploiting ontological semantics, query context, and information theory.* We will compare Explass with RelFinder and RelClus in a user study.

3 Overview of Explass

Before formally introducing Explass, in this section, we illustrate the exploration operations it supports. A prototype based on DBpedia is available online.[1]

As illustrated in Fig. 1, after obtaining a set of associations between two entities, Explass recommends a set (top-K) of path-structured patterns (cf. Sect. 5)

[1] `http://ws.nju.edu.cn/explass/`

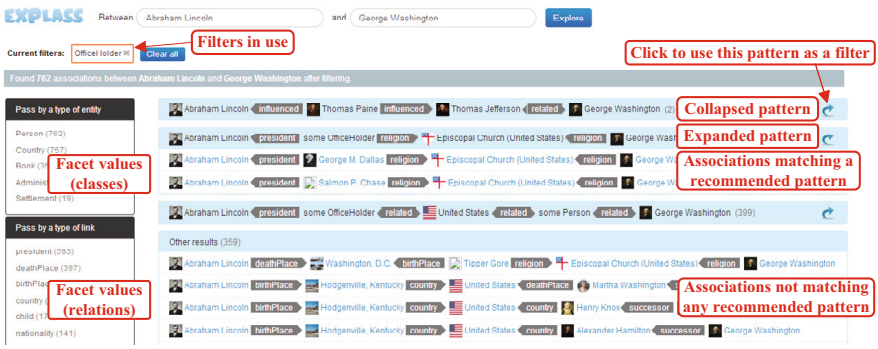

Fig. 1. A prototype of Explass based on DBpedia

and a set (top-K) of facet values (cf. Sect. 6) for further exploration. Firstly, all the associations matching a recommended pattern are clustered and placed under this pattern, which provides a conceptual summary of these associations. It is followed by the number of these associations in parentheses, and is expandable/collapsible to show/hide them for refocusing. Associations not matching any recommended pattern are placed at the end. Secondly, a pattern can also be used as a filter to refine the search. After that, search results will be limited to those matching this pattern, and all the recommendations will be re-computed. Filters in use can be canceled. Thirdly, classes of entities and relations appearing in associations comprise facet values, each of which is followed by the number of associations to expect if using this class/relation as a filter to refine the search and limit search results to those containing its instance/occurrence.

4 Preliminaries

Table 1 and Fig. 2–4 comprise a running example in this paper.

Let $\Sigma_E, \Sigma_C, \Sigma_R$ be the sets of all entities, classes, and relations (i.e. properties connecting entities), respectively. An entity is an instance of one or more classes, as illustrated in Table 1. For each class c, let $I(c)$ be the set of all its instances. Classes are organized into a class hierarchy describing the subclass-superclass relation denoted by $\sqsubseteq_C$, as illustrated in Fig. 3. At the top of the class hierarchy, `ENTITY` represents a superclass of all other classes, and every entity is an instance of this class. Similarly, a relation hierarchy describing the subrelation-superrelation relation denoted by $\sqsubseteq_R$ is illustrated in Fig. 4, the top of which is called `RELATED`.

Entities and the relations connecting them form an *entity-relation graph*, as illustrated in Fig. 2, which is formalized as a labeled directed graph $G = \langle V, A, s, t, l_V, l_A \rangle$, where

- V is a finite set of vertices,
- A is a finite set of directed arcs,

Table 1. Entities and Their Classes

Entity	Class
`Alice`, `Bob`	`Person`, `ENTITY`
`PaperA`, `PaperB`, `PaperC`, `PaperD`	`ConfPaper`, `Publication`, `ENTITY`
`ArticleA`	`JArticle`, `Publication`, `ENTITY`
`ConfA`, `ConfB`	`Conference`, `ENTITY`

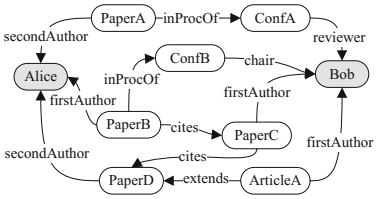

Fig. 2. An entity-relation graph

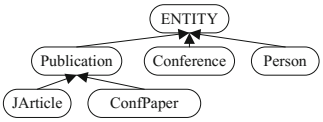

Fig. 3. A class hierarchy

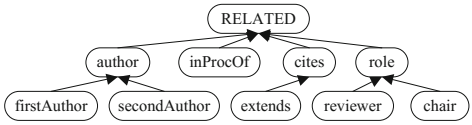

Fig. 4. A relation hierarchy

- $s : A \mapsto V$ returns the source vertex of each arc,
- $t : A \mapsto V$ returns the target vertex of each arc,
- $l_V : V \mapsto \Sigma_E$ returns the unique label of each vertex, which is an entity, and
- $l_A : A \mapsto \Sigma_R$ returns the label of each arc, which is a relation.

