Facilitating Entity Navigation Through Top-K Link Patterns

Liang Zheng, Yuzhong Qu(✉), Jidong Jiang, and Gong Cheng

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, People's Republic of China
{zhengliang,jdjiang}@smail.nju.edu.cn, {yzqu,gcheng}@nju.edu.cn

Abstract. Entity navigation over Linked Data often follows semantic links by using Linked Data browsers. With the increasing volume of Linked Data, the rich and diverse links make it difficult for users to traverse the link graph and find target entities. Besides, there is a necessity for navigation paradigm to take into account not only single-entity-oriented transition, but also entity-set-oriented transition. To facilitate entity navigation, we propose a novel concept called link pattern, and introduce link pattern lattice to organize semantic links when browsing an entity or a set of entities. Furthermore, to help users quickly find target entities, top-K link patterns are selected for entity navigation. The proposed approach is implemented in a prototype system and then compared with two Linked Data browsers via a user study. Experimental results show that our approach is effective.

Keywords: Entity navigation · Link pattern · Formal concept analysis · Link pattern selection

1 Introduction

With the advent of Linked Data, its navigational feature has been largely recognized during its use in practice. Just as traditional Web browsers allow users to navigate between HTML pages by following hypertext links, Linked Data browsers [1,8–11,15] allow users to navigate between entities by following semantic links. However, with the enrichment of available Linked Data on the Web, challenges in navigating the data space arise: large numbers of linked entities and high diversity of links among entities. For example, `Steven Spielberg` in DBpedia [2] is linked to 117 entities (e.g., `Cincinnati`, `Los Angeles`) through 51 semantic links (e.g., `birthPlace`, `residence`). Relying solely on link traversal, users would have to browse and choose among a potentially long list of semantic links, and synthesize information by themselves. This procedure is often time-consuming and error-prone.

Besides, there is a necessity for navigation paradigm to take into account not only single-entity-oriented transition, but also entity-set-oriented transition. Existing solutions allow users to navigate over Linked Data through common

M. Arenas et al. (Eds.): ISWC 2015, Part I, LNCS 9366, pp. 163–179, 2015.
DOI: 10.1007/978-3-319-25007-6_10

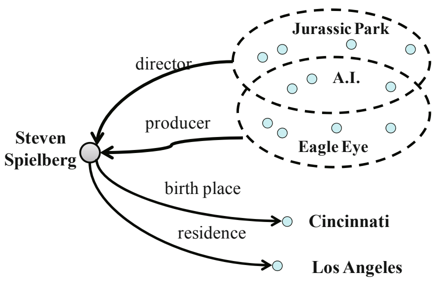

Fig. 1. The context of browsing an entity

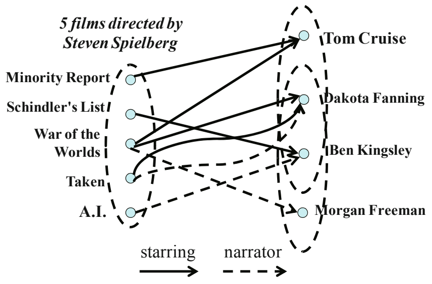

Fig. 2. The context of browsing a set of entities

links [10,11]. Yet, there are many potential relationships between the current entity (entities) and its (their) related entities. As shown in Figure 1, `Steven Spielberg` is the producer and also the director of `A.I.`. As shown in Figure 2, `Tom Cruise` starred in 2 films directed by `Steven Spielberg`. Moreover, there is a hierarchical relationship among semantic links (e.g., both `residence` and `birthplace` are subproperties of `location`). These rich structural features could be leveraged to improve entity navigation.

In order to mitigate the effect of these problems and improve the efficiency of navigation, we propose a novel approach that facilitates link traversal and assists users' navigation. In our approach, a link pattern lattice is constructed to organize semantic links based on Formal Concept Analysis (FCA) [7], a methodology of data analysis and knowledge representation. Here, link pattern represents the rich semantic relationships between the current entity (entities) and its (their) related entities (e.g., "starred at least k films", "direct and also produce"). Further, there could be an inclusion relationship between link patterns. The link pattern lattice provides a visual navigation method to explore the information space [5,6]. However, users' direct interaction with the complex lattice could cause the problem of disorientation and cognitive overhead. To lighten users' navigational burden, we give a method to select top–K link patterns for entity navigation based on the Budgeted Maximum Coverage (BMC) model [12]. The contribution of this paper is summarized as follows.

– We present a novel way to organize semantic links. We propose a new notion of link pattern and give a way to construct the link pattern lattice in the context of entity browsing.
– We introduce a measure of the "goodness" of link pattern, and give a method to select top–K link patterns based on the Budgeted Maximum Coverage (BMC) model.
– We implement the proposed approach in a prototype system and compare it with two Linked Data browsers by conducting a user study. The experimental results demonstrate the effectiveness of our approach.

The remainder of this paper is structured as follows. Section 2 discusses related work. Section 3 introduces the notion of link pattern and a way to construct the link pattern lattice in the context of browsing entities. Section 4 describes an optimization method for link pattern selection. Our evaluation is reported in Section 5. Section 6 concludes this paper.

