Data & Knowledge Engineering 112 (2017) 79–93

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

# Constructing faceted taxonomy for heterogeneous entities based on object properties in linked data

Nansu Zong[a], Hong-Gee Kim[b], Sejin Nam[c,*]

[a] Department of Biomedical Informatics, School of Medicine, UC, San Diego, United States of America
[b] Biomedical Knowledge Engineering Laboratory, Seoul National University, Republic of Korea
[c] National Center of Excellence in Software, Chungnam National University, Republic of Korea

ARTICLE INFO

*Keywords:*
Ontology learning
Taxonomy construction
T-Box learning
Faceted taxonomy
Linked data

ABSTRACT

The interlinking of data across the web, a concept known as Linked Data, fosters opportunities in data sharing and reusability. However, it may also pose some challenges, which includes the absence of concept taxonomies by which to organize heterogeneous entities that are from different data sources and diverse domains. Learning T-Box (Terminology Box) from A-Box (Assertion Box) has been studied to provide users with concept taxonomies, and is considered a better solution than mapping Linked Data sets with published ontologies. Yet, the existing process of automatically generated taxonomies that classify entities in a particular manner can be improved. Thus, this study aims to automatically create a faceted taxonomy to organize heterogeneous entities, enabling varying classifications of entities by diverse sub-taxonomies, to support faceted search and navigation for linked data applications. The authors have developed a framework on which each facet represented by an object property is used to extract portions of data in the data space, and an Instance-based Concept Taxonomy generation algorithm is developed to build a sub-taxonomy. Additionally, the strategies for sub-taxonomy refinement are proposed. Two experiments have been conducted to prove the promising performances of the proposed method in terms of efficiency and effectiveness.

## 1. Introduction

Following a standard data interchange model known as Resource Description Framework (RDF), Linked Data (LD) provides a data format method to bridge the local data to the related resources and has become popular among data publishers. Published Linked Data (LD) sets are usually transformed from relational databases or web pages with the aid of transformation programs [1,2]. As a result, these transformed LD are incomplete due to lack of organization (typically without supporting ontologies) [3]. An ontology, with declarations about the general properties of concepts (i.e., classes) in T-Box (Terminology Box) and assertions about individuals (i.e., instances) in A-Box (Assertion Box) [4,5], is generally used to organize LD [6]. With an ontology, instances sharing common properties are grouped and represented by a concept and can further be organized hierarchically (*is_A* relationships). Without the expressive T-Box and A-Box of an ontology to describe the relations between concepts and to organize instances (A.K.A., entities. The terminologies of "entity" and "instance" are used interchangeably in this article), the LD is limited in knowledge acquisition of such options as inferencing, searching, and natural language understanding [3].

In order to gain complete data organized in a way to support different applications, two methods are conventionally used: (1) mapping instances to an existing ontology [7]; or (2) generating an ontology directly from data sources [8]. However, it is not ideal to

* Corresponding author.
*E-mail address:* sjnam@cnu.ac.kr (S. Nam).

http://dx.doi.org/10.1016/j.datak.2017.09.006
Received 26 March 2017; Received in revised form 17 September 2017; Accepted 26 September 2017
Available online 30 September 2017
0169-023X/ 

squeeze every RDF repository under a single ontology, nor enforce unwilling data providers to make their LD adhere to any published ontology. Therefore, learning T-Box from A-Box for LD is considered as a better way of describing the local data set and representing the knowledge induced from instances [3,9]. The learning methods that generate a taxonomy for a LD set reflect a single implicit perspective of viewing or understanding the data (entities). Often, there is difficulty categorizing the compound entities that can be viewed from diverse angles [10]. Multiple perspectives of viewing data are occasionally considered when constructing an ontology. This organization can confuse users in understanding the data. For example, YAGO2 considers both ethic and occupation to classify the concept "*person*", and makes "*bad person*" and "*dancer*" to be siblings and both sub-concepts of "*person*". Therefore, instead of using one single static organization, faceted browsing or navigation that provides a more flexible organization to view LD entities with multiple dimensions (i.e., properties) draws the attention of the Semantic Web community [11–13]. Although faceted navigation has received much attention in research, automatic construction of faceted taxonomy for LD entities still requires further investigation.

To meet different needs arising from various uses of taxonomies, we propose a robust method for generating a faceted taxonomy based on object properties of entities in LD. We have developed a framework to automatically build a faceted taxonomy with potentially multiple sub-taxonomies based on object properties, where each provides a single perspective to view data. Each sub-taxonomy in a facet organizes entities in a single dimension (object property), which is generated with Instance-based Concept Taxonomy generation algorithm called ICT, adapted from an instance-based ontology alignment algorithm [14] as well as the strategies for instantiation and refinement. Our experiments comprise two tasks to evaluate the proposed method: (1) to test the ability of constructing a hierarchy for a sub-taxonomy, and (2) to test the ability of constructing a faceted organization for navigation. For Task 1, single sub-taxonomies built based on "*rdf:type*" in DBpedia [15] and YAGO2 [16], respectively, are evaluated with Taxonomic F-measure score. For Task 2, faceted taxonomies built based on different object properties in Diseasome[1] [17] and DrugBank[2] [18] are evaluated with Inheritance Richness, Maximum Resolution and Class Importance. We have achieved encouraging results in the two tasks. For Task 1, the proposed method takes 49 ms for DBpedia and 11,790 ms for YAGO2 in building the concept taxonomies with 0.917 and 0.780 on Taxonomic F-measure scores (i.e., effectiveness of taxonomy construction) after having loaded them in main memory. For Task 2, the method shows 2032 ms for Diseasome and 2525 ms for DrugBank in building a faceted taxonomy with 1.65 and 1.03 on Maximum Resolution scores with 2 facets (i.e., retrieval effectiveness with a faceted taxonomy).

The contributions of this study are: (1) to generate a faceted taxonomy by defining the notions of a facet and sub-taxonomy in the facet taxonomy in LD; (2) to generate a faceted taxonomy by proposing a framework to dynamically generate a faceted concept taxonomy with object properties; (3) to generate a sub-taxonomy by proposing the ICT with an instantiation and a taxonomy refinement strategy. In the following, Section 2 introduces related works and gives the basic principles of our solution. Section 3 introduces the proposed solution. Section 4 details on the method of faceted taxonomy generation. Sections 5 and 6 demonstrate the results of the experiments. Section 7 discusses limitations of this study and concludes our work.

## 2. Background and related works

### 2.1. Background

Resource Description Framework (RDF) is a metadata data model for conceptual description or information expression, proposed and promoted by the World Wide Web Consortium (W3C) [19]. A resource in RDF denotes an entity that is identified with a de-referenceable URL. A resource can be anything on the Web. For example, disease "*Parkinson's disease*" identified with "http://dbpedia.org/resource/Parkinson's_disease" (Dbpedia:parkinson's_disease) is a resource. An RDF statement is a subject-predicate-object triple. A subject is a resource; an object can be a resource or a literal text; and a predicate (or attribute or property) denotes a relation between the subject and the object. There are two types of property: object type and data type. In a triple, if the object is a resource, the property is of object type, and if a literal text, it is of data type. For example, "parkinson's disease can be possibly treated with dopamine" can be represented with two triples, (1) " < Dbpedia:parkinson's_disease > < *Rdf:label* > *'Parkinson's disease"*, *and (2)* " < Dbpedia:parkinson's_disease > < *Diseasome:possibleDrug* > < *Dbpedia:Dopamine* > ", where the "*Rdf:label*" is a data type property and "*Diseasome:possibleDrug*" is an object type property. Since all resources are described with properties, the vocabularies are defined and can be reused by other RDF documents. For example, "*rdfs:Class*", denoting that a subject is a class, is defined in "http://www.w3.org/2000/01/rdf-schema#". The vocabularies defined by the RDF specification can be found in [19].

