Knowledge-Based Systems 000 (2017) 1–11

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# A hierarchical co-clustering approach for entity exploration over Linked Data

Liang Zheng, Yuzhong Qu*, Xinqi Qian, Gong Cheng

*National Key Laboratory for Novel Software Technology, Nanjing University, China*

ARTICLE INFO

*Article history:*
Received 20 February 2017
Revised 10 November 2017
Accepted 13 November 2017
Available online xxx

*Keywords:*
Linked Data
Entity exploration
Hierarchical co-clustering

ABSTRACT

With the increasing amount of Linked Data on the Web, large numbers of linked entities often make it difficult for users to find the entities of interest quickly for further exploration. Clustering as a fundamental approach, has been adopted to organize entities into meaningful groups. In general, link and entity class are semantically labelled and can be used to group linked entities. However, entities are usually associated with many links and classes. To avoid information overload, we propose a novel hierarchical co-clustering approach to simultaneously group links and entity classes. In our approach, we define a measure of intra-link similarity and intra-class similarity respectively, and then incorporate them into co-clustering. Our proposed approach is implemented in a Linked Data browser called CoClus. We compare it with other three browsers by conducting a task-based user study and the experimental results show that our approach provides useful support for entity exploration. We also compare our algorithm with three baseline co-clustering algorithms and the experimental results indicate that it outperforms baselines in terms of the Clustering Index score.

## 1. Introduction

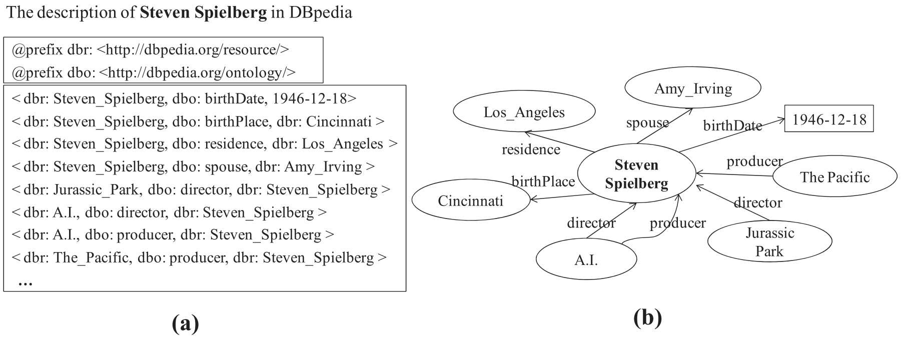
The volume of Linked Data on the Web has increased rapidly [1]. It can be represented as a collection of triples by using the Resource Description Framework (RDF) [2]. In Fig. 1, there is an RDF description of Steven Spielberg in DBpedia [3]. These entity-centric structured data can be reused to facilitate a wide variety of applications such as Web search and business intelligence. However, with the enrichment of available entity-centric Linked Data, large numbers of linked entities often make it difficult for users to find the entities of interest quickly for further exploration. For instance, from Steven Spielberg in DBpedia, it is possible to navigate toward 117 entities. The user would have to browse a large number of entities and choose the desired ones from these entities. A common technique that has been adopted to organize the large amount of data is clustering [4,5]. Clustering refers to the process of grouping data objects into multiple groups based on different similarity measures such as text-based [6], CoCitation [7] and SimRank [8]. In the case of entity exploration over Linked Data, clustering linked entities can provide a good way to help users navigate and seek the needed entities.

Since link and entity class are two key entity facets, they are generally used to group linked entities. In addition, they can provide semantic labels for the generated clusters. By using different types of links, generic Linked Data browsers (e.g., Tabulator [9]) naturally group linked entities and then present users with a list of links. These links reflect the relationships between current entities and their linked entities. Suppose the user is looking for the works produced by Steven Spielberg. The user can select *producer* link to further explore the works he produced. On the other hand, since entity class provides users an intuitive understanding of entity, it can be used to distinguish and group entities. For instance, Aemoo [10] is an exploratory search application over Linked Data, which groups the entities' neighbours (i.e., their linked entities) based on entity classes and then displays the most relevant entity classes. Suppose the user is looking for the films related to Steven Spielberg. The user can find linked films by an entity class *Film*. Moreover, there are several advanced systems based on faceted browsing (e.g., /facet [11]). They provide different kinds of facets to group entities in multiple dimensions. In fact, facets are raw links and classes. Users can explore a collection of entities by selecting links and classes. For instance, the user can find the films produced by Steven Spielberg through the link *producer* and the class *Film*.

As mentioned above, link and entity class offer effective ways to group entities and thus can be used to help users orient and control entity exploration. However, entities are usually associated with many classes and links. Fig. 2 shows the context of browsing

[*] Corresponding author.
*E-mail address:* yzqu@nju.edu.cn (Y. Qu).

https://doi.org/10.1016/j.knosys.2017.11.017
0950-7051/© 2017 Elsevier B.V. All rights reserved.