2 Related Work

Navigation as an important feature of Linked Data, has been supported by many Linked Data browsers. Tabulator [1] allows users to browse data by starting from a single resource and following links to other resources. It also allows users to select a resource for further exploration in a nest tree view. gFacet [10] is a tool that supports the exploration of the Web of data by combining graph-based visualization with faceted filtering functionalities. With gFacet it is possible to choose one class and then pivot to a related class keeping those filters for the instances of the second class connected to the filtered instances in the first class. OpenLink Faceted Search & Find Service[1], offers several paths of DBpedia data exploration, starting from Keyword, URI or Label. It represents metadata by an entity-attribute-value view. It also provides a facet filter view by selecting different attributes. Parallax [11] is one of the first browsers to offer pivoting (or set-oriented browsing) but it is originally tied to Freebase. It shows the set of resources, accompanied by a list of facets for filtering. It also provides a list of connections, showing those properties that can be used in a pivoting operation. VisiNav [9] is a system based on an interaction model designed to easily search and navigate large amounts of Web data. It provides four atomic operations over object structured datasets: keyword search, object focus, path traversal, and facet specification. Users incrementally assemble complex queries that yield sets of objects. Rhizomer [8] addresses the exploration of semantic data by applying the data analysis mantra of overview, zoom and filter. Users can interactively explore the data using facets. Moreover, facets also feature a pivoting operation. Visor [15] is a generic RDF data explorer that can work over SPARQL endpoints. In Visor, exploration starts by selecting a class of interest from the ontology. Then, users can pivot to related collections and continue browsing. Visor provides a hierarchical overview of the collections and also provides a spreadsheet requiring manual customization to filter the collection.

Whereas the above efforts mainly focus on providing the user with powerful interaction modes, we aim to appropriately organize and select links, which is complementary to all of them.

3 Link Pattern Lattice Construction

This section introduces link pattern lattice for organizing semantic links based on Formal Concept Analysis (FCA) [7]. First, we formally define the notion of link pattern in Section 3.1 and introduce FCA in Section 3.2. Then we construct link pattern lattice in our context in Section 3.3.

[1] http://dbpedia.org/fct/

3.1 Link Pattern

Let U be a set of URI named entities and L be a set of links including object properties, property chains and inverse of them. In the implementation of this study, we only consider those links that directly connect entities or indirectly connect entities through blank nodes. A link graph $T \subseteq U \times L \times U$ is a set of triples. There is a partial ordering $\preceq$ on L, which is deduced from `rdfs:subPropertyOf` relationship.

Definition 1 (Link Pattern with Minimum Number Restriction). *Let T be a link graph, k be a positive integer, $l \in L$. A link pattern of l with minimum k restriction, denoted by LP((min k), l), is a function from 2^U to 2^U such that*
$LP((min\ k), l)(S) = \{v \in U \| \{u \in S | (u, l, v) \in T\}| \geq k\}$ for $S \subseteq U$.

The link pattern $LP((min\ k), l)$ is proposed to express the degree of connection between current entities and target entities. For simplicity, we use $(min\ k)l$ to denote $LP((min\ k), l)$. Note that $(min\ 1)l$ represents the same meaning as the traditional link l. We abbreviate $(min\ 1)l$ to l.

In Figure 1, $E_1 = \{$`Steven Spielberg`$\}$. $director^{-1}(E_1) \supseteq \{$`A.I.`, `Jurassic Park`$\}$.[2] $producer^{-1}(E_1) \supseteq \{$`A.I.`, `Eagle Eye`$\}$. In Figure 2, $E_2 = \{$`War of the Worlds`, `Taken`, `A.I.`, `Minority Report`, `Schindler's List`$\}$, S=$\{$`War of the Worlds`, `Taken`, `A.I.`$\} \subseteq E_2$, $narrator(S)$=$\{$`Dakota Fanning`, `Ben Kingsley`, `Morgan Freeman`$\}$. $((min\ 2)starring)(S) = \{$`Dakota Fanning`$\}$, which represents that `Dakota Fanning` starred at least 2 films in S.

Definition 2 (Conjunctive Link Pattern). *Given two link patterns LP_1 and LP_2, the conjunctive link pattern of LP_1 and LP_2, denoted by $LP_1 \wedge LP_2$, is a function from 2^U to 2^U such that*
$(LP_1 \wedge LP_2)(S) = LP_1(S) \cap LP_2(S)$ for $S \subseteq U$.

In Figure 1, $(director^{-1} \wedge producer^{-1})(E_1) \supseteq \{$`A.I.`$\}$, which represents that `Steven Spielberg` is the producer and also the director of `A.I.`. In Figure 2, $(narrator \wedge ((min\ 2)starring))(S)$=$\{$`Dakota Fanning`$\}$, which represents `Dakota Fanning` narrated at least 1 film and also starred at least 2 films in S.

In this paper, a link pattern can be a link pattern with minimum number restriction or a conjunctive link pattern. Besides, link patterns with minimum number restriction can be called atomic link patterns.