An *association* from an entity e_S to an entity e_E comprises the labels of the vertices and arcs (i.e. entities and relations) in a path in G from e_S to e_E where no vertices are repeated and arcs not necessarily go the same direction. To differentiate between the two directions of an arc, the label r of each "reverse" arc going from e_E to e_S is substituted by a pseudo-relation $\hat{}r$. In particular, $\hat{}r_i \sqsubseteq_R \hat{}r_j$ if and only if $r_i \sqsubseteq_R r_j$; and for the top relation, $\hat{}\texttt{RELATED} = \texttt{RELATED}$. Then formally, corresponding to a path $v_0 a_1 \cdots a_n v_n$ from $e_S = l_V(v_0)$ to $e_E = l_V(v_n)$ which is an alternating sequence of vertices and arcs, an association of length n from e_S to e_E is an alternating sequence of relations and entities, $r_1 e_1 \cdots e_{n-1} r_n$, beginning and ending with a relation, and subject to

- for $1 \leq i \leq n-1$, $e_i = l_V(v_i)$, and
- for $1 \leq i \leq n$, if $s(a_i) = v_{i-1}$, then $r_i = l_A(a_i)$; otherwise, $r_i = \hat{} l_A(a_i)$.

For instance, the entity-relation graph in Fig. 2 contains five associations of length 3 from `Alice` to `Bob`:

$$
\begin{aligned}
Z_1 &: \hat{}\texttt{secondAuthor PaperA inProcOf ConfA reviewer} \\
Z_2 &: \hat{}\texttt{firstAuthor PaperB inProcOf ConfB chair} \\
Z_3 &: \hat{}\texttt{firstAuthor PaperB cites PaperC firstAuthor} \\
Z_4 &: \hat{}\texttt{secondAuthor PaperD } \hat{}\texttt{cites PaperC firstAuthor} \\
Z_5 &: \hat{}\texttt{secondAuthor PaperD } \hat{}\texttt{extends ArticleA firstAuthor}
\end{aligned}
\tag{1}
$$

An *ontological association pattern* (or pattern for short) provides an abstraction of association by substituting entities with classes they belong to and optionally substituting relations with their superrelations. More formally, a pattern

of length n is an alternating sequence of relations and classes, $r_1 c_1 \cdots c_{n-1} r_n$, beginning and ending with a relation. An association $Z = r_1 e_1 \cdots e_{n-1} r_n$ *matches* a pattern $P = r'_1 c'_1 \cdots c'_{n-1} r'_n$, denoted by $Z \in M(P)$, if

- for $1 \leq i \leq n-1$, $e_i \in I(c'_i)$, and
- for $1 \leq i \leq n$, $r_i \sqsubseteq_R r'_i$.

For instance, both Z_1 and Z_2 in Eq. (1) match several different patterns such as

$$P_1 : \texttt{\^{}author ConfPaper inProcOf ENTITY RELATED}. \tag{2}$$

To also allow entities to appear in a pattern, for each entity e, a pseudo-class $psc(e)$ is introduced that has e as its only instance, i.e. $I(psc(e)) = \{e\}$, and is a subclass of every other class that e belongs to. Then, Z_1 also matches

$$P_2 : \texttt{\^{}author } psc(\texttt{PaperA}) \texttt{ inProcOf ENTITY RELATED}. \tag{3}$$

5 Pattern Recommendation

Given $\mathcal{Z}$, a set of associations from an entity e_S to another e_E found in the entity-relation graph G, we aim to recommend up to K patterns for exploring $\mathcal{Z}$. We firstly mine all the significant patterns that are highly relevant to the query context, and then find up to K of them that are as frequent and informative as possible while sharing small overlap between each other.

We assume the associations in $\mathcal{Z}$ are all of length n. Otherwise, we can group them by length, and recommend patterns for each group and show all of them.

5.1 Mining Significant Patterns

Given $\mathcal{Z}$ and a pattern P, to characterize the relevance of P to the query context, we define the *frequency* of P w.r.t. $\mathcal{Z}$ as

$$\begin{aligned} freq(P) &= \frac{|hits(P)|}{|\mathcal{Z}|} \\ hits(P) &= \{Z \in \mathcal{Z} \,:\, Z \in M(P)\}, \end{aligned} \tag{4}$$

which is in the range $[0, 1]$. For instance, given $\mathcal{Z}$ comprising the five associations in Eq. (1), the frequency of P_1 in Eq. (2) is $\frac{2}{5}$ because it is matched by Z_1 and Z_2, i.e. by 2 out of the 5 associations.

