Faceted Exploring for Domain Knowledge over Linked Open Data

Meng Wang+, Jun Liu+, Wenqiang Liu+, Qinghua Zheng+, Wei Zhang++, Lingyun Song+, Siyu Yao+

+MOEKLINNS Lab, Department of Computer Science, Xi’an Jiaotong University, 710049, China
++Amazon.com Inc, Seattle, WA 98109, USA

wangmengsd@stu.xjtu.edu.cn, liukeen@mail.xjtu.edu.cn, liuwenqiangcs@stu.xjtu.edu.cn, qhzheng@mail.xjtu.edu.cn,
wzhan@amazon.com, xjtuslysm@gmail.com, cheryl@stu.xjtu.edu.cn

ABSTRACT
The rapidly increasing RDF data in the Linked Open Data (LOD) community project is a valuable resource for obtaining domain knowledge. However, RDF data of specific topics also shows a trend of being more decentralized and fragmented, which makes it difficult and inefficient for the users to get an overview of a specific topic and retrieve the desired information. In this paper, we demonstrate a novel system called KFM, which can aggregate the distributed RDF data of a topic according to the facets of this topic. KFM provides a new way for users to obtain and explore domain knowledge in the LOD cloud.

Categories and Subject Descriptors
H.3.5 [On-line Information Services]: Web-based services;
H.3.3 [Information Search and Retrieval]: Clustering;

General Terms
Algorithms, Management, Design

Keywords
RDF, linked open data, data fusion, domain knowledge

1. INTRODUCTION
As of September 2011, Linked Open Data (LOD) contains 295 datasets with more than 31 billion RDF triples from a variety of domains1. A plethora of valuable information regarding domain knowledge can be found within these data sets. But RDF data of specific topics also shows a steady trend of being more decentralized and fragmented. More specifically, when searching a specific topic via the traditional semantic search engine, the results usually distribute across a large number of datasets which are rather loosely connected. The RDF data in each dataset only covers a few facets (a facet in this study means an aspect or dimension) of the topic. We call this the RDF data fragmentation problem.

For instance, when searching k-means in Sindice[3], about 800 RDF documents, which contain about 63,000 RDF triples will be returned (as of June 23th, 2014). However, these RDF data are distributed in more than 30 datasets, and are blended into a long and daunting list. Furthermore, information about k-means in each result item includes different facets and is incomplete. For example, some result items are about the algorithm steps of k-means, whereas some only contain the references about k-means.

RDF data fragmentation problem makes it difficult for the users to get an overview of a specific topic, or efficiently retrieve their desired information. To solve this RDF data fragmentation problem, we present a novel system called KFM (Knowledge Fusion Map)2 in this demonstration. KFM fuses the distributed RDF data according to the topic’s different facets, and provides an intuitive and visual presentation for users. This differs from the traditional semantic search engine and provides a new way for users to explore the domain knowledge over the LOD. Figure 2 shows the fusion result of k-means.

KFM tries to solve three problems:
- How to determine what facets a given topic has so that we can generate a complete fusion result?
- How to fuse the extracted RDF data to the topic’s different facets?
- How to organize and visualize the fusion results to users?

The rest of this demonstration is organized as follows. Section 2 introduces several concepts about KFM. Section 3 shows the architecture of KFM and explains important technical issues. Section 4 demonstrates how KFM offers a new visual way for users to obtain domain knowledge and reports the preliminary results on three domains. Section 5 summarizes the conclusions of this demonstration.

2. PRELIMINARY

The important concepts in KFM include:

1 The LOD cloud diagram: http://lod-cloud.net/
2 http://kfm.skyclass.net
**Topic-specific RDF Graph:** The RDF data about a specific topic \( T \) is a collection of RDF triples \( Q_T \). Each RDF triple consists of a subject, a predicate, and an object. \( Q_T \) can also be represented as a graph, called topic-specific RDF graph (TRG). TRG can be formally defined as \( Q_T=\{V, E, P\} \). \( V \) refers to the vertices in the graph, it is a set of subjects or objects; \( E \subset V \times V \) represents a set of directed edges, with each edge corresponding to a triple in \( Q_T \); \( P \) is a set of predicates and indicates the labels of edges.