Linked Data refers to machine-readable data with meanings explicitly defined. It is published on the Web in such a way that it is linked to other external data sets and it can in turn be linked to from external data sets. Linked Data (LD) uses RDF links to connect a subject with a de-referenceable URL in a local set to an object with a URL reference in an external data set. When an object is de-referenced over the HTTP protocol, a server of this URL will return an RDF document about the object to a client, which helps users to get more related information.

### 2.2. Related works

With the rapid growth of large data sets in commercial, industrial, administrative and other applications, there arises a need to

[1] http://diseasome.eu/
[2] http://www.drugbank.ca/

automatically organize different kinds of data, such as unstructured text data, semi-structured XML data or structured databased data [20]. As a result, the automatic concept hierarchy generation has been studied from 1990s [21,22]. Taxonomies can be generated either with the heuristic rules [23,24] or from external knowledge-bases as references [25]. Nevertheless, the most popular methods are based on probabilistic models. Prior to taxonomy construction, terms are extracted by different Natural Language Processing (NLP) [26–29] or pattern matching methods [23,24] and determined by diverse computation metrics [10,26,27,29] to label the concepts.

The most straightforward way to build a taxonomy is to iteratively group the similar entities together. Therefore, the hierarchical clustering algorithms known as agglomerative UPGMA and bisecting k-means [30] are frequently used, where the latter is reported as a better solution than the former [31]. There are some variances in adapting hierarchical clustering, such as marring UPGMA with heuristic rules for biomedical linked sets [32]. Beside the hierarchical clustering algorithms mentioned above, other clustering methods, such as Self-Organizing Map (SOM), are also used to reduce the dimensions of data (instances) features for clustering data at each level of a taxonomy [33]. These methods have limitations on concepts in the taxonomy being unable to be interpreted with a natural language; thus they are hard to be understood by users.

By considering shared property values, the Formal Concept Analysis (FCA) uses property values as the intensions of a concept and together with extensions (instances) in building a taxonomy [26,29]. Similar to the FCA-based method, our proposed method also utilizes the shared values of the properties in building the hierarchy, while the FCA-based methods focus on organizing instances and disregards concept interpretation or labeling, where the label of a concept is either unreadable or missing. In our experiment with the YAGO2 dataset, the FCA obtains a low precision by generating meaningless none-existent concepts, such as "*YAGO:Wordnet_Abstraction100002137, YAGO:Wordnet_PhysicalEntity100001930, owl#Thing*" that contains 3887 common instances.

Different with the FCA, the Subsumption [27,34] considers property values as concepts and determines a subsumption relation between two concepts with the common instances based on a similarity measure. This method has been considered as one of the most classical methods for concept hierarchy generation and has been adapted with many variances that increases precision based on different probability models, such as the DSP [35] and the DisCover [28]. The EXT [36], with similar definition of concepts, induces a high efficient and effective extensible greedy algorithm that places concepts, ordered with importance of a similarity graph, into a hierarchy based on similarity measures for social tags. Furthermore, the IUT [14] adapts the EXT to align ontologies in LD. The IUT changes the sorting algorithm in the EXT and develops new similarity measures to determine two relations (equivalence and subsumption) of two concepts. Targeting a different problem, the proposed method uses a different sorting strategy based on the number of instances and modifies the similarity measures of the IUT to determine subsumption relations. In contrast to the above methods, the proposed method builds a faceted taxonomy that consists of multiple sub-taxonomies for each facet defined by an object property. To the best of our knowledge, this paper is the first study that puts a way to automatically generate faceted taxonomies for LD.

## 3. Faceted taxonomy for organizing instances in linked data

### 3.1. Method overview

The objective of this study is to automatically construct a concept taxonomy that fully describes its instances. Considering that different instances in the same topic may have same objects for an object property, we use the objects to cluster the instances and formalize a concept hierarchy for the instances.

Given a LD set containing an A-Box $A = \{I, P, O\}$ that consists of a set of instances $I$, a set of object properties $P$ and a set of object $O$, where each instance $i_u \in I$ is described with a set of property $P_u = \{p_1,\ldots,p_k\}$, each of which has a set of object $O_{i_u p_v} = \{o_1,\ldots,o_t\}$ and $t \geq 1$, We propose a solution of building a faceted taxonomy as shown in Fig. 1 based on the object properties, where each property can be considered as a facet. We adapt the concept of facet in [12] and define a facet in this paper as:

**Definition 1:** a facet $f_u$ for a LD set is an object property $p_u$ in the data set.

For example in Fig. 1, there are two facets called "*cause*" and "*symptom*" that are the object properties for the LD set about diseases.

**Definition 2:** a sub-taxonomy $F_u$ in a facet $f_u$ is a concept taxonomy with the triples extracted with the property $p_u$. The sub-taxonomy $F_u(C_u, R_u)$ consists of a set of concepts $C_u = \{c_1, c_2,\ldots c_t\}$ and a set of subsumption relations $R_{S_u} = \{r_{S_{u1}}, r_{S_{u2}},\ldots r_{S_{ux}}\}$. A subsumption relation $r_s(c_i, c_j)$ is a subsumption relation between two concepts $c_i$ and $c_j$, where $c_i$ and $c_j \in C_u$.

For example in the "Facet1: cause" in Fig. 1, there is a sub-taxonomy that contains five subsumption relations, such as the subsumption connected between the concepts "*Obese*" and "*Acquired*".

**Definition 3:** a faceted taxonomy $F$ includes a set of sub-taxonomies $\{F_1, F_2,\ldots F_k\}$, where each sub-taxonomy $F_u$ organizes the concepts in a facet $f_u$.

For example in Fig. 1, a faceted taxonomy has two sub-taxonomies about the facets "*cause*" and "*symptom*", and each sub-taxonomy in a facet uses different concepts that are organized with a different hierarchy.

**Definition 4:** a materialized faceted taxonomy $\mathcal{F}(F, R)$ includes a set of sub-taxonomies $F = \{F_1, F_2,\ldots F_k\}$ and a set of "instance of" relations $R_I = \{r_{I_1}, r_{I_2},\ldots r_{I_e}\}$. An "instance of" relation $r_I(i_u, c_v)$ is a classification of instance $i_u$ to a concept $c_v$, where $i_u \in I$ and $c_v \in C_u$.

For example in Fig. 1, the materialized faceted taxonomy has three instances to instantiate the concepts in two sub-taxonomies, such as "*Translocation Down syndrome*" is an instance of "*Hereditary*" and "*Speech disturbance*".