We aim to find all the *significant patterns*, denoted by $\mathcal{P}_{\mathcal{Z}}$, namely those having a frequency higher than a threshold $\tau \in [0, 1]$. We formulate it as a frequent closed itemset mining problem (FCIMP), which has been extensively studied in the field of data mining [8]. A tricky issue in the formulation is how to encode the path structure of association and pattern.

Specifically, each association in $\mathcal{Z}$ corresponds to a "transaction" (which is a set of "items") in FCIMP, and an "item" is a position-relation pair in $\{1, 3, \ldots, 2n-1\} \times \Sigma_R$ or a position-class pair in $\{2, 4, \ldots, 2n-2\} \times \Sigma_C$. An

association $Z = r_1 e_1 \cdots e_{n-1} r_n$, as a "transaction", contains a position-relation pair $\langle 2i-1, r\rangle$ if $r_i \sqsubseteq_R r$, and contains a position-class pair $\langle 2i, c\rangle$ if $e_i \in I(c)$. For instance, Z_1 in Eq. (1) contains $\langle 1, \hat{}\texttt{secondAuthor}\rangle$, $\langle 1, \hat{}\texttt{author}\rangle$, $\langle 1, \texttt{RELATED}\rangle$, $\langle 2, psc(\texttt{PaperA})\rangle$, $\langle 2, \texttt{ConfPaper}\rangle$, $\langle 2, \texttt{Publication}\rangle$, $\langle 2, \texttt{ENTITY}\rangle$, etc.

Then, we use CHARM [18] to find all the frequent closed "itemsets" being subsets of at least $\tau|\mathcal{Z}|$ "transactions". From such a frequent closed "itemset", we try to obtain a pattern by selecting, if possible, one position-relation or position-class pair for each position in $\{1, 2, \ldots, 2n-1\}$ and arranging these relations and classes in ascending order of their positions. Once achieved, it can be proved that the pattern obtained is a significant pattern, and all the significant patterns can be obtained in this way. The proof is straightforward and is omitted due to lack of space.

5.2 Finding Frequent, Informative, and Small-Overlapping Patterns

Among all the significant patterns, we aim to find up to K ones that are as frequent and informative as possible while sharing small overlap between each other. In the following, firstly we define the informativeness of a pattern and the overlap between two patterns. Then we formulate and solve an optimization problem to integrate frequency, informativeness, and overlap.

Informativeness. A significant pattern may provide little information and become meaningless, e.g. one comprising only `ENTITY` and `RELATED`. However, we prefer to recommend informative patterns. To quantify the informativeness of a pattern, we measure the informativeness of each class and relation in the pattern.

As to classes, the idea is that a class having fewer instances is more specific and thus more informative. We formulate it using information theory. Specifically, for each class c, let $pr(c)$ be the probability that a random entity belongs to c. By estimating it based on the entity-relation graph $G = \langle V, A, s, t, l_V, l_A\rangle$, we measure $sinf(c)$, the self-information of the event that c is indeed observed as a class of some entity:

$$\begin{aligned} sinf(c) &= -\log pr(c) \\ pr(c) &= \frac{|\{v \in V \,:\, l_V(v) \in I(c)\}|}{|V|}\,. \end{aligned} \tag{5}$$

For instance, given G in Fig. 2, $sinf(\texttt{ConfPaper}) = -\log\frac{4}{9}$ because 4 out of the 9 entities in G belong to `ConfPaper`. Further, we normalize $sinf(c)$ into the range $[0, 1]$ as the *informativeness of class c*:

$$sinf_N(c) = \frac{sinf(c)}{\log|V|}\,. \tag{6}$$

As to relations, the idea is similar but more complex because a relation has two ends (i.e. connecting two entities), called the source end and the target end, and each of them can be treated as a random variable. We separately process the two

ends and integrate the results. Firstly, we treat the target end of a relation r as a random variable and measure its entropy, denoted by $\overrightarrow{eta}(r)$, which quantifies the expected value of the self-information of its outcomes (i.e. all possible entities appearing at the target end of r, denoted by $\overrightarrow{val}(r)$). By estimating $\overrightarrow{pr}(r, e)$, the probability of observing each outcome e based on $G = \langle V, A, s, t, l_V, l_A \rangle$, we have

$$\begin{aligned} \overrightarrow{eta}(r) &= -\sum_{e \in \overrightarrow{val}(r)} \overrightarrow{pr}(r, e) \log \overrightarrow{pr}(r, e) \\ \overrightarrow{val}(r) &= \{e \in \Sigma_E : \exists a \in A, (l_A(a) \sqsubseteq_R r, t(a) = e)\} \\ \overrightarrow{pr}(r, e) &= \frac{|\{a \in A : l_A(a) \sqsubseteq_R r, t(a) = e\}|}{|\{a \in A : l_A(a) \sqsubseteq_R r\}|}. \end{aligned} \tag{7}$$