Definition 3 (Sub-pattern Relationship). *Given two link patterns LP_1 and LP_2, LP_1 is called a sub-pattern of LP_2, denoted by $LP_1 \subseteq LP_2$, if $LP_1(S) \subseteq LP_2(S)$ holds for every subset S of U.*

We have the following proposition, the proof of which can be easily obtained from the definition of `sub-pattern` and the inference rule for `rdfs:subPropertyOf`.

[2] We use l^{-1} to denote the inverse of link l.

Proposition 1. *Let l, l_1, $l_2 \in L$, k, k_1, $k_2 \in Z^+$, and then we have*

1. *if $k_1 \leq k_2$, then $(min\ k_2)l \subseteq (min\ k_1)l$.*
2. *if $l_1 \preceq l_2$, then $(min\ k)l_1 \subseteq (min\ k)l_2$.*

3.2 Formal Concept Analysis

In FCA [7], there are three main concepts: formal context, formal concept and concept lattice.

Definition 4 (Formal Context *K*). *A formal context is a triple K=(G, M, I), where G denotes a set of objects, M a set of attributes, and $I \subseteq G \times M$ a binary relation between G and M. The statement $(g, m) \in I$ is interpreted as "the object g has attribute m". The two derivation operators $(\cdot)^{'}$ define a Galois connection between the powersets $(2^G, \subseteq)$ and $(2^M, \subseteq)$: $A^{'} = \{m \in M \mid \forall\ g \in A : (g, m) \in I\}$ for $A \subseteq G$, and $B^{'} = \{g \in G \mid \forall\ m \in B : (g, m) \in I\}$ for $B \subseteq M$.*

Definition 5 (Formal Concept *c*). *Given a formal context K=(G, M, I) and $A \subseteq G$, $B \subseteq M$, a pair c= (A, B) satisfying $A^{'} = B$ and $B^{'} = A$, is called a formal concept of K. A and B are called the extent and intent of c, respectively.*

A partial ordering $\preceq$ over the concepts C of K can be defined as follows: (A_1, B_1) $\preceq$ (A_2, B_2) $\Longleftrightarrow A_1 \subseteq A_2$ ($\Longleftrightarrow B_2 \subseteq B_1$).

For two concepts c_1 and c_2, if $c_1 \preceq c_2$ and there is no concept c_3 with $c_3 \neq c_1$, $c_3 \neq c_2$, $c_1 \preceq c_3 \preceq c_2$, then c_1 is called a child of c_2, and c_2 is called a parent of c_1. This relationship is denoted by $c_1 \prec c_2$.

Definition 6 (Concept Lattice *L*). *With respect to a formal context K and the partial order $\prec$, the concepts in C constitute a lattice, called the concept lattice of K.*

3.3 Link Pattern Lattice Construction Using FCA

FCA is a mathematically well founded classification framework allowing to derive implicit relationships from a set of objects and their attributes. We construct link pattern lattice by using FCA. The construction process includes two steps: constructing a formal context K and generating a link pattern lattice of K.

Firstly, given a link graph $T \subseteq U \times L \times U$ and a set of entities $S \subseteq U$ being the focus, we consider the set of links $L' = \{l \in L | \exists u \in S, \exists v \in U, (u, l, v) \in T\}$. A formal context $K = (G, M, I)$ in FCA can be defined as follows: $G = \{v \in U | \exists u \in S, \exists l \in L, (u, l, v) \in T\}$ denotes the set of linked entities. M is a subset of atomic link patterns, i.e., the attributes in M take the form $(min\ k)l$. $I \subseteq G \times M$, $(v, (min\ k)l) \in I$ iff $v \in ((min\ k)l)(S)$, which means there are at least k entities in S having link l to v. The algorithm for constructing a formal context K is in Algorithm 1. Line 3 generates G and line 8-14 generate M and I.

Secondly, we choose a well-known lattice generation algorithm called Bordat [3], which produces both the concepts and the concept lattice. The worst-case running time of Bordat is $O(|G||M|^2|N|)$, where $|N|$ is the number of link patterns in the resulting lattice. We will show the running time on real-life data in our experiments.

With the following examples, we illustrate how to use FCA to construct link pattern lattice in two cases: single-entity-oriented and entity-set-oriented transitions.

Algorithm 1. Construct Formal Context

Input: T: a link graph; S: a set of entities
Output: K: a formal context

Initialize a formal context $K = (G, M, I)$, $G \leftarrow \emptyset$, $M \leftarrow \emptyset$, $I \leftarrow \emptyset$;
$L' \leftarrow \{l \in L | \exists u \in S, \exists v \in U, (u, l, v) \in T\}$;
$G \leftarrow \{v \in U | \exists u \in S, \exists l \in L, (u, l, v) \in T\}$;
foreach $l \in L'$ **do**
  Find each sup-link l_{sup} of l;
  if $l_{sup} \notin L'$ **then**
    $L' \leftarrow L' \cup \{l_{sup}\}$, $T \leftarrow T \cup \{(u, l_{sup}, v) | \exists u \in S, \exists v \in U, (u, l, v) \in T\}$;
foreach $l \in L'$ **do**
  foreach $v \in G$ **do**
    $k = |\{u \in S | (u, l, v) \in T\}|$;
    if $k > 0$ **then**
      for $i \leftarrow 1$ **to** k **do**
        $M \leftarrow M \cup \{(min\ i)l\}$;
        $I \leftarrow I \cup \{(v, (min\ i)l)\}$.
return K;

Single-Entity-Oriented. Suppose a user is viewing the RDF description of `Steven Spielberg`, as shown in Figure 1. In this case $E_1 = \{$`Steven Spielberg`$\}$ be the focus, the linked entity set $G = \{$`A.I.`, `Jurassic Park`, `Eagle Eye`, `Cincinnati`, `Los Angeles`$\}$, and the semantic links $L' = \{director^{-1}, producer^{-1}, birthPlace, residence\}$. Moreover, there is a `subLinkOf` hierarchy among these links in Figure 3. The link $participator^{-1}$ and $location$ are added to L'.