The description of **Steven Spielberg** in DBpedia

```
@prefix dbr: <http://dbpedia.org/resource/>
@prefix dbo: <http://dbpedia.org/ontology/>
```

```
< dbr: Steven_Spielberg, dbo: birthDate, 1946-12-18>
< dbr: Steven_Spielberg, dbo: birthPlace, dbr: Cincinnati >
< dbr: Steven_Spielberg, dbo: residence, dbr: Los_Angeles >
< dbr: Steven_Spielberg, dbo: spouse, dbr: Amy_Irving >
< dbr: Jurassic_Park, dbo: director, dbr: Steven_Spielberg >
< dbr: A.I., dbo: director, dbr: Steven_Spielberg >
< dbr: A.I., dbo: producer, dbr: Steven_Spielberg >
< dbr: The_Pacific, dbo: producer, dbr: Steven_Spielberg >
...
```

**Fig. 1.** An excerpt of the description of Steven Spielberg in DBpedia ((a): a set of RDF triples; (b): an RDF graph).

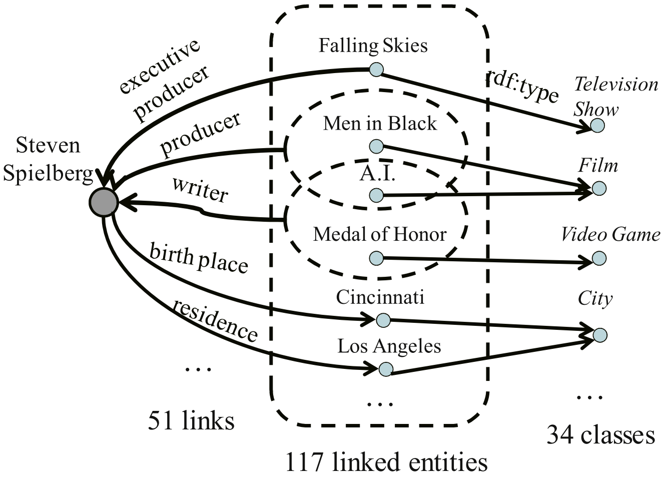


**Fig. 2.** The context of browsing Steven Spielberg.

Steven Spielberg. There are 117 linked entities, which are associated with 51 links and 34 classes. Users' direct interaction with a large number of links and entity classes could cause the problem of cognitive overhead. To avoid this problem, efforts have been made to rank links and entity classes based on certain criteria (e.g., link frequency and entropy [12]), or organize them as a hierarchy based on their structured features (e.g., SynopsViz [13]). Yet, existing studies take into account only the raw knowledge granularity (link and class) but ignore the intra-similarity among links or classes, as well as the potential relationships between links and classes. As shown in Fig. 2, the link *executive producer* and *producer* are similar based on the lexical similarity between their labels. The class *Television Show* and *Film* are similar based on the semantic similarity (they are subclasses of *Work*). Links and entity classes can be further clustered. In addition, there is a connection between the link *producer* and the class *Film* through an entity A.I. It is often desirable to simultaneously cluster both links and entity classes by exploiting their relationships.

To facilitate entity exploration, we propose a novel hierarchical co-clustering approach to simultaneously group links and entity classes. Our approach is implemented in a prototype system called CoClus.[1] We use the following example to illustrate our approach.

Suppose a user is exploring an entity Steven Spielberg and he wants to find some linked entities of interest for further exploration. The user inputs an entity URI[2] in CoClus (A), as shown in Fig. 3. The system enters a cluster interface. The left-hand side of the interface lists 3 link clusters, such as {*is relative of* ,...}, {*is foundedBy of* ,...} and {*is director of* ,...} (B). The right-hand side lists 3 class clusters, such as {*Place*,...}, {*Organisation*,...} and {*Work*,...} (C). The user knows that Steven Spielberg has three major kinds of linked entities based on links and classes, respectively. Each cluster has its subhierarchy. The user can choose a link cluster or class cluster to further observe its sub-clusters at will. Then, the user selects the {*is director of*,...} link cluster to further observe its sub-clusters. The interface refreshes and shows some sub-clusters such as {*is writer of*,...}, {*is director of*,...} and {*is executiveProducer of*,...} (D). Meanwhile, the interface shows some related class clusters such as {*Work*,...}, {*Software*,...} and {*TelevisionShow*,...} (E). The interface also shows the navigation path ("*is director of*") and supports rollback (F). The user selects a class cluster {*Software*,...} and there are 4 linked softwares (G). By iteratively exploring link and class clusters step-by-step, the user understands the overview of information and finally captures an interesting fact that Steven Spielberg is the *writer* of a *Video Game* Medal of Honor.

In our previous work [14], we began to study the co-clustering approach of entity exploration over Linked Data. The objective of this paper is to continue this line of research by studying more effective approaches, performing a more complete evaluation of their performance, and providing an effective approach to explore the Linked Data. In particular, the contributions of this paper are as follows:

- We detail some existing similarity measures and co-clustering algorithms. Moreover, we introduce a hierarchical co-clustering approach, which leverages the idea of hierarchical clustering and tackles every partition of the node as a process of co-clustering. It attempts to overcome the shortcoming of previous work: it could obtain various local minima when starting from an initial random partition, and need to compute the joint distribution over the whole data set during each iteration.
- We perform an empirical evaluation of the performance of our approach. We implement our approach in a Linked Data exploratory system called CoClus, and provide a detailed description of the system. In order to assess its usability, we compare CoClus with its two restricted versions and one conventional Linked Data system by conducting a task-based user study, and test the statistical significance of the results. We also compare our algorithm with three baseline co-clustering algorithms in terms of the Clustering Index score.

The rest of this manuscript is organized as follows. Section 2 reports related work. Section 3 introduces our proposed approach. Section 4 provides our experimentation. Section 5 concludes this paper.

## 2. Related work

Since our work leverages hierarchical co-clustering for entity exploration, in this section we will separately review the previous

[1] http://ws.nju.edu.cn/coclus/.
[2] http://dbpedia.org/resource/Steven_Spielberg.