For instance, given G in Fig. 2, $\overrightarrow{val}(\texttt{firstAuthor}) = \{\texttt{Alice}, \texttt{Bob}\}$ because only `Alice` and `Bob` (2 times) appear at the target end of `firstAuthor`, and `Bob` appears 2 out of the 3 times so that $\overrightarrow{pr}(\texttt{firstAuthor}, \texttt{Bob}) = \frac{2}{3}$. Further, we normalize $\overrightarrow{eta}(r)$ into the range $[0, 1]$:

$$\overrightarrow{eta}_N(r) = \frac{\overrightarrow{eta}(r)}{\log |\{a \in A : l_A(a) \sqsubseteq_R r\}|}. \tag{8}$$

The source end of r is processed analogously, and its normalized entropy is denoted by $\overleftarrow{eta}_N(r)$. To integrate $\overrightarrow{eta}_N(r)$ and $\overleftarrow{eta}_N(r)$, we calculate their harmonic mean in the range $[0, 1]$ as the *informativeness of relation* r:

$$eta(r) = \frac{2 \cdot \overrightarrow{eta}_N(r) \cdot \overleftarrow{eta}_N(r)}{\overrightarrow{eta}_N(r) + \overleftarrow{eta}_N(r)}. \tag{9}$$

Finally, the *informativeness of a pattern* $P = r_1 c_1 \cdots c_{n-1} r_n$ is obtained by adding up the informativeness of the classes and relations it contains:

$$inf(P) = \sum_{i=1}^{n-1} sinf_N(c_i) + \sum_{i=1}^{n} eta(r_i). \tag{10}$$

Overlap. Patterns sharing considerably large overlap are redundant and will not be recommended together. We identify two types of overlap between patterns.

Firstly, given two patterns $P = r_1 c_1 \cdots c_{n-1} r_n$ and $P' = r'_1 c'_1 \cdots c'_{n-1} r'_n$, we check the subclass-superclass and subrelation-superrelation relations in all their corresponding positions. Based on the following two functions:

$$\begin{aligned} ss_C(c_i, c_j) &= \begin{cases} 1 & \text{if } c_i \sqsubseteq_C c_j \text{ or } c_j \sqsubseteq_C c_i, \\ 0 & \text{otherwise,} \end{cases} \\ ss_R(r_i, r_j) &= \begin{cases} 1 & \text{if } r_i \sqsubseteq_R r_j \text{ or } r_j \sqsubseteq_R r_i, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{11}$$

we define the *ontological overlap* between P and P' in the range $[0,1]$ as

$$ovlp_O(P,P') = \frac{\sum_{i=1}^{n-1} ss_C(c_i,c'_i) + \sum_{i=1}^{n} ss_R(r_i,r'_i)}{2n-1}. \tag{12}$$

For instance, the ontological overlap between P_1 in Eq. (2) and P_2 in Eq. (3) is $\frac{5}{5}$ because $\sqsubseteq_C$ or $\sqsubseteq_R$ holds in all the 5 positions.

Secondly, we check to what extent P and P' are matched by common associations in $\mathcal{Z}$. By using the Jaccard similarity, we define the *contextual overlap* between P and P' in the range $[0,1]$ as

$$ovlp_C(P,P') = \frac{|hits(P) \cap hits(P')|}{|hits(P) \cup hits(P')|}, \tag{13}$$

where *hits* is given by Eq. (4). For instance, given $\mathcal{Z}$ comprising the five associations in Eq. (1), the contextual overlap between P_1 in Eq. (2) and P_2 in Eq. (3) is $\frac{1}{2}$ because P_1 is matched by Z_1 and Z_2, and P_2 is matched by Z_1.

Optimization. In $\mathcal{P}_\mathcal{Z}$, the set of significant patterns mined from $\mathcal{Z}$, we aim to find up to K ones that are as frequent and informative as possible while sharing small overlap between each other. It can be formulated as a multidimensional 0-1 knapsack problem (MKP) [13]. Specifically, each $P_i \in \mathcal{P}_\mathcal{Z}$ corresponds to a candidate "item" to be selected whose "profit" is $freq(P_i) \cdot inf(P_i)$ and whose "weight" is 1, when the "capacity" of the "knapsack" is K. For each pair of patterns sharing considerably large ontological or contextual overlap, an additional constraint is introduced to require that they are not selected together.