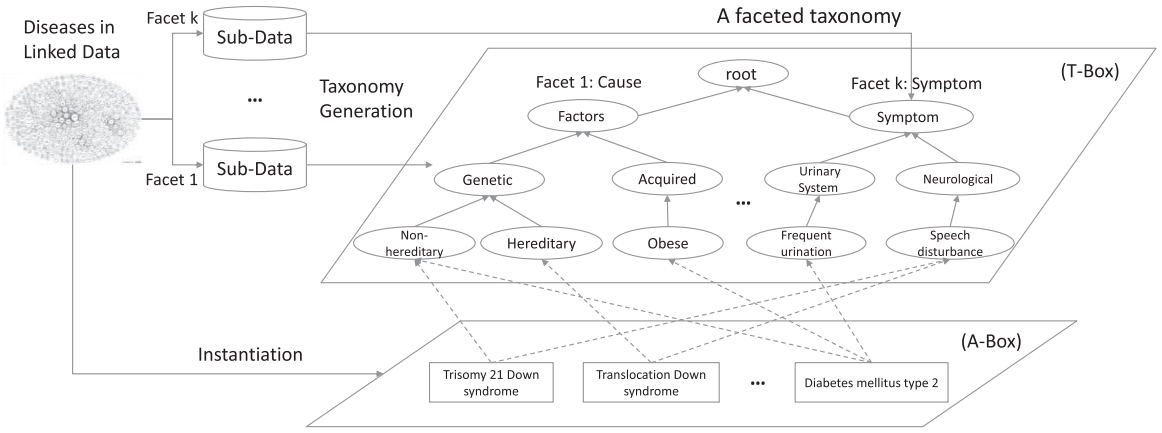



**Fig. 1.** A faceted taxonomy for a sample of LD.

### 3.2. Concept generation and naming

Classes (concepts) provide an abstraction mechanism to generalize the characteristics of a group of similar instances in a taxonomy. A concept can be interpreted with its extensions and intensions [37]. Therefore, we use the instances as the extensions to define the concept [38] in a sub-taxonomy for a facet as follows:

**Definition 5:** A concept that contains the extensions, a set of instances, is a binary vector $\vec{c} = [i_1, i_2,\ldots,i_m]$, where $i_m = 1$ when the concept contains $i_m$.

With the class axiom on instances [38], a concept can be determined if its extensions contain the extensions of its sub-concepts. Therefore, we can generate a concept with the extensions of its sub-concepts as:

$$\vec{c} = \vec{c}_1 OR \vec{c}_2 OR \ldots OR \vec{c}_i \tag{1}$$

where $\vec{c}_i$ is a sub-concept of $\vec{c}$.

The intensions of a concept are the features, which follow inheritance axiom, a sub-concept inherits all the intensions from its super concept [39]. In LD, the objects contained in an instance are overlapped with the objects of other instances. Therefore, in a facet defined by an object property, the objects are considered as the intensions as follows:

**Definition 6:** Given an object set for an object property, the intentions $c$ of a concept $\vec{c}$ are the object set $\{o_1, o_2,\ldots,o_k\}$ contained by the concept.

For any concept $\vec{c}$ in a taxonomy, we can obtain the intensions $c = \{o_1, o_2,\ldots,o_m\}$ of this concept based on the intersection of the intensions $\{c_1, c_2,\ldots,c_i\}$ for all the sub-concepts:

$$c = \bigcap c_i \tag{2}$$

where $c_i$ is the intensions of a sub-concept $\vec{c}_i$. For example in Fig. 1, the objects "*Genetic*", "*Non-hereditary*", and "*Hereditary*" of the property "*Cause*" for two instances "*Trisomy 21 Down syndrome*" and "*Translocation Down syndrome*" can generate a concept with the intention {"*Genetic*"} and two sub-concepts with the intentions {"*Genetic*", "*Non-hereditary*"} and {"*Genetic*", "*Hereditary*"}. According to the concept axiom in OWL 2 [40], a concept can be considered as its own instances. Therefore, with the instances, a super concept can be formed from the concepts in the bottom.

We name a concept with the reduced labeling strategy [29] based on the intensions. The name of a concept is the intensions that exclude its intention from the intensions of its super concepts:

$$name(c) = (c \not\supset \bigcup c_i) \tag{3}$$

where $c_i$ is the intensions of a super concept $\vec{c}_i$. In Fig. 1, the sub-concept with the intentions {"*Genetic*", "*Non-hereditary*"} has only one super-concept with the intention {"*Genetic*"}; therefore, its name is generated by excluding "*Genetic*" from "*Genetic*" and "*Non-hereditary*". A concept is considered valid when its name contains only one object.

**Definition 7:** A concept $\vec{c}$ is valid only if $|name(c)| = 1$.

With Definition 7, a concept $\vec{c}$ will be named after the only factor left in $name(c)$, and cannot be named if $|name(c)|>1$. Those concepts that cannot be named with the definition will simply be removed. This strategy is proved to be more efficient and more effective than other methods, such as the FCA-based method described in Results.

### 3.3. Framework

Based on the definitions of a concept in the last subsection, we have designed a framework to automatically generate a faceted taxonomy in two stages as shown in Fig. 2: Pre-processing and Taxonomical relationship generation.

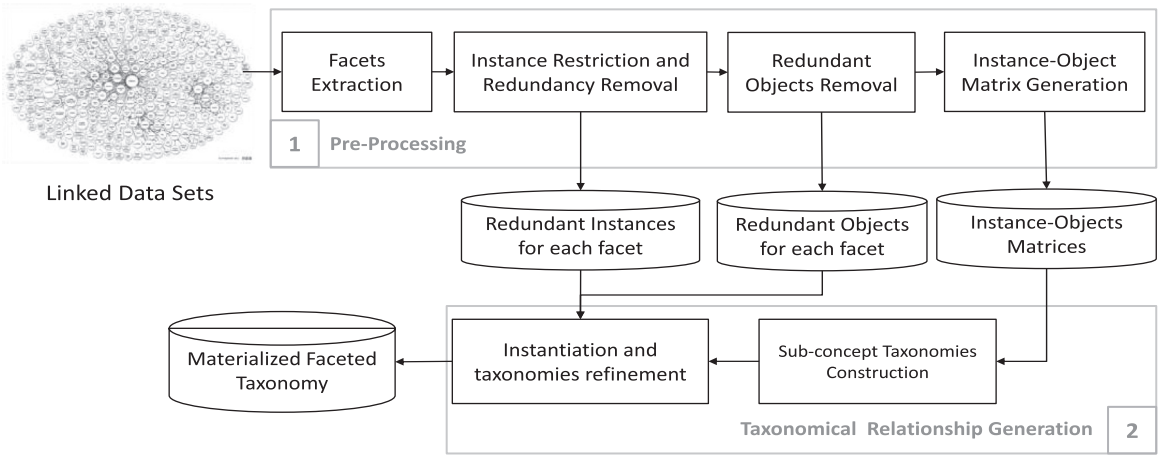
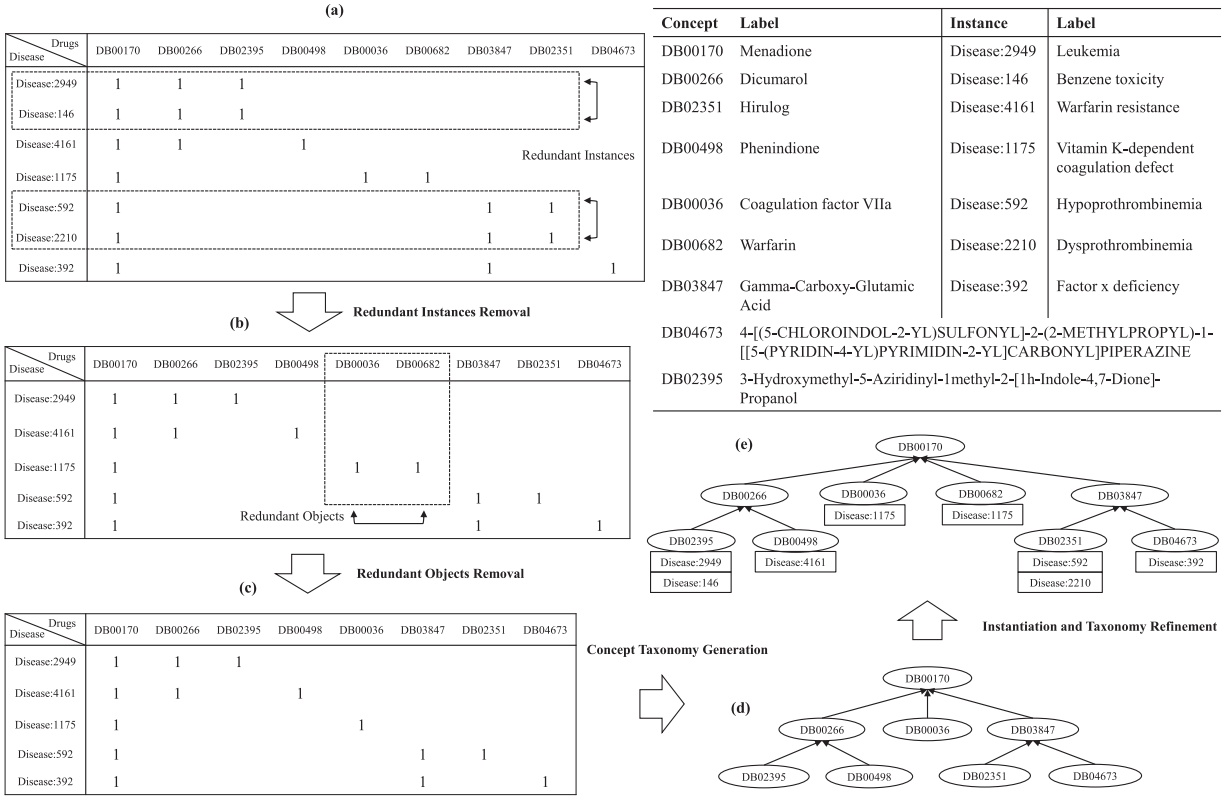