**Facet Template:** The RDF data of a specific topic often contains diverse facets. Every facet has a facet label (such as the *Feature of k-means*) which summarizes the meaning of RDF data it represents. A facet template describes the facets of a topic \( T \) and can be formally defined as \( FT_T=\{l_i\}_m \), where \( l_i \) denotes a label to represent a topic’s facet. With an example shown in Figure 1, the facet template of \( k\)-means is \{Reference, Same as, Feature, Overview, Algorithm step, See also, Type, Application\}.

**Knowledge Fusion Map:** A knowledge fusion map is the fusion result for a given topic \( T \), which can be formally defined as \( KFM_T = \{l_i, F_i\}_m \), where \( l_i \in FT_T, F_i \subset Q_T \). \( F_i \) corresponds to the related facet and satisfy \( \forall x, y \in [1 \ldots n], x \neq y; F_x \cap F_y = \emptyset \).

### 3. The KFM SYSTEM

#### 3.1 Architecture

For a certain domain, KFM generates knowledge fusion maps for specific topics by employing a three-module framework, as shown in Figure 2.

![Figure 2. Architecture of KFM](Image)

Module I generates a facet template for a group of similar topics from DBpedia and Wikipedia.

Module II crawls RDF data of a specific topic from LOD and fuses them based on two topological properties of TRG.

Module III assigns fused RDF data to corresponding facets and generates a knowledge fusion map.

#### 3.2 Facet Template Generation

Similar topics usually have similar facets. For example, \( k\)-means and \( k\)-medoid are algorithms in Data mining. They have common facets such as *Algorithm step, Feature, Application*. So generating facet template can be divided into two subtasks: clustering topics and generating facet labels for every cluster.

To generate topic clusters, KFM uses DBpedia as the initial dataset because DBpedia encompasses numerous domains and almost any topic can be found there. Each topic corresponds to an RDF entity in DBpedia (in this section, we use the term of topic and entity interchangeably). By analyzing the RDF entities, we find two features that can be used for clustering: predicate similarity and category relatedness.

**Predicate Similarity:** Similar RDF entities in DBpedia usually have similar predicates. KFM compresses the predicates of each entity into a bag of words and represents each entity with a word vector based on TF-IDF[1]. The predicate similarity \( ps(e_i, e_j) \) of two entities \( e_i, e_j \) can be further defined as the cosine of the angle between the two vectors.

**Category Relatedness:** Each RDF entity usually links to an arbitrary number of categories. Categories in DBpedia are connected by *skos:broader* relations (“a-type-of” relationship)\(^5\). For example, *Array is skos:broader of Data structure*. Given two RDF entities \( e_i \) and \( e_j \), KFM determines the category relatedness value \( cr(e_i, e_j) \) by using the method proposed in [5], which is based on the topological properties of category graph.

With these two features, the similarity between RDF entity \( e_i \) and \( e_j \) can be defined as:

\[
sim(e_i, e_j) = \lambda ps(e_i, e_j) + (1 - \lambda)cr(e_i, e_j),
\]

where \( \lambda \) is an empirical value. KFM uses the similarity to cluster the entities with an agglomerative hierarchical clustering algorithm. Preliminary result using 6 datasets shows that KFM achieves a best average clustering precision of 89.6% at a recall of 84.7% when \( \lambda \) is set to 0.3.

To generate facet labels, we find each DBpedia entity contains an RDF triple to link its article in Wikipedia. The *Contents table* and *Infobox* in a Wikipedia article provide a set of facet labels for the topic. These candidate labels have two problems, schema duplication and sparsity. As an example of schema duplication, *abstract and overview* are used to describe the same meaning of a facet. Sparsity means many labels are used very rarely. KFM currently uses a statistical method to extract representative labels for each template from Wikipedia articles.

#### 3.3 Faceted Fusion

We find that two topological properties of TRG serve as the foundation in the module II, namely homogeneity of adjacent vertices and homogeneity of similar predicate vertices.

**Homogeneity of adjacent vertices:** By scrutinizing the degree distribution of the TRGs, we find that the average degree \( k \) is between 2 and 3.5. Around 98% of the vertices in each TRG are isolated. This indicates that a TRG is a highly-sparse graph. We find that two connected vertices are very likely belonging to the same facet, by focusing on the vertices that are connected by links in a TRG.