More formally, we number the patterns in $\mathcal{P}_\mathcal{Z}$ from P_1 to $P_{N=|\mathcal{P}_\mathcal{Z}|}$, and introduce a series of binary variables x_i to indicate whether pattern P_i is selected. Then we formulate a MKP as:

$$\begin{aligned}
&\text{maximize } \sum_{i=1}^{N} x_i \cdot freq(P_i) \cdot inf(P_i) \\
&\text{subject to} \\
&\quad \sum_{i=1}^{N} x_i \le K, \\
&\quad \sum_{i=1}^{N} x_i w_i^{j,k} \le 1 \text{ for } j,k = 1,\ldots,N \text{ s.t. } j \ne k \text{ and} \\
&\qquad\qquad ovlp_O(P_j,P_k) \ge \mu_O \text{ or } ovlp_C(P_j,P_k) \ge \mu_C, \\
&\quad x_i \in \{0,1\} \text{ for } i = 1,\ldots,N,
\end{aligned} \tag{14}$$

where $\mu_O, \mu_C \in [0,1]$ are thresholds, and

$$w_i^{j,k} = \begin{cases} 1 & \text{if } i = j \text{ or } i = k, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

MKP is NP-hard [13]. To find a reasonably good feasible solution within reasonable running time, we use a greedy algorithm that considers the "items" (i.e. patterns) one after another and puts an "item" into the "knapsack" if adding this "item" would not violate any constraint. We use the following greedy heuristic to order the "items" in descending order:

$$g(P_i) = \frac{freq(P_i) \cdot inf(P_i)}{w(P_i)}, \tag{16}$$

where $w(P_i)$ returns the total "weight" of P_i in all the constraints. That is, priority is given to patterns that are more frequent, more informative, and share considerably large overlap with fewer patterns in $\mathcal{P}_{\mathcal{Z}}$.

6 Facet Value Recommendation

We also aim to recommend up to K classes of entities and K relations appearing in the associations in $\mathcal{Z}$ as facet values. In accordance with the recommendation of patterns, we adapt our solution described in Sect. 5 to recommend facet values that are as frequent and informative as possible while sharing small overlap between each other. To achieve this, we only need to redefine frequency, informativeness, and overlap for facet values. In this section, we will do this only for classes due to lack of space. Relations can be processed in an analogous way.

Firstly, given $\mathcal{Z}$ and a class c, similar to Eq. (4), we define the frequency of c w.r.t. $\mathcal{Z}$ as

$$\begin{aligned} freq(c) &= \frac{|hits(c)|}{|\mathcal{Z}|} \\ hits(c) &= \{r_1 e_1 \cdots e_{n-1} r_n \in \mathcal{Z} : \exists e_i \in I(c)\}. \end{aligned} \tag{17}$$

For instance, given $\mathcal{Z}$ comprising the five associations in Eq. (1), the frequency of `Conference` is $\frac{2}{5}$ because its instances `ConfA` and `ConfB` appear in Z_1 and Z_2, respectively, i.e. in 2 out of the 5 associations.

Secondly, the informativeness of c has been given by Eq. (6).

Thirdly, two classes c and c' share ontological overlap if one of them is a subclass of the other, i.e. $ss_C(c, c') = 1$ according to Eq. (11). Contextual overlap between classes is defined similar to Eq. (13) by using *hits* given by Eq. (17). When formulating a MKP, for each pair of classes sharing ontological or considerably large (i.e. $\geq \mu_C$) contextual overlap, an additional constraint is introduced to require that they are not selected together.

7 User Study

To investigate how patterns and facets help users explore associations in practice, we invited twenty university students to carry out association exploration tasks over DBpedia by using Explass and two existing approaches to association exploration. By analyzing subjects' responses to questionnaires and their behavior during the experiment, we mainly aimed to test the following two hypotheses.

H1. For association exploration, providing a flat list (top-K) of frequent, informative, and small-overlapping patterns (as on Explass) is more satisfying than an inclusive hierarchy of patterns (as on RelClus [19]).
H2. Patterns and facets are notably complementary in terms of usage in association exploration, and thus providing both of them (as on Explass) is more satisfying than only one of them (as on RelFinder [10] and RelClus [19]).