**Fig. 2.** The framework of faceted taxonomy construction.

The pre-processing is to generate a set of instance-object matrices, each of which represents the relations between instances in one facet (object property). Four steps, (a) facets extraction, (b) instance restriction and filtering, (c) object filtering, and (d) instance-object matrix generation, are used at this stage in order to filter redundant instances and objects for reducing the computations of sub-taxonomy generation.

The taxonomical relationship generation is to construct sub-taxonomies based on instance-object matrices generated from multiple facets. For each matrix, we propose an algorithm to build a sub-taxonomy. An instantiation and concept taxonomy refinement strategies are also proposed to get a materialized faced taxonomy in the following section. In order to help users better understand the proposed method, we will use a sample data from Diseasome to explain the procedure of generating a sub-taxonomy in one facet. The input dataset in Fig. 3 shows a partial view of the disease instances extracted from Diseasome with the object property "*Diseasome:possibleDrug*". For each disease instance, there are multiple drug entities as the values of its property. For example, the disease "*Disease:2949*" has three drug entities as the values: "*DB00170*", "*DB00266*", and "*DB02395*". Disease names are labeled by "*rdf:label*", for example, "*Leukemia*" for "*Disease:2949*".




| Concept | Label | Instance | Label |
|---|---|---|---|
| DB00170 | Menadione | Disease:2949 | Leukemia |
| DB00266 | Dicumarol | Disease:146 | Benzene toxicity |
| DB02351 | Hirulog | Disease:4161 | Warfarin resistance |
| DB00498 | Phenindione | Disease:1175 | Vitamin K-dependent coagulation defect |
| DB00036 | Coagulation factor VIIa | Disease:592 | Hypoprothrombinemia |
| DB00682 | Warfarin | Disease:2210 | Dysprothrombinemia |
| DB03847 | Gamma-Carboxy-Glutamic Acid | Disease:392 | Factor x deficiency |
| DB04673 | 4-[(5-CHLOROINDOL-2-YL)SULFONYL]-2-(2-METHYLPROPYL)-1-[[5-(PYRIDIN-4-YL)PYRIMIDIN-2-YL]CARBONYL]PIPERAZINE | | |
| DB02395 | 3-Hydroxymethyl-5-Aziridinyl-1methyl-2-[1h-Indole-4,7-Dione]-Propanol | | |

**Fig. 3.** An ongoing example of building a sub-taxonomy with an object property "*Diseasome:possibleDrug*".

(a) Facets extraction

In our definition, each object property is considered as a facet, and object properties $P$ are identified from all the properties. Any triple that contains an object property is extracted, and the entire instances (subjects) of the extracted triples are used to build a $|P|$ faceted taxonomy. In Fig. 3, "*Diseasome:possibleDrug*" is considered as a facet.

(b) Instance restriction and filtering

First, for each facet, instances are restricted into the domain that contains an object property. The instances without a value for the property were excluded. For example, in Fig. 3, since there is only one facet in the given data set, none of the instances were excluded, and we obtained 21 triples containing seven disease instances and nine drug entities as input data as shown in Fig. 3 (1). Second, the instances that have the same objects are filtered and only one instance is kept as a representative instance for those filtered ones. Therefore, after the first step, the unique instances that have different object sets are extracted for the next step in each facet. As Fig. 3 (1) shows, the instances "*Disease:2949*" and "*Disease:146*" have the same objects "*DB00170*", "*DB00266*", and "*DB02395*". Therefore, "*Disease:146*" is filtered and only "*Disease:2949*" is kept as a representative for the two instances. In Fig. 3 (1), two instances "*Disease:146*" and "*Disease:2210*" are filtered during this step.

(c) Object filtering

After filtering the redundant instances that have same property values (objects) in each facet, we filter the redundant objects that are contained by same instances and keep only one arbitrary object as a representative object for those filtered ones. In another word, only the objects that are contained by unique set of instances are kept for generating an instance-object matrix in a facet. For example in Fig. 3 (2), the objects "*DB00036*" and "*DB00682*" are contained by same instance "*Disease:1175*". Therefore, "*DB00682*" is filtered and only "*DB00036*" is kept as a representative for the two objects in Fig. 3 (2).

(d) Instance-object matrix generation

Based on above three steps, the instances with objects in a facet will form a binary matrix $A_{m\times n}$ with each instance saved as row and each object as a column, and the matrix will be used to generate a sub-taxonomy for this facet. For each entry of the matrix, $a_{uv} = 1$ if the instance $u$ contains the object $v$. There are $|P|$ matrices for all object properties $P$, and each matrix has different number of instances and objects. In the example of Fig. 3 (3), for the facet "*Diseasome:possibleDrug*", the instance-object matrix that has a five (instances) by eight (objects) matrix is generated.

## 4. Generating faceted taxonomy

### 4.1. Instance-based taxonomy generation algorithm

Our objective is to obtain a faceted taxonomy that contains sub-taxonomies generated with instance-object matrices. With the concept definition and the naming strategy, we adapt the IUT [14] to generate a taxonomy based on instance-concept matrix and call our variation ICT (Instance-based Concept Taxonomy generation). There are two steps to generate the concept taxonomy: first, the objects in the matrix $A_{m\times n}$ are sorted in descending order by the number of instances contained by the object and are put into a queue $Q$ (line 1 in Algorithm 1); second, in each iteration, a concept is de-queued and put onto the right position in a tree by computing the subsumption relation with existing concepts (lines 3–9 in Algorithm 1). We adapted the equation in [34] to determine the subsumption relation between two concepts $\vec{c}_u$ and $\vec{c}_v$ ($\vec{c}_u$ is a sub-concept of $\vec{c}_v$) as follows:

$$sub(\vec{c}_u, \vec{c}_v) = \frac{|\vec{c}_u \cap \vec{c}_v|}{|\vec{c}_u|} \geq \varphi,\ sub(\vec{c}_v, \vec{c}_u) = \frac{|\vec{c}_v \cap \vec{c}_u|}{|\vec{c}_v|} < 1 \tag{4}$$

where $\varphi$ is a threshold to control the ratio of the common instances to determine a subsumption relation of the two concepts.

