The RelFinder User Interface: Interactive Exploration of Relationships between Objects of Interest

Steffen Lohmann  
Carlos III University of Madrid  
Leganés, Spain  
slohmann@inf.uc3m.es

Philipp Heim  
University of Stuttgart  
Stuttgart, Germany  
philipp.heim@vis.uni-stuttgart.de

Timo Stegemann  
University of Duisburg-Essen  
Duisburg, Germany  
timo.stegemann@uni-due.de

Jürgen Ziegler  
University of Duisburg-Essen  
Duisburg, Germany  
juergen.ziegler@uni-due.de

ABSTRACT

Being aware of the relationships that exist between objects of interest is crucial in many situations. The RelFinder user interface helps to get an overview: Even large amounts of relationships can be visualized, filtered, and analyzed by the user. Common concepts of knowledge representation are exploited in order to support interactive exploration both on the level of global filters and single relationships. The RelFinder is easy-to-use and works on every RDF knowledge base that provides standardized SPARQL access.

ACM Classification Keywords
H.5.2 User Interfaces: Graphical user interfaces (GUI)

General Terms
Human Factors, Management, Design

INTRODUCTION

Today’s world is complex, and so are the relationships within most knowledge domains. Especially non-obvious relationships can easily be overlooked and therefore not considered in decision making or similar processes. This might result in wrong decisions and unwanted consequences. Algorithms that detect and extract relationships between objects of interest help to reduce this risk [1]. However, such algorithms only provide benefit when combined with sophisticated user interfaces that assist in the analysis and exploration of the found relationships. We developed the RelFinder user interface in order to cope with these challenges. In the following, we describe how it supports the interactive exploration of even large amounts of relationships.

THE RELFINDER USER INTERFACE

The Adobe Flex based user interface of the RelFinder offers an interactive visualization of relationships between given objects of interest. The relationships are found by an algorithm consisting of several SPARQL queries that can be applied to arbitrary RDF knowledge bases [2]. The labels of the objects are entered in input boxes in the upper part of the sidebar (see Fig. 1). They are then mapped to unique objects of the knowledge base by executing an automatic (if possible) or manual (if not) disambiguation. These user-given objects serve as starting nodes in a graph that is drawn in the presentation area of the user interface. The starting nodes (indicated by a stronger border) are elliptically arranged in the graph visualization and are either directly connected with each other or indirectly via further objects that were found in between by the algorithm. The links between the objects are visualized as labeled and directed edges in accordance with their representation in the knowledge base1. Thus, relationships in the graph visualization follow the well-known subject-predicate-object schema making them easy to read.

The graph visualization and the sidebar offer sophisticated functionality for the interactive exploration of the found relationships. Whereas graph interaction takes place mainly on the level of single objects and relationships, the sidebar also supports the exploration on a more global level, as described in the following.

Filtering Relationships on Four Dimensions

The sidebar offers four types of filters that facilitate the exploration of the graph visualization by highlighting or removing certain elements. The first two filters are based on type information derived from the knowledge base:

• **Class Filter**: Objects in knowledge bases are typically assigned to certain classes (e.g., in RDF via rdf:type). These classes can be helpful in filtering objects that are not of interest. In Fig. 1, for instance, all objects belonging to the class GrandPrixDrivers of the YAGO ontology2 have been removed from the graph.

• **Link Filter**: Links between objects are usually of a certain type that can also be used for filtering. In Fig. 1, five links of type companyType have been removed from the graph as they were not of interest to the user.

The two remaining filters use properties that characterize the topology of relationships:

• **Length Filter**: Relationships found by the algorithm are typically of different lengths, i.e., the number of objects the relationships consist of varies. This value is used by

---

1 Actually, edge labels are also handled as nodes in the graph layout for readability reasons.

the length filter. In Fig. 1, direct relationships (i.e., those without any objects) were removed from the visualization since the user was only interested in indirect ones.

- **Connectivity Filter**: The found relationships can also be classified according to the number of user-given objects (i.e., starting nodes in the graph) they connect. In Fig. 1, only relationships that contain the object (“Repubblica Italiana”) connect all three user-given objects. The connectivity filter usually gains in importance when the number of user-given objects increases. This filter can then be used, for instance, to remove less connected relationships or to highlight those relationships that connect all user-given objects.

**Direct Exploration of the Relationship Graph**

Besides the filtering features offered in the sidebar, the graph visualization can also be directly explored on the level of single objects and relationships. On this level, the RelFinder provides the following interaction support:

- **Red Thread**: If an object node is selected, a “red thread” through the graph is drawn highlighting all relationships that contain this object (Fig. 1). That helps the user to visually track relationships within the graph.

- **Pick & Pin**: Single object nodes can be picked and repositioned, making it possible to decouple relationships from the automatic layout and analyze them separately (repositioned nodes are indicated by a needle symbol).

- **Infobox**: Additional information about the selected object (such as an image and a short description) is shown in the lower part of the sidebar. This information is gathered from the knowledge base by searching for well-known descriptors (e.g., foaf:Image or rdfs:comment).

In general, all visible objects that belong to the same class as the selected one are highlighted in the graph (by using the above-mentioned class filter). This helps to create some awareness of further objects that might be of interest to the user. In Fig. 1, for instance, another object (“Enzo Ferrari”) is highlighted that also belongs to the class Person of the DBpedia ontology.

**DISCUSSION**

The described features of the RelFinder assist users in exploring relationships between given objects of interest. The RelFinder therefore fills the gap between algorithms that find relationships within knowledge bases and their efficient usage in the context of real-world scenarios. In particular, it aims to facilitate the separation of relevant relationships from irrelevant ones and the discovery on non-obvious relationships by offering innovative visual features.

We tested the RelFinder against several RDF knowledge bases (e.g., DBpedia, LOD4, LinkedMDB5) and evaluated it in a small user study. In sum, it shows that the interface is easy-to-use even for non-experts and very suitable to get a quick overview on relationships between objects of interest. Drawbacks are long response times in some cases or many irrelevant relationships in others. However, these are rather issues of the accessed knowledge bases than the RelFinder.

The RelFinder implements several further features that cannot be detailed here, such as an automatic initial filtering of relationships or a correction of false disambiguations. The accessed knowledge bases can easily be configured and new ones can be added via a settings menu or an XML-file. The current version of the RelFinder is accessible at [3].

**REFERENCES**

3. RelFinder: http://relfinder.semanticweb.org

3 [http://wiki.dbpedia.org/Ontology](http://wiki.dbpedia.org/Ontology)
5 [http://linkedmdb.org/](http://linkedmdb.org/)