7.1 Data Sets

We used DBpedia in our experiment. Specifically, the entity-relation graph was obtained from the *mapping-based properties* data set, excluding RDF triples containing literals. Classes of entities were obtained from the *mapping-based types* data set. Class and relation hierarchies were obtained from the *DBpedia ontology*. The *short abstracts* and *images* data sets were used to provide a textual description and an image for each entity, respectively, which will be detailed later.

7.2 Tasks

To the best of our knowledge, there were no benchmark association exploration tasks available for evaluation. So we established a set of association exploration tasks to be used in our experiment as well as in future research. Our association exploration tasks were derived from the 100 training queries[2] provided by the multilingual question answering challenge of the QALD-3 evaluation campaign, which mentioned a total of 72 distinct entities in DBpedia. The names of these entities (e.g. *Abraham Lincoln*) were submitted to Google Search, some triggering Google's Knowledge Graph to return related entities that "people also search for" (e.g. *George Washington*, *John F. Kennedy*). For each search, among the entities returned that could also be found in DBpedia, the first one (e.g. *George Washington*) was selected, and then an association exploration task was defined in the following way.

> Suppose you will write an article about the associations between *Abraham Lincoln* and *George Washington*. Use the given system to explore their associations and identify several themes to discuss in the article.

In this way, 30 distinct tasks were defined. However, three were removed because in each of these tasks, the number of associations of length 1–4 (which was the setting for the systems in the experiment) found between the two entities was less than one hundred, making the task not very challenging; and one was removed because the two entities belonged to different classes, making this task inconsistent with the others. Finally, the remaining 26 tasks[3] were to be used in the user study, one of which was specifically for tutorials.

[2] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/3/dbpedia-train.xml`
[3] `http://ws.nju.edu.cn/explass/tasks.txt`

Table 2. Pre-task Questions and Responses about Exploration Context

Question	Response: Mean (SD)			$F(2, 38)$
	Explass	RelClus	RF	(p-value)
I think this task is difficult.	3.30 (1.22)	3.80 (0.77)	3.35 (1.27)	2.372 (0.107)
I'm familiar with the domain of this task.	1.75 (0.97)	1.30 (0.47)	2.00 (1.08)	2.684 (0.081)

7.3 Participant Approaches

We (re-)implemented three association exploration approaches over DBpedia to be compared in the user study: Explass, RelClus [19], and RF (based on [10]).

In all these systems, subjects started with two entity names, which were then mapped to two entities by the autocomplete functionality. Of each length from 1 to 4, up to one thousand associations between the two entities were found. When presenting them, each entity involved was accompanied by its image (if available, as illustrated in Fig. 1), and hovering the mouse over an entity activated a pop-up showing its textual description. Both images and pop-ups were to help subjects quickly understand entities and thus understand associations.

However, these systems organized associations in different ways, and supported different sets of exploration operations.

- Explass, as described in this paper, recommended a total of up to 10 patterns (giving priority to patterns of a short length), and up to 10 classes and 10 relations as facet values. We set τ, μ_C, μ_R to $0.1, 0.7, 0.7$, respectively.
- RelClus [19] organized associations inclusively into an expandable/collapsible hierarchy of clusters for refocusing. Each cluster was labeled with a unique pattern matched by all the associations in the cluster.
- RF reproduced the core feature of RelFinder [10], namely faceted association exploration. However, we did not reproduce the visualization technique adopted by RelFinder in order to make it comparable with the other two systems. Besides, in order to be comparable with Explass, RF also recommended up to 10 classes and 10 relations according to Sect. 6, and the parameters were set to the same values as in Explass.

7.4 Procedure

Subjects were instructed not to use their prior knowledge of the tasks, and they were not permitted to use tools other than the given system. Each subject carried out two random tasks using each of the three systems arranged in random order, and all the six tasks were different. Before using each system, a tutorial was given to demonstrate its functionality. The subject was then given the first task as a warmup. After that, she was given the second task and responded to two pre-task questions in Table 2 about exploration context. She had to complete this

Table 3. Post-task Questions and Responses about Exploration Effectiveness

Question	Response: Mean (SD)			$F(2,38)$ (p-value)	LSD post-hoc ($p < 0.05$)
	Explass	RelClus	RF		
Q1: The system helped me get an overview of all the information.	4.25 (0.85)	3.80 (0.77)	3.05 (0.94)	14.989 (0.000)	Explass, RelClus > RF
Q2: The system helped me easily find information relevant to this task.	4.30 (0.57)	3.25 (0.79)	3.15 (0.99)	18.769 (0.000)	Explass > RelClus, RF
Q3: The system helped me easily compare and synthesize all kinds of relevant information.	4.00 (0.86)	3.25 (0.85)	2.60 (0.99)	14.901 (0.000)	Explass > RelClus > RF
Q4: The system provided me with much support for carrying out this task.	4.10 (0.72)	3.45 (0.94)	2.85 (0.88)	16.172 (0.000)	Explass > RelClus > RF
Q5: The system provided me with sufficient support for carrying out this task.[a]	3.85 (0.88)	3.20 (1.11)	2.65 (0.75)	11.636 (0.000)	Explass > RelClus, RF