If a concept $\vec{c}_u$ has two super concepts that have a subsumption relation between them, the concept will be assigned to the leaf concept. For example, if "DB02395" is found to have two super concepts "DB00266" and "DB00170", where "DB00266" is a sub-concept of "DB00170", "DB02395" is going to be assigned to the sub-concept "DB00266". The details of the process of the ICT are shown in Algorithm 1.

**Algorithm 1.** Instance-based Concept Taxonomy generation algorithm (ICT).

Input: an instance-object matrix $A_{m\times n}$, $\varphi$Output: A concept taxonomy $T(C,R_c)$
1. Queue $Q$ = all the objects by descending order of the number of instances contained
2. Initiate a tree $H$ with a root concept $r$
3. **While** $size(Q)>0$ **do**
4. $c_i := dequeue(Q)$
5. Initiate an empty super concepts set $\sup(c_i)$
6. **For** $c_j$ in $H$ **do**
7. **If** $(sub(c_i, c_j) \geq \varphi \,\&\, sub(c_j, c_i) < 1)$ $\sup(c_i) \leftarrow c_j$
8. **If** $\sup(c_i) \neq \emptyset$, put $c_i$ onto the sub-concept of the leaf concepts of $\sup(c_i)$
9. **Else** put $c_i$ onto the sub-concept of $r$
10. Return $H$

### 4.2. Instantiation and taxonomy refinement

In this step, we need to materialize the faceted taxonomy based on multiple sub-taxonomies generated. First, we instantiate the concepts in each sub-taxonomy. We define a following instantiation rule based on an instance-object matrix.

#### 4.2.1. Instantiation rule:

If an instance belongs to two concepts $\vec{c}_u$ and $\vec{c}_v$, where $\vec{c}_u$ is the super concept of $\vec{c}_v$, the instance will be used to populate $\vec{c}_v$, else the instance will be used to populate all the concepts.

Instantiation Rule makes an instance populate the leaf concepts in a sub-taxonomy and also supports multiple instantiation.

Second, the filtered redundant instances and objects in the pre-processing stage are placed where the representative instances and objects are located. For example, "*Disease:146*" is assigned to the concept "*DB02395*" as well since "*DB02395*" is instantiated with "*Disease:2949*".

Third, assemble all sub-taxonomies with renamed concepts. Each sub-taxonomy should be independent and contains different concepts. However, object properties can have the same objects as their values. For example, the objects typed "*DrugBank:references*" are used in both object properties "*DrugBank:drugReference*" and "*DrugBank:generalReference*" for the instances typed "*DrugBank:targets*". Therefore, in order to disjoint all the facet concepts, we prefix each concept name with the name of the property in a facet [12]. In order to browse each sub-taxonomy from an anchor node, all concepts will be put under a "Root" concept that contains all the instances where a concept to be considered as the "Root" concept does not exist (e.g., DB00170 in Fig. 3(a)). Finally, we put all the sub-taxonomies under the concept "*Owl:Thing*" to get the faceted taxonomy.

## 5. Experiments

We have implemented the proposed method based on Oracle JDK 1.6 using an Intel® Xeon CPU E5-2630 with 130 GB RAM on Windows 7 64 bit version. Since our aim is to construct faceted taxonomies for LD, our tests are separated into two tasks: (1) generating a sub-taxonomy for one facet, and (2) generating multiple faceted taxonomies with different object properties. To evaluate the efficiency, running times were calculated for the tasks and to evaluate the effectiveness, the quality of a taxonomy, Taxonomic F-measure was calculated for the first task and Maximum Resolution for the second.

### 5.1. Task 1-Construction of taxonomy with "rdf:type"

#### 5.1.1. Data sets and experiment design

In LD, some data sets that contain ontologies with concept taxonomies publish the classification of instances with "*rdf:type*". Generating taxonomies with "*rdf:type*" can be viewed as the reverse engineering of RDF publishing. Therefore, in order to evaluate the performance of the proposed method to build a sub-taxonomy with one object property, we used the values of "*rdf:type*" in the RDF dumping file of a gold standard ontology to construct a taxonomy. The taxonomy will be evaluated by comparing with the taxonomy of the gold standard ontology. We chose two most well-known sets in LOD, DBpedia [15] and YAGO2 [16] that provide the mature concept taxonomies reflecting on the values of "rdf:type". The dumping files, which contain "*rdf:type*" for all the instances from DBpedia and YAGO2, are used as the input data sets. Two concept taxonomies, extracted from DBpedia and YAGO2 ontology respectively, are used as the gold standard, which includes DBpedia ontology concepts for one, and WordNet synsets (i.e., concepts) based on subsumption relations for the other. We gained our study population, with 2,885,951 instance and 8674 concepts from YAGO2 2.5.3, and 3,243,477 instances and 389 concepts from DBpedia 3.9 shown in Table 1. We filtered the redundant instances and objects, and only kept the unique instances and objects to construct the instance-object matrix during the pre-processing stage. As Table 1 shows, the instances are reduced to 155,602 in YAGO2 and 348 in DBpedia, and the objects are reduced to 7327 in YAGO2 and 375 in DBpedia. The size of the representative instances in DBpedia is smaller than those in YAGO2, since the number of concepts is smaller in DBpedia. Large size of the concepts indicates a complex taxonomy that potentially contains large size of the representative instances having unique object sets. For example, 9320 instances in DBpedia have a same unique object set compared with only 19 instances in YAGO2 in average.

#### 5.1.2. Algorithms in comparison

In task 1, we compared the ICT with two classic concept taxonomy construction algorithms, Subsumption-based (named as Subsumption) [34] and FCA-based (name as FCA) [29]. The two methods generate a concept hierarchy based on the co-occurrence

**Table 1**
Statistic of the data sets originally and after pre-processing.

| | **Original data** | | **After pre-processing** | |
|---|---|---|---|---|
| | # instances | # objects (concepts) | # instances | # objects(concepts) |
| YAGO2 | 2,885,951 | 8674 | 155,602 | 7327 |
| DBpedia | 3,243,477 | 389 | 348 | 375 |

of the terms for the documents. Similar to the proposed method, a document-term matrix is extracted from the raw documents data and used as the input.

For the Subsumption, we iterated all the concept pairs and established a subsumption relation of a pair of two concepts $c_1$ and $c_2$ if the two concepts satisfy the condition $P(c_1c_2) = 1$, $P(c_2c_1)<1$. For the FCA, we used Colibri,[3] an open source project to compute the formal concept lattice. A concept in the lattice was used to build a taxonomy if the concept contained at least one instance, and named by the reduced labeling with the extensional interpretation of the concept [29]. The data pre-processing did not affect the effectiveness of the two methods, therefore, the two methods used the same pre-processed data as the ICT used.