For each link $l \in L'$, we obtain the link patterns of l with minimum k restriction. k is equal to 1 in the single-entity-oriented transition. The atomic link patterns $M = \{director^{-1}, producer^{-1}, participator^{-1}, birthPlace, residence, location\}$. The formal context K is shown in Table 1.

We have $\{director^{-1}, producer^{-1}, participator^{-1}\}' = \{$`A.I.`$\}$ and $\{$`A.I.`$\}' = \{director^{-1}, producer^{-1}, participator^{-1}\}$. $\{$`A.I.`$\}$ and $\{director^{-1}, producer^{-1}, participator^{-1}\}$ satisfy a Galois connection. According to Definition 5, $(\{$`A.I.`$\}, \{director^{-1}, producer^{-1}, participator^{-1}\})$ is a formal concept emerging from Table 1. Its intent $\{director^{-1}, producer^{-1}, participator^{-1}\}$ represents a conjunctive link pattern.

Figure 4 shows the link pattern lattice associated with Table 1. In the diagram, each node denotes a link pattern while edges reflect the partial ordering $\prec$ between link patterns.

Table 1. An example of formal context

	$director^{-1}$	$producer^{-1}$	$participator^{-1}$	$birthPlace$	$residence$	$location$
A.I.	$\times$	$\times$	$\times$			
Jurassic Park	$\times$		$\times$			
Eagle Eye		$\times$	$\times$			
Cincinnati				$\times$		$\times$
Los Angeles					$\times$	$\times$

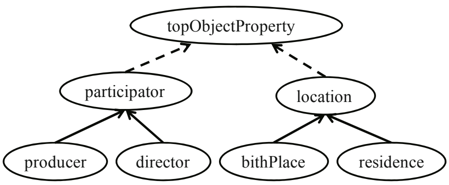

Fig. 3. An example of link hierarchy

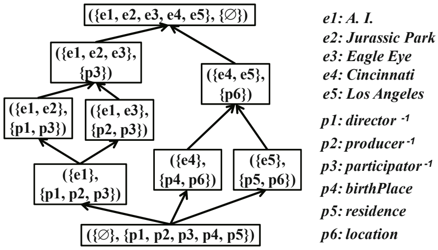

Fig. 4. Link pattern lattice associated with Table 1.

Entity-Set-Oriented. Suppose the user explores the films directed by `Steven Spielberg` by following a `director` link, as shown in Figure 2. In this case $E_2 =$ {`War of the Worlds`, `Taken`, `A.I.`, `Minority Report`, `Schindler's List`} be the focus, the semantic links $L' =$ {`starring`, `narrator`} and the linked entity set $G =$ {`Tom Cruise`, `Dakota Fanning`, `Ben Kingsley`, `Morgan Freeman`}. Note that the atomic link patterns $M = \{starring, (min\ 2)starring, narrator\}$. The formal context K is shown in Table 2.

({`Dakota Fanning`, `Ben Kingsley`}, $\{starring, narrator\}$) is a formal concept emerging from Table 2. Its intent $\{starring, narrator\}$ represents a conjunctive link pattern. Besides, ({`Dakota Fanning`}, $\{starring, (min\ 2)starring, narrator\}$) is another concept of this context. Furthermore, we have ({`Dakota Fanning`}, $\{starring, (min\ 2)starring, narrator\}$) $\prec$ ({`Dakota Fanning`, `Ben Kingsley`}, $\{starring, narrator\}$). The link pattern lattice for Table 2 is shown in Figure 5.

4 Link Pattern Selection

A link pattern lattice provides a multi-granular, progressive navigation assistance. In some cases, the lattice may have a complex structure so that users feel disoriented and require several interactions to arrive at target entities.

For lightening users' burden, we give a method to select top–K link patterns from lattice to enable users to find target entities more quickly. Firstly, we introduce three metrics to measure the "goodness" of link patterns in Section 4.1. Then we select top–K link patterns that are as "good" as possible while being able to retrieve as many linked entities as possible in Section 4.2.

Table 2. An example of formal context

	starring	(*min*2)*starring*	*narrator*
Cruise	×	×	
Fanning	×	×	×
Kingsley	×		×
Freeman			×

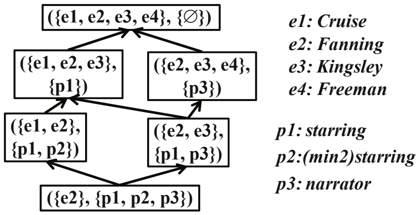

Fig. 5. Link pattern lattice for Table 2.