[a] Different from Q4, this question targets the functions that are expected but missing.

task in ten minutes, during which all her operations were recorded. Finally, she responded to five post-task questions in Table 3 about exploration effectiveness (which were inspired by [4]), responded to the widely-used system usability scale (SUS), and commented on the system. Questions were responded using a five-point Likert item from 1 for strongly disagree to 5 for strongly agree.

7.5 Results and Discussion

Exploration Context. Pre-task questions in Table 2 capture subject-perceived task difficulty and domain familiarity. Repeated measures ANOVA revealed that the differences in subjects' mean ratings with different systems were not statistically significant ($p > 0.05$), which supported that tasks were carried out with different systems in comparable contexts in terms of task difficulty and domain familiarity. So these two factors can be excluded from the following discussion.

User Experience. Post-task questions Q1–Q5 in Table 3 capture subjects' exploration experience with different systems. Repeated measures ANOVA revealed that the differences in subjects' mean ratings were all statistically significant ($p < 0.01$). LSD post-hoc tests ($p < 0.05$) revealed that, according to Q1, Explass and RelClus provided a better overview of all the associations than RF due to the use of patterns. According to Q2 and Q3, compared with RF and

Table 4. SUS Scores

Mean (SD)			$F(2, 38)$	LSD post-hoc
Explass	RelClus	RF	(p-value)	($p < 0.05$)
76.13 (12.53)	68.00 (17.93)	62.75 (14.93)	9.062 (0.001)	Explass > RelClus, RF

Table 5. Average Number of Exploration Operations Performed per Task

Operation	Explass	RelClus	RF
Refocusing by expanding or collapsing a pattern	9.55	19.60	n/a
Refining the search by a pattern filter or canceling it	0.35	n/a	n/a
Refining the search by a facet value filter or canceling it	5.35	n/a	9.60

RelClus, Explass helped subjects more easily find, compare, and synthesize associations by using frequent, informative, and small-overlapping patterns and facet values. Finally, according to Q4 and Q5, Explass provided subjects with more comprehensive support for exploring associations than RF and RelClus.

Table 4 summarizes SUS scores of different systems. Repeated measures ANOVA revealed that the difference in SUS score was statistically significant ($p < 0.01$). LSD post-hoc tests ($p < 0.05$) revealed that Explass was more usable than RF and RelClus.

User Behavior. Table 5 summarizes the average number of exploration operations performed per task on different systems. On Explass, both patterns and facets were frequently used, indicating that they were notably complementary in terms of usage. However, patterns were mostly used to refocus but rarely used to refine the search. Besides, compared with RelClus whose hierarchical organization of patterns needed to be explored step by step, fewer pattern operations were performed on Explass mainly due to its flat organization of patterns.

User Feedback and Discussion. We summarized all the major comments that were made by at least five subjects. On RelClus, 6 subjects (30%) said a hierarchy of clusters labeled with patterns provided a good overview of all the associations and helped refocus on a particular theme, but 11 subjects (55%) said patterns at a high level were often too general to be useful, and they were often confused about the deep and complicated hierarchies. On RF, 5 subjects (25%) said recommended classes and relations were useful filters, but 8 subjects (40%) said they needed a better overview for summarizing associations. On Explass, 14 subjects (70%) said recommended patterns provided a good summary of associations and helped refocus on a particular theme when recommended facet values helped filter associations, but 11 subjects (55%) said some very large clusters could be divided into small ones.

These comments were consistent with subjects' experience and behavior reported previously. All of these collectively supported our hypotheses H1 and H2.

- As to H1, Explass better leveraged patterns than RelClus because, firstly, RelClus may provide a deep and complicated hierarchy of patterns, whereas Explass recommended a size-controllable flat list (top-K) of patterns. Secondly, RelClus may provide very general and meaningless patterns, whereas Explass considered the informativeness of patterns in recommendation.
- As to H2, patterns and facets were complementary because frequent, informative, and small-overlapping patterns provided an overview that meaningfully summarized significant subsets of associations covering diverse themes to be refocused on, when facets provided useful filters for refining the search.