**Homogeneity of similar predicate vertices:** We also find that the Jaccard similarity score between two vertices is associated with the facet of the topic. The similarity score \( s_{v_i, v_j} \) of vertices \( v_i \) and \( v_j \) can be computed based on their shared predicates as follows.

\[
s_{v_i, v_j} = \frac{|c(v_i, v_j)|}{|p(v_i)| + |p(v_j)| - |c(v_i, v_j)|}
\]

---

3 http://wiki.dbpedia.org/Datasets
4 http://en.wikipedia.org/
5 http://www.w3.org/2009/08/skos-reference/skos.html#broader
where \( p(v_i) \) and \( p(v_j) \) are the sets of predicates for vertices \( v_i \) and \( v_j \). \( c(v_i, v_j) \) represents the set of common predicates/objects shared by \( v_i \) and \( v_j \). By analyzing the similarity of any two distinct vertices in each TRG from six different domains, we find that the probability of two vertices belonging to the same facet increases with their similarity. It indicates that the higher the similarity, the more likely the two vertices belong to the same facet.

For a specific topic, the procedure of fusing RDF data includes: (1) KFM uses LDspider [2] to crawl its RDF data returned by Sindice; (2) KFM generates a set of fused facets by leveraging the homogeneity of similar predicate vertices; (3) KFM merges these facets based on the homogeneity of adjacent vertices, and outputs the final fused RDF data.

### 3.4 Knowledge Fusion Map Generation

The fused RDF data generated by Module II are not corresponding to the facet labels of the facet template. So KFM automatically assigns these fused results to the template items and generates a knowledge fusion map. This can be seen as a multi-class classification problem.

Inspired by [4], KFM automatically extracts the RDF data from DBpedia entity, their corresponding faceted labels from Wikipedia and heuristically labels the training sets. This makes KFM robust to domain changes. The predicate, the part of speech tags of the objects, and the labels of data source are utilized as features for classification. KFM employs a Maximum Entropy Model as the classifier, which is suitable for multi-class classification.

The data model of a knowledge fusion map is a hierarchical tree whose root is the topic and leaf nodes are RDF data. There exist edges with facet labels between the topic and corresponding RDF data. With this model, knowledge fusion maps can be easily presented by XML or JSON data on the Web.

### 4. DEMONSTRATION

#### 4.1 Visualization

KFM is implemented as a Java Web application. It allows users to search their interested topics via a Web-based interface. To visualize the knowledge fusion map, JavaScript InfoVis Toolkit 6 (an open-source tool for creating interactive data visualizations on the Web) is incorporated into KFM. We also designed a new layout algorithm so that the knowledge fusion map looks more reasonable and intuitive.

A screenshot of the UI is shown in Figure 3, when the search button is clicked, a knowledge fusion map will be shown in the left panel. The topic is represented by the central node. Its sub-nodes indicate the facets of the topic, and are represented as rounded rectangles. Rectangles following the rounded rectangles contain the corresponding knowledge. The right panel contains some related topics of the searching topic.

### 4.2 Preliminary Results

By experimenting in three domains, KFM has constructed 13 categories of facet templates, totally 1,708 topics and their knowledge fusion maps, as shown in Table 1. All RDF data about topics is crawled by LDspider. The knowledge fusion maps are updated dynamically.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Categories of facet templates</th>
<th>#Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data structure</td>
<td>Graph, Tree, Array, List, Computer algorithm</td>
<td>519</td>
</tr>
<tr>
<td>Data mining</td>
<td>Data, Classification, Cluster algorithm, Artificial neural network</td>
<td>625</td>
</tr>
<tr>
<td>Computer network</td>
<td>Routing algorithm, Network architecture, Protocol, Networking standards</td>
<td>564</td>
</tr>
</tbody>
</table>

### 5. CONCLUSION AND FUTURE WORK

To solve the RDF data fragmentation problem, we have developed a novel system KFM, which automatically generates facet templates for domain topics and fuses RDF data by leveraging topological properties of TRG. KFM not only helps users get a thorough understanding about a specific topic, but it also provides an effective way of facet navigation for topics in LOD. In the future, we will improve the search model over knowledge fusion maps so that KFM can support more functions, such as facet searching and question answering.

### 6. ACKNOWLEDGMENTS

The research was supported in part by the National High-Tech R&D Program of China under Grant No. 2012AA011003, the National Science Foundation of China under Grant Nos. 61221063, 61173112, the National Key Technologies R&D Program of China under Grant No. 2013BAK09B01, the Doctoral Fund of Ministry of Education of China under Grant No. 20130201130002, the Ministry of Education Innovation Research Team under Grant No. IRT13035.

### 7. REFERENCES


---