4.1 Metrics of Link Pattern

Given a link pattern lattice *LPL* of a formal context K = *(G, M, I)* and a link pattern c, the "goodness" of link pattern c can be defined from various perspectives. In this paper, we prefer to provide informative (measured by informativeness), understandable (measured by conciseness) and specific (measured by specificness) link patterns.

Informativeness. As to link patterns, the idea is that a link pattern having fewer reachable linked entities is more informative. We compute the self-information of the link pattern c using information theory [13],

$$info(c) = -\log pr(c), \quad pr(c) = \frac{|ext(c)|}{|G|} . \tag{1}$$

ext(c) is the extent of c. G is the set of linked entities in K. Further, we normalize $info(c)$ into the range $[0, 1]$ as the *informativeness* of link pattern c:

$$info_K(c) = \frac{info(c)}{\log |G|} . \tag{2}$$

Conciseness. In practice, we use the label of the intent of link pattern c as a "road sign" in users' navigation (e.g., $director^{-1} \wedge producer^{-1} \wedge participator^{-1}$). The longer the lengths of intents become, involving many links at various levels of generality, the harder it becomes to understand what the link patterns mean or represent.

A concise link pattern having a shorter label is more understandable and preferable. So, we formalize the conciseness of link pattern c as follows:

$$conc(c) = a^{-(|inte(c)|-1)} \quad (a > 1)\,. \tag{3}$$

inte(c) is the intent of c and the value of $conc(c)$ is in the range $(0, 1]$.

Specificness. As shown in Figure 4, the link pattern lattice *LPL* provides a hierarchy among link patterns. The depth of link pattern in the hierarchy is useful. The larger the depth of link pattern is, the more specific the link pattern is. We measure the depth of link pattern c:

$$depth(c) = distance(a, c)\,, \tag{4}$$

where *distance(a, c)* is the length of a shortest path from a (the greatest element of *LPL*) to c. Further, we normalize $depth(c)$ into the range $[0, 1]$ as the *specificness* of link pattern c:

$$\begin{aligned} spec(c) &= \frac{depth(c)}{D(c)}\,, \\ D(c) &= distance(a, c, b)\,. \end{aligned} \tag{5}$$

distance(a, c, b) is the length of a shortest path from a, through c, to b (the least element of *LPL*).

4.2 Selecting Link Patterns

For diversity and coverage considerations, we aim to select top–K link patterns that are as informative, concise and specific as possible while being able to retrieve as many linked entities as possible.

Our problem can be formalized based on the Budgeted Maximum Coverage (BMC) model [12]. The BMC problem is defined as follows: Let $S = \{S_1, S_2, \ldots, S_m\}$ be a collection of sets defined over a domain of elements $X = \{x_1, x_2, \ldots, x_n\}$. Each set has a cost $\{c_i\}_{i=1}^m$ while each element has a weight $\{w_i\}_{i=1}^n$. The goal is to find a collection of sets $S' \subseteq S$, such that the total cost of S', denoted by $c(S')$, does not exceed a given budget B, while the total weight of elements covered by S', denoted by $w(S')$, is maximized. $c(S')$ and $w(S')$ are defined as follows:

$$c(S') = \sum_{S_i \in S'} c_i\,, \tag{6}$$

$$w(S') = \sum_{j=1}^{n} \left(w_j \cdot f(x_j, S')\}\right), \tag{7}$$

where

$$f(x_j, S') = \begin{cases} 1 & \text{if } x_j \text{ is covered by } S'. \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

In our context, each link pattern can be considered as a set $S_i \in S$ and all the linked entities as the elements X. The weight of each element is trivially set to 1. The cost of S_i is defined as follows:

$$c_i = \left(\frac{1}{e}\right)^{\sigma(S_i)}. \tag{9}$$

$\sigma(S_i)$ is a scoring function of link pattern S_i as follows:

$$\sigma(S_i) = \alpha_1 \cdot info_K(S_i) + \alpha_2 \cdot conc(S_i) + \alpha_3 \cdot spec(S_i)\,, \tag{10}$$

where $\alpha_1, \alpha_2, \alpha_3 \in [0, 1]$ indicate the weights for each metric to be tuned empirically. According to Equation (9), the higher the score of a link pattern is, the less the cost is.

BMC is an NP-hard problem and several efficient approximation algorithms have been developed. By comparing the approximation ratio and the time complexity of these algorithms, we use the $\frac{1}{2} \cdot (1 - \frac{1}{e})$ approximation algorithm with time complexity $O(m^2 n)$ provided by [12] in our implementation.

5 Evaluation

In this section, we first present the frequency distribution of link patterns in two real-life datasets (Section 5.1). Then, we describe an overview of the prototype system (Section 5.2) and compare it with two Linked Data browsers by conducting a user study in section 5.3. Finally, we evaluate the performance of our approach by measuring the average execution time in section 5.4.

5.1 Data Sets

This section aims to show that *minimum number restriction* and *conjunctive* link patterns do exist widely in real-life data sets such as DBpedia[3] and Semantic Web Dog Food[4].