#### 5.1.3. Evaluation criteria

The objective of Task 1 is to test how fast the proposed method can generate a concept taxonomy with a good quality; therefore, we tested all the algorithms with two criteria: efficiency and effectiveness. For efficiency measurement, the running time of a method in the same environment to generate a taxonomy is calculated. Similar to the evaluation of Information Retrieval, Precision and Recall are frequently used to evaluate the effectiveness in Ontology Learning [41], and the two variations, Lexical Precision (LP) & Lexical Recall (LR) and Taxonomic Precision (TP) & Taxonomic Recall (TR) are considered as the one of the most popular criteria in many studies [27,42,43]. Since the object values of "*rdf:type*" are used in Task 1, the LP and LR that favor the methods defining the concepts based on objects are unsuitable as the criteria. In contrast, the TP and TR that use ancestor and descendant relations to calculate the similarity of two concepts based on the Semantic Cotopy (SC) [41] can provide unbiased evaluation results. The Semantic Cotopy of a concept $c$ in an ontology $O$ is defined as:

$$SC(c, O) = \{c_i \mid c_i \in C \wedge (c_i \leq c \bigvee c \geq c_i)\} \tag{5}$$

where $C$ is the concept set of $O$, and $c_i \leq c \bigvee c \geq c_i$ is a ancestor and descendant of $c$. Therefore, the semantic cotopy of two concepts can be used to compute the local taxonomic precision of the two concepts as follows:

$$tp_{sc}(c_1, c_2, O_1, O_2) = \frac{|\, sc(c_1, O_1) \bigcap sc(c_2, O_2)|}{|\, sc(c_1, O_1)|} \tag{6}$$

The $TP_{SC}$ and $TR_{SC}$ are computed based on the local taxonomic precisions and recalls as follows:

$$TP_{SC}(O_1, O_2) = \frac{1}{|C_{O_1}|} \sum_{c_i \in C_{O_1}} \begin{cases} tp_{sc}(c_i, c_i, O_1, O_2) & if \quad c_i \in C_{O_2} \\ 0 & if \quad c_i \notin C_{O_2} \end{cases} \tag{7}$$

where $TR_{SC}(O_1, O_2) = TP_{SC}(O_2, O_1)$.

The Taxonomic F-measure (TF) calculates the harmonic mean of $TP_{SC}$ and $TR_{SC}$ as:

$$TF(O_1, O_2) = \frac{2 \times TR_{SC}(O_1, O_2) \times TP_{SC}(O_1, O_2)}{TR_{SC}(O_1, O_2) + TP_{SC}(O_1, O_2)} \tag{8}$$

We use Semantic Cotopy instead of using Common Semantic Cotopy (CSC) because some approaches, such as FCA, will generate new concepts rather than the existing concepts (value of object property "*rdf:type*") provided in the data. The usage of the Common Semantic Cotopy ignores the new generated concepts of these approaches and over-measures the precision.

### 5.2. Task 2-Construction of multiple faceted taxonomies

#### 5.2.1. Data sets and experiment design

We tested multiple faceted taxonomies with different facets (object properties) in two biomedical LD sets, DrugBak and Diseasome , which do not have ontologies to organize instances. The DrugBank contains 4772 drug instances, and the Diseasome contains 4213 disease instances. We used 5 and 16 object properties from Diseasome and DrugBank to generate faceted taxonomies for the disease and drug instances. Please note that not all the instances have a specific object property. For example, there are 4213 disease instances in Diseasome, and only 1456 of them have values of the object property "*Diseasome:possibleDrug*". We list the statistic information of the two data sets we used in Table 2.

#### 5.2.2. Evaluation criteria

Similar with Task 1, efficiency is measured with running time in the task. However, in contrast to evaluating the quality of a mono-dimensional sub-taxonomy in Task 1, we focus more on how the faceted taxonomy with multiple mono-dimensional sub-taxonomies performs on faceted navigation. Therefore, rather than comparing the hierarchical structure with a gold standard ontology, we tested information distribution over the entire faceted taxonomy and the navigation performance with such distribution. We have adapted three evaluation criteria for this purpose, (1) Inheritance Richness (IR) (OntoQA: Metric-Based Ontology Quality Analysis)[44], (2) Maximum Resolution (MR) [12], and (3) Class Importance (CI) [44].

*5.2.2.1. Inheritance Richness (IR).* The Inheritance Richness describes the distribution of concepts that are across different levels of

[3] http://code.google.com/p/colibri-java/

**Table 2**
Statistic of the two data sets.

| | Object properties | # Instance | # Object |
|---|---|---|---|
| Diseasome | (P1) Diseasome:omim | 2929 | 1778 |
| | (P2) Diseasome:associatedGene | 4213 | 3919 |
| | (P3) Diseasome:chromosomalLocation | 2929 | 915 |
| | (P4) Diseasome:possibleDrug | 1456 | 2235 |
| | (P5) Diseasome:class | 4213 | 24 |
| | (P6) Diseasome:diseaseSubtypeOf | 2929 | 1284 |
| DrugBank | (P1) Drugbank:keggCompoundId | 1331 | 1316 |
| | (P2) Drugbank:pdrhealthLink | 280 | 273 |
| | (P3) Drugbank:brandedDrug | 524 | 1593 |
| | (P4) Drugbank:drugCategory | 1879 | 584 |
| | (P5) Drugbank:chebiId | 736 | 721 |
| | (P6) Drugbank:contraindicationInsert | 1112 | 1112 |
| | (P7) Drugbank:target | 4408 | 4553 |
| | (P8) Drugbank:keggDrugId | 913 | 910 |
| | (P9) Drugbank:interactionInsert | 1036 | 1036 |
| | (P10) Drugbank:rxlistLink | 998 | 994 |
| | (P11) Drugbank:dosageForm | 1209 | 215 |
| | (P12) Drugbank:swissprotPage | 74 | 48 |
| | (P13) Drugbank:drugType | 4,772 | 8 |
| | (P14) Drugbank:patientInformationInsert | 762 | 762 |
| | (P15) Drugbank:possibleDiseaseTarget | 1362 | 1456 |
| | (P16) Drugbank:casRegistryNumber | 2240 | 2218 |

a taxonomy. The Inheritance Richness can be used to detect the shape of the concept taxonomy. A low value of Inheritance Richness indicates a horizontal hierarchy (flat structure) that has a low degree of inheritance level where each concept has a large number of sub-concepts. A high value of Inheritance Richness indicates a vertical hierarchy that has a high degree of inheritance level where each concept has a small number of sub-concepts.

The Inheritance Richness is computed as follows:

$$\mathrm{IR} = \frac{\sum_{c_i \in C} |\, descend(c_i)|}{|C|} \tag{9}$$

where $C$ is the concept set of a taxonomy and $|C|$ is the cardinality of the set. $|descend(c_i)|$ is the cardinality of the set of the descendants of a concept $c_i$. For the taxonomy of the facet 1 in Fig. 1, $|descend(C=_{\text{„}Factors\text{„}})| = 5$ and the taxonomy $IR = \frac{11+5+4+2+1+1+1}{12} = \frac{25}{12}$.

*5.2.2.2. Maximum resolution (MR).* To evaluate the quality of a generated faceted taxonomy, Maximum Resolution is used to measure the retrieval effectiveness with the taxonomy. A Maximum Resolution measures the average minimum number of instances to be manually inspected after a refinement through operations on the faceted taxonomy. A small value of Maximum Resolution illustrates a good classification of a concept taxonomy for reducing the search space. The Maximum Resolution is computed as the average number of instances of $k$-intersection concepts in a $k$ facets taxonomy:

$$MR = \frac{\sum |\, I(\bigcap_{i=1}^{k} c_i)|}{|\{\bigcap_{i=1}^{k} c_i\}\,|} \tag{10}$$

where $\bigcap_{i=1}^{k} c_i$ is a $k$-intersection leaf concepts from $k$ sub-taxonomies that share at least one common instance and $|\{\bigcap_{i=1}^{k} c_i\}\,|$ is the cardinality of such intersection. $\sum |I(\bigcap_{i=1}^{k} c_i)|$ is the total number of instances shared in all the $k$-intersection concepts. For example in Fig. 1, there are four 2-interction concepts (“Non-hereditary and Frequent urination”, “Non-hereditary and Speech disturbance”, “Hereditary and Speech disturbance”, “Obese” and “Frequent urination”), therefore, MR for the taxonomy is $MR = \frac{1+1+1+1}{4} = 1$. Please note that, for $k = 1$, MR is equal with the average number of instances belonging to the leaf concepts.