8 Conclusion and Future Work

We have realized exploratory association search in a new way by recommending top-K patterns and facet values, which have been shown to be notably complementary in terms of usage: patterns for summarizing and refocusing, and facets for refining and filtering. Compared with RelClus, our Explass provides a flat list (top-K) of clusters, which avoids deep and complicated hierarchies as on RelClus but sometimes produces very large clusters. Whereas such a large cluster could be divided into small ones by using this pattern as a filter to refine the search and obtaining its subclusters, such an operation was rarely performed by subjects in the user study, indicating that our design of user interface still needs to be carefully improved. In the future, we will also extend the notion of pattern to support the exploration of associations between more than two entities, or more generally, the entire entity-relation graph.

To recommend appropriate patterns and facet values, our novel solution has considered their frequency, informativeness, and overlap, and has exploited ontological semantics, query context, and information theory. Though it was proposed to deal with associations, the solution or its components may also be applied to recommend facet values for entity search. In the future, we will compare it with existing methods in this direction.

Acknowledgments. The authors would like to thank all the participants and reviewers. This work was supported in part by the NSFC under Grant 61100040, 61223003, and 61170068, and in part by the JSNSF under Grant BK2012723.

References

1. Aleman-Meza, B., Halaschek-Wiener, C., Arpinar, I.B., Ramakrishnan, C., Sheth, A.P.: Ranking Complex Relationships on the Semantic Web. IEEE Internet Comput. 9(3), 37–44 (2005)
2. Anyanwu, K., Maduko, A., Sheth, A.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web. In: 14th International Conference on World Wide Web, pp. 117–127. ACM, New York (2005)

3. Anyanwu, K., Sheth, A.: ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In: 12th International Conference on World Wide Web, pp. 690–699. ACM, New York (2003)
4. Arguello, J., Wu, W.-C., Kelly, D., Edwards, A.: Task Complexity, Vertical Display and User Interaction in Aggregated Search. In: 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 435–444. ACM, New York (2012)
5. Chen, N., Prasanna, V.K.: Learning to Rank Complex Semantic Relationships. Int'l J. Semant. Web Inf. Syst. 8(4), 1–19 (2012)
6. Fang, L., Das Sarma, A., Yu, C., Bohannon, P.: REX: Explaining Relationships between Entity Pairs. Proc. VLDB Endowment 5(3), 241–252 (2011)
7. Gubichev, A., Neumann, T.: Path Query Processing on Very Large RDF Graphs. In: 14th International Workshop on the Web and Databases (2011)
8. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann, Waltham (2011)
9. Hearst, M.A.: Clustering versus Faceted Categories for Information Exploration. Commun. ACM 49(4), 59–61 (2006)
10. Heim, P., Lohmann, S., Stegemann, T.: Interactive Relationship Discovery via the Semantic Web. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 303–317. Springer, Heidelberg (2010)
11. Janik, M., Kochut, K.: BRAHMS: A Workbench RDF Store and High Performance Memory System for Semantic Association Discovery. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 431–445. Springer, Heidelberg (2005)
12. Kasneci, G., Elbassuoni, S., Weikum, G.: MING: Mining Informative Entity Relationship Subgraphs. In: 18th ACM Conference on Information and Knowledge Management, pp. 1653–1656. ACM, New York (2009)
13. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer, Heidelberg (2004)
14. Lee, J., Hwang, S.-W., Nie, Z., Wen, J.-R.: Query Result Clustering for Object-level Search. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1205–1214. ACM, New York (2009)
15. Luo, G., Tang, C., Tian, Y.-L.: Answering Relationship Queries on the Web. In: 16th International Conference on World Wide Web, pp. 561–570. ACM, New York (2007)
16. Marchionini, G.: Exploratory Search: From Finding to Understanding. Commun. ACM 49(4), 41–46 (2006)
17. Oren, E., Delbru, R., Decker, S.: Extending Faceted Navigation for RDF Data. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 559–572. Springer, Heidelberg (2006)
18. Zaki, M.J., Hsiao, C.-J.: CHARM: An Efficient Algorithm for Closed Itemset Mining. In: 2nd SIAM International Conference on Data Mining, pp. 457–473. SIAM, Philadelphia (2002)
19. Zhang, Y., Cheng, G., Qu, Y.: Towards Exploratory Relationship Search: A Clustering-based Approach. In: Kim, W., Ding, Y., Kim, H.-G. (eds.) JIST 2013. LNCS, vol. 8388, pp. 277–293. Springer, Heidelberg (2014)
20. Zhou, M., Pan, Y., Wu, Y.: Conkar: Constraint Keyword-based Association Discovery. In: 20th ACM International Conference on Information and Knowledge Management, pp. 2553–2556. ACM, New York (2011)