Data Collection. As to DBpedia, we used the DBpedia *mapping-based properties* dataset, excluding RDF triples containing literals. We selected 8 classes (i.e., Scientist, Artist, Athlete, City, River, Company, University, Film).

For each class, we firstly collected the top 1000 entities according to descending order of the number of their related entities (i.e., the degree of node in RDF graph). Secondly, we established 100 entity sets and each set included 10 entities by selecting at random from these 1000 entities. Finally, we calculated the percentage of entity sets having *minimum number restriction (k >1)* and *conjunctive* link patterns in these 100 entity sets.

As to Semantic Web Dog Food, we firstly selected 3 classes: Person (7,180 entities), Organization (1,965 entities) and Conference (20 entities). For the first two classes, we collected the top 100 entities (using the same method as above).

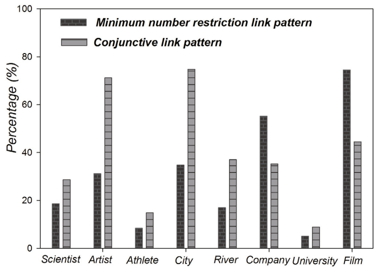

Fig. 6. Link patterns in DBpedia.

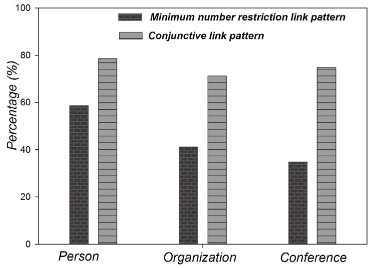

Fig. 7. Link patterns in Semantic Web Dog Food.

For Conference, we collected all the entities. Secondly, we established entity sets for each class. For Person and Organization, we established 10 entity sets and each set included 10 entities by selecting at random from these 100 entities. For Conference, we established 5 entity sets and each set included 4 entities by selecting at random from these 20 entities. The method calculating the frequency distribution of link patterns was the same as above.

Data Analysis. In Figure 6, *minimum number restriction* link patterns (e.g., $(min\ k)distributor$, $(min\ k)developer^{-1}$) are found in more than 50% of Film and Company entity sets. Artist and City have more *conjunctive* link patterns (e.g., $occupation \wedge foundedBy^{-1}$, $birthPlace \wedge residence$), which occupied 70% of entity sets.

In Figure 7, around 60% of entity sets of Person have *minimum number restriction* link patterns (e.g., $(min\ k)made$, $(min\ k)based\_near$). Every class has more *conjunctive* link patterns (e.g., $made \wedge author^{-1}$, $affiliation^{-1} \wedge member$).

In summary, we investigate entities having the largest number of linked entities in two datasets because the above link patterns are more likely to be observed there. As expected *minimum number restriction* and *conjunctive* link patterns exist widely in many classes, which can be used to improve entity navigation.

5.2 Overview of Prototype

We implemented our proposed approach as a navigation module (called "Link") in a Link Data browser, SView[5]. Figure 8 shows a screenshot of "Link" in SView.

Users can start browsing with an entity URI by entering into the input box (A). Navigation was provided in the "Link" panel. The left-hand side of the interface lists the label of link patterns (B). The right-hand side lists linked entities (C). Users can click the button "browse all" to explore all the linked entities (D). Also, users can choose some link patterns to filter the target entities (E).

[3] http://wiki.dbpedia.org/Downloads2014
[4] http://data.semanticweb.org/
[5] http://ws.nju.edu.cn/sview/

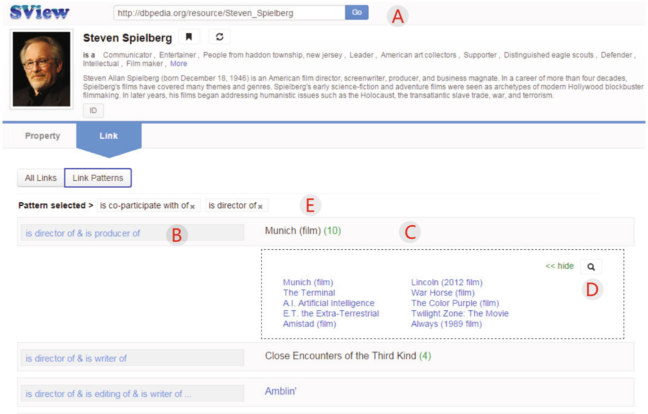

Fig. 8. A screenshot of "Link" in SView.

5.3 User Study

We conducted a user study to compare our approach with two Linked Data browsers (i.e., OpenLink Faceted Search & Find Service, Rhizomer[6]), and to evaluate the effectiveness of our approach.

Table 3. An example of navigation tasks about `Steven Spielberg`

		Tasks
G1	*E1*	Explore the information related to `Steven Spielberg`.
	F1	Find the films directed and also produced by `Steven Spielberg`.
G2	*E2*	Explore the information related to the films directed by `Steven Spielberg`.
	F2	Find the actors starred in at least 2 films directed by `Steven Spielberg`.