*5.2.2.3. Class importance (CI).* In order to obtain the important classes, we adapt Class Importance to show the focused concepts depending on the instance distribution, and help users to identify where to get data if the intentions of users’ are to get consistent coverages of all concepts.

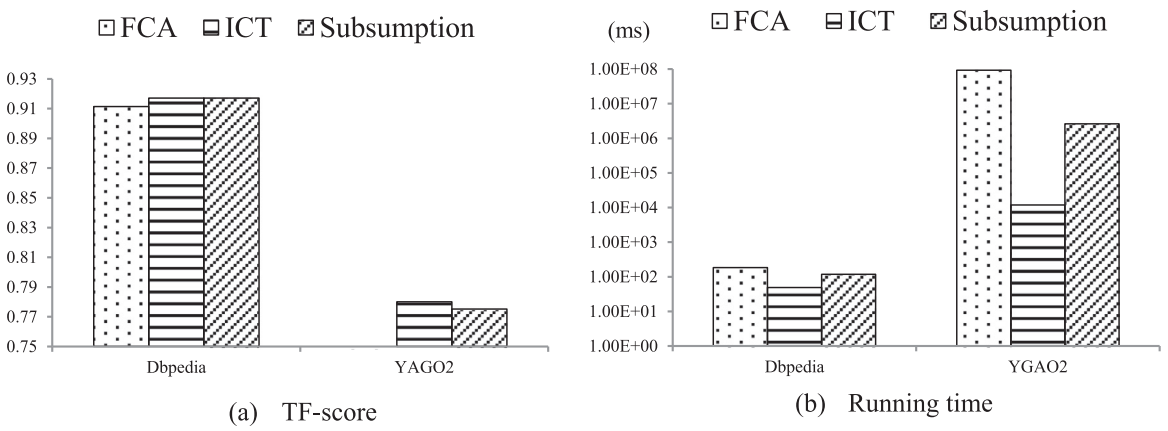



**Fig. 4.** TF-score and running time of the methods. The ICT uses the parameter setting $\varphi = 1.0$.

The importance of a concept $c_i$ is computed as follows:

$$\mathrm{CI}(c_i) = \frac{|I_{c_i}|}{|I|} \tag{11}$$

where $I_{c_i}$ is the instance set of $c_i$. Please note that the instances belonging to $c_i$ contain all the instances belonging to each sub-concept of $c_i$. For example, CI for the concept "*factors*" is computed as $\mathrm{CI}(c_{=\text{"factors"}}) = \frac{3}{3} = 1$.

## 6. Results

### 6.1. Results of task 1

We have run three methods over each of the two data sets, YAGO2 and DBpedia. For each method, one of the data sets is loaded into the main memory in its entirety for each task and running times of the methods are calculated. As Fig. 4(b) shows, the ICT is the fastest method: 49 ms for DBpedia (58% and 73% less compared to Subsumption and FCA, respectively), and 11,790ms for YAGO2 (99.5% and 99.99% less compared to Subsumption and FCA, respectively). The ICT and Subsumption reduce the search space into the concepts already existing in the concept taxonomy with Definition 7, which excludes concepts containing more than one object in the names. However, instead of calculating all the concept pairs [14], the ICT improves the efficiency by restricting the calculation of Eq. 4 with the concept in the tree (118 ms for DBpedia and 2,597,424 ms for YAGO2). The FCA (184 ms for DBpedia and 92,656,444 ms for YAGO2) is the slowest method since the size of the lattice can get exponential for large data size [29] . In our two tests, over than 55% of the running time is spent on the extraction of the subsumption relations from all the discovered relations (103/184 for DBpedia and 90,942,222/92,656,444 for YAGO2).

To evaluate the effectiveness of each method, the concept taxonomies generated by the method are compared with the gold standards of the two data sets (DBpedia ontology and YAGO-WordNet), to obtain the Taxonomic F-measure scores. The taxonomies are stored in the memory as graph structures that keep the subsumptions and instantiations of each concept. As Fig. 4(a) shows, the ICT obtains the best f-scores for two data sets (0.917 for DBpedia and 0.780 for YAGO2). Eq. 4 can successfully establish a subsumption relation between two concepts. For example, "*DBpedia:FloweringPlant*" contains two instances "*Dinka_(grape)*" and "*Miconia_laxa*", and "*DBpedia:Grape*" contains one instance "*Dinka_(grape)*". Therefore, the subsumption relation can be easily built with this equation.

There are two kinds of failures to affect the precision and recall known as false negative and false positive:

(1) Two concepts A and B, having a subsumption relation but containing a same instance set, can cause a false negative. For example, "*YAGO:Wordnet_art_school_102746978*" and "*YAGO:Wordnet_school_104146050*" have the same instance set "*St._Martin's_Lane_Academy*", "*Cranbrook_Educational_Community*", "*Faculty_of_Theatre_(Prague)*". This problem is recognized as insufficient taxonomic description on the instance level [14], which is caused by single child node in taxonomy or none instantiation of the sibling nodes [5,45]. In YAGO2 and DBpedia, there are 1634 and 16 pairs of concepts that cause false negatives.

(2) Two concepts A and B, not having a subsumption relation but containing two instance sets that one subsumes the another, can cause a false positive. For example, the concept "*YAGO:Wordnet_Television106277280*" has 68 instances that includes the only instance "*Plats_bruts*" contained by concept "*YAGO:Wordnet_TeachingAid104397261*", where there does not exists a subsumption relation between the two concepts. This problem is recognized as multi-instantiation whereby one instance can be used to populate multiple concepts [14]. In DBpedia, there exists none false positives, but in YAGO2, there are 1769 pairs of concepts cause false positives due to multiple instantiation [46]. The ICT performs same with the Subsumption on DBpedia (0.917) but better on YAGO2 (0.775). The $\varphi$ controls the level of tolerance for detecting a subsumption relation. The precision

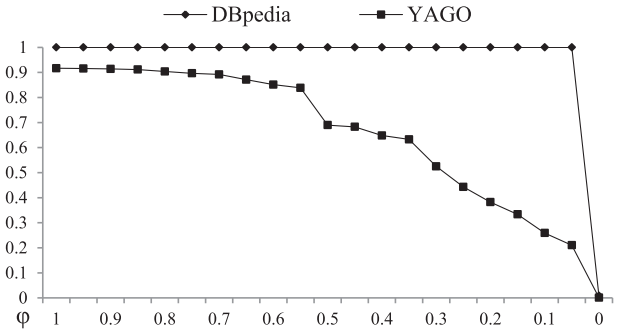



**Fig. 5.** TP-scores of ICT for DBpedia and YAGO2 with different $\varphi$.

decreases along with the decrease of $\varphi$ if there is multi-instantiation in the data set, as the YAGO2 shown in Fig. 5. In Fig. 5, DBpedia does not have multi-instantiation, so the precision is not affected by $\varphi$. The parameter "$\varphi$" was selected empirically. Our experiments showed that the threshold value of 1 achieved the best results with the given data; hence, 1 is set as the default value for "$\varphi$".