Participant Systems. As reviewed in the Related Work section, the only active tools capable of entity-set-oriented browsing are gFacet, Parallax, OpenLink Faceted Search & Find Service, Rhizomer and Visor. We did not include Visor and gFacet because their interfaces are based on graphs. We did not consider Parallax because it only tied to Freebase. OpenLink Faceted Search & Find Service and Rhizomer provide a user interface with HTML and components similar to those we propose.

Tasks. In a browsing scenario, navigation tasks can be divided into two types: *Explore* (a user has a fuzzy need) and *Find* (a user has a clear need) tasks [14,16]. According to navigation paradigm, tasks can also be divided into two groups: single-entity-oriented (*G1*) and entity-set-oriented (*G2*) tasks.

[6] http://rhizomik.net/html/rhizomer/

We used 8 classes of entities from DBpedia dataset in section 5.1. For each class, we selected 10 entities from the top 1000 entities at random as the starting points of user navigation. For each starting point, we established 4 navigation tasks. The navigation tasks about `Steven Spielberg` is shown in Table 3.

Procedure. The subjects consisted of 24 students majoring in computer science who were familiar with the Web, but with no knowledge of our project. The evaluation was conducted in three phases.

Table 4. Navigation questionnaire

	Questions
Q1:	The number of navigation options (links) was overwhelming.
Q2:	The navigation options (links) were well organized.
Q3:	The navigation option (link) titles were understood well.
Q4:	The navigation options (links) were pleasantly surprising.
Q5:	It was easy to reorient myself in the navigation.

Table 5. Results of navigation questionnaire

	Response: Mean (SD)			$F(2,69)$	LSD post-hoc
	OpenLink	Rhizomer	SView	(p-value)	($p < 0.05$)
Q1:	3.919 (0.717)	3.75 (0.854)	2.667 (1.095)	21.643 (0.000)	OpenLink, Rhizomer > SView
Q2:	3.026 (1.052)	3.24 (1.014)	4.11 (0.887)	13.580 (0.000)	SView > OpenLink, Rhizomer
Q3:	3.833 (0.717)	4.00 (0.582)	2.583 (0.62)	9.658 (0.000)	Rhizomer, OpenLink > SView
Q4:	2.58 (0.793)	3.25 (1.055)	4.33 (0.778)	11.958 (0.000)	SView > OpenLink, Rhizomer
Q5:	3.917 (0.514)	3.667 (0.778)	3.5 (0.937)	11.367 (0.000)	OpenLink > Rhizomer, SView

First, the subjects learned how to use the given systems through a 20 min tutorial, and had additional 10 minutes for free use and questions. Second, the subjects used each of the three systems arranged in random order. For each system, the subjects were randomly assigned to one starting point, and required to complete 4 navigation tasks. Meanwhile, the starting points of user navigation among the three systems 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.

With regard to each system, the subjects responded to the navigation questionnaire, as shown in Table 4. Then, for each system, the subjects responded to the widely-used system usability scale (SUS) questionnaire [4]. The questions in the two above questionnaires were responded by using a five-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). Finally, the subjects were asked to comment on the three systems.

Results and Discussion

User Experience. Navigation questionnaire Q1–Q5 captured subjects' navigation experience with different systems in Table 5. 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, OpenLink and Rhizomer provided too many links compared with SView. According to Q2, SView provided a better organization of links than OpenLink and Rhizomer. According to Q3, OpenLink and Rhizomer helped subjects more easily understand the label of links. According to Q4, SView directly provided subjects with more interesting relationships among the entities. Finally, according to Q5, OpenLink and Rhizomer helped subjects keep track of browsing and provided easy rollback.

Table 6 summarizes SUS scores of different systems. Repeated measures ANOVA revealed that the difference in SUS score was statistically significant ($p < 0.05$). LSD post-hoc tests ($p < 0.05$) revealed that SView was more usable than OpenLink and Rhizomer.

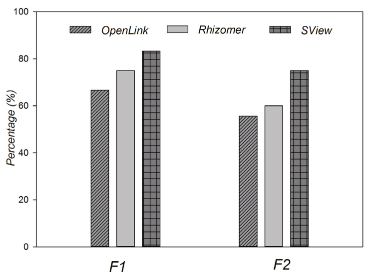

Fig. 9. Success rate of *Find* tasks.

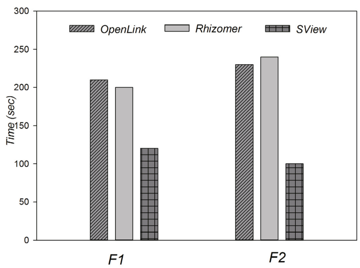

Fig. 10. Average consumption time of *Find* tasks.

Table 6. SUS scores

Mean (SD)			$F(2, 69)$	LSD post-hoc
OpenLink	Rhizomer	SView	(p-value)	($p < 0.05$)
59.62 (9.177)	67.31 (9.098)	75 (7.706)	10.195 (0.001)	SView > Rhizomer > OpenLink

User Behavior. Figure 9 shows the success rate of *Find* tasks. In F1, using SView, subjects achieved the highest overall success rate. In F2, the situation was similar. Figure 10 shows the average time spent on *Find* tasks. According to F1 and F2, using SView, subjects required far less time to complete these tasks, because links were appropriately organized and selected.