The FCA achieves favorable results on DBpedia (0.911) but fails on YAGO2 (0.00058). In YAGO2, the FCA exploits every possible concept (1,397,220) containing common instances, comparing the concepts (8,674) in the gold standard ontology, which causes high recall (0.679) but extremely low taxonomic precision (0.00029). The ICT solves this issue with Definition 7, which ignores the concepts that cannot obtain meaningful names after reducing labels.

### 6.2. Results of task 2

We measured the running time for each sub-taxonomy with a single property in Diseasome and DrugBank. Similar with Task 1, the data sets are loaded into the main memory. As Fig. 6 shows, "Diseasome:associatedGene" (1202 ms) and "*Drugbank:target*" (1453 ms) spend the longest time on constructing a sub-taxonomy in one facet for the two data sets. We learned that the time spent is related to the number of objects contained in an object property, and the more objects contained the longer it takes for a property [14]. For example, "*Diseasome:class*" has 24 objects and spends 16 ms on creating a sub-taxonomy comparing with "*Diseasome:possibleDrug*" that spends 206 ms on creating a sub-taxonomy with 2235 objects.

Similar with Task 1, the same graph structures are used to store the generated taxonomies in memory. The subsumption relations and instantiations are easily extracted to perform three evaluation metrics, Inheritance Richness, Maximum Resolution, and Class Importance, for evaluating the effectiveness when there is no a gold standard for comparison.

We used Inheritance Richness (IR) to pry into the structure of each sub-taxonomy. As Table 3 shows, the "*Diseasome:possibleDrug*" and "*Drugbank:target*" get the highest IR scores on Diseasome (15.22) and DrugBank (17.72). The concept taxonomies generated with these two properties have 5 levels and 7 levels of inheritance. Therefore, we can obtain a vertical shaped concept taxonomies with "*Diseasome:possibleDrug*" and "*Drugbank:target*" comparing with the horizontal concept taxonomies generated with "*Diseasome:omim*" and "*Drugbank:keggCompoundId*" that have only 2 levels of inheritance.

The Maximum Resolution shows the effectiveness of classification of a sub-taxonomy. The best scores obtained by "*Diseasome:omim*" (1.65) with Diseasome, and six properties (1.0) with DrugBank. The effectiveness of classification is contrary to the ability of generalizing instance properties in a taxonomy. A high effective classification may result in a weak ability of

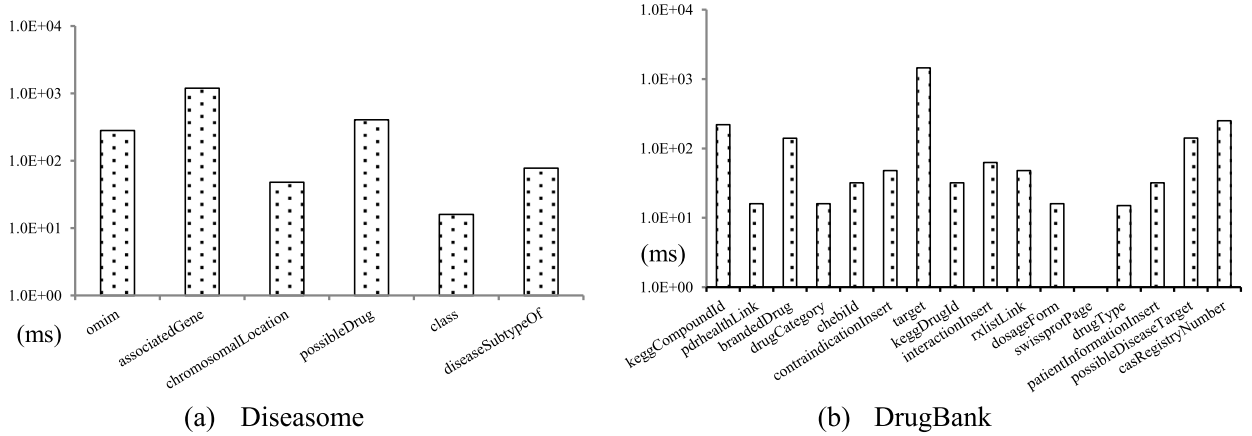



**Fig. 6.** Running time of building a sub-taxonomy with a single property.

**Table 3**
Results of conceptualizing disease and drug instances with multiple object properties. The highest IR and lowest MR scores are in bold.

| Object properties | Inheritance Richness | Maximum Resolution |
|---|---|---|
| (P1) Diseasome:omim | 1.00 | **1.65** |
| (P2) Diseasome:associatedGene | 1.66 | 2.05 |
| (P3) Diseasome:chromosomalLocation | 1.00 | 3.20 |
| (P4) Diseasome:possibleDrug | **15.22** | 3.96 |
| (P5) Diseasome:class | 0.96 | 175.54 |
| (P6) Diseasome:diseaseSubtypeOf | 1.00 | 2.28 |
| Object properties | Inheritance Richness | Maximum Resolution |
| (P1) Drugbank:keggCompoundId | 1.00 | 1.01 |
| (P2) Drugbank:pdrhealthLink | 1.00 | 1.03 |
| (P3) Drugbank:brandedDrug | 1.00 | **1.00** |
| (P4) Drugbank:drugCategory | 1.76 | 3.53 |
| (P5) Drugbank:chebiId | 1.00 | 1.02 |
| (P6) Drugbank:contraindicationInsert | 1.00 | **1.00** |
| (P7) Drugbank:target | **17.72** | 1.27 |
| (P8) Drugbank:keggDrugId | 1.00 | **1.00** |
| (P9) Drugbank:interactionInsert | 1.00 | **1.00** |
| (P10) Drugbank:rxlistLink | 1.00 | **1.00** |
| (P11) Drugbank:dosageForm | 3.24 | 1.82 |
| (P12) Drugbank:swissprotPage | 0.98 | 1.54 |
| (P13) Drugbank:drugType | 1.33 | 676.33 |
| (P14) Drugbank:patientInformationInsert | 1.00 | **1.00** |
| (P15) Drugbank:possibleDiseaseTarget | 7.56 | 2.45 |
| (P16) Drugbank:casRegistryNumber | 1.00 | 1.01 |

generalizing instance properties. For example, "*Diseasome:class*" get the highest MR score (175.54) and can best generalize the characteristic of the instances.

We tested the two data sets with different combinations of multiple object properties. We separated all combinations with different numbers of properties used. For example, choosing two properties of Diseasome may use "*Diseasome:omim*" and "*Diseasome:possibleDrug*" or "*Diseasome:possibleDrug*" and "*Diseasome:diseaseSubtypeOf*". We measured the average running times of creating a faceted taxonomy with a different number of properties shown in Fig. 7.

As Fig. 7 shows, the running time increases along with the increment of the number of properties. In Diseasome, the average running time increases from 338 ms up to 2032 ms with one property and six properties. In DrugBank, the average running time increases from 157 ms up to 2525 ms with one property and sixteen properties. We learned that the increments of the number of properties could cost more time for our algorithm to generate a faceted taxonomy since each property extracts the unique concepts for a facet and needs an additional job to build a sub-taxonomy [47].

The more facets a taxonomy contains, the better classification of the taxonomy. As Fig. 8 shows, the Maximum Resolution dramatically decreases when two sub-taxonomies are used. For example, the Maximum Resolution decreases from 31.45 to 1.65 with Diseasome, and decreases from 43.56 to 1.03 with DrugBank. When the number of facet used increases up to three, the Maximum Resolution decreases slightly. As it is designed for helping narrow down search space [48,49], the results show that using two facets is sufficient enough to meet users' needs to search the two data sets.

We have counted the top 500 important concepts in the generated faceted taxonomies with two data sets, and show the number