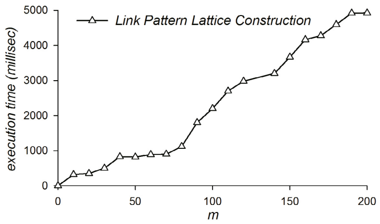

Fig. 11. Execution time of link pattern lattice construction.

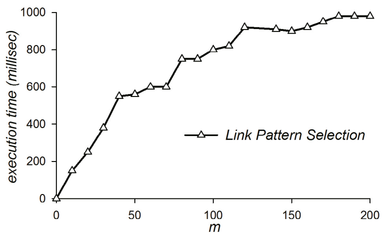

Fig. 12. Execution time of link pattern selection.

User Feedback and Discussion. We summarized all the major comments that were made by at least five subjects. On OpenLink, 21 subjects (88%) said the large quantities of links often made it difficult to retrieve the target entities. 17 subjects (71%) said the browsing track assisted them to browse intermediate entities. On Rhizomer, 20 subjects (83%) said faceted navigation helped them filter out those entities that were not interesting. On SView, 22 subjects (92%) said recommended patterns provided a fast locating mechanism, but 6 subjects (25%) said it had some potential risks, such as target losses (i.e., not covering all the needed entities). 10 subjects (42%) said the diverse link patterns made users know more potential relationships among the entities, but 8 subjects (33%) said some link pattern labels were too long to be understood.

5.4 Performance Evaluation

We evaluated the performance of link pattern lattice construction and link pattern selection by measuring the average execution time for varying number of current entities denoted by m (m from 5 to 200). The two algorithms were implemented in Java and carried out on an Intel Xeon E3 3.2GHz CPU, Windows 7 with 10GB JVM.

As can be seen from Figure 11 and Figure 12, the two algorithms were reasonably fast in practice. When m increased, the curves of the two algorithms kept ascend slowly. In Figure 11, it took 2 seconds to construct a lattice for 80 current entities, and 5 seconds for 200 current entities.

6 Conclusion

In this paper, we propose a novel concept called link pattern, in particular link pattern with minimum number restriction as well as conjunctive link pattern. It enables a new way of semantic navigation over linked entities. We also describe how to generate link pattern lattice and how to select top-K link patterns in the context of entity browsing. The proposed approach is implemented in a prototype system. The evaluation results demonstrate that link patterns effectively make explicit complex relationships among entities, and help users discover target entities more quickly.

Currently, link patterns are generated based on the "local" context, i.e. the data about current entities being visited. It is interesting to consider a way to extract link patterns from the "global" context, i.e. the Web of Data. Another future work is to study "human factors" in the context of entity navigation. For example, users' preference on link patterns can be collected and then leveraged to select patterns more intelligently.

Acknowledgments. This work is supported by the 863 Program under Grant 2015AA015406, in part by the National Science Foundation of China under Grant Nos. 61170068, and in part by National Social Science Foundation of China under Grant 11AZD121 and JSNSF under Grant BK2012723. We are also grateful to all the participants in the experiments of this work.

References

1. Berners-lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D.: Tabulator: exploring and analyzing linked data on the semantic web. In: 3rd International Semantic Web User Interaction Workshop (2006)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia-A crystallization point for the Web of Data. J. Web Sem. **7**(3), 154–165 (2009)
3. Bordat, J.P.: Calcul pratique du treillis de Galois dune correspondance. Math. Sci. Hum. **96**, 31–47 (1986)
4. Brooke, J.: SUS-A quick and dirty usability scale. Usability evaluation in industry **189**(194), 4–7 (1996)
5. Carpineto, C., Romano, G., Bordoni, F.U.: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. J. UCS. **10**(8), 985–1013 (2004)
6. Ferre, S.: Camelis: a logical information system to organise and browse a collection of documents. International Journal of General Systems **38**(4), 379–403 (2009)
7. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
8. Garcia, R., Gimeno, J.M., Perdrix, F., et al.: Building a usable and accessible semantic web interaction platform. World Wide Web **13**(1–2), 143–167 (2010)
9. Harth, A.: VisiNav: A system for visual search and navigation on web data. J. Web Sem. **8**(4), 348–354 (2010)
10. Heim, P., Ertl, T., Ziegler, J.: Facet graphs: complex semantic querying made easy. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 288–302. Springer, Heidelberg (2010)

11. Huynh, D.F., Karger, D.: Parallax and companion: set-based browsing for the data web. In: 18th International World Wide Web Conference. ACM, Madrid (2009)
12. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. Information Processing Letters **70**(1), 39–45 (1999)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2008)
14. Pandit, S., Olston, C.: Navigation aided retrieval. In: 16th International Conference on World Wide Web, pp. 391–400. ACM, New York (2007)
15. Popov, I.O., Schraefel, M.C., Hall, W., Shadbolt, N.: Connecting the dots: a multipivot approach to data exploration. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 553–568. Springer, Heidelberg (2011)
16. Wagner, A., Ladwig, G., Tran, T.: Browsing-oriented semantic faceted search. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 303–319. Springer, Heidelberg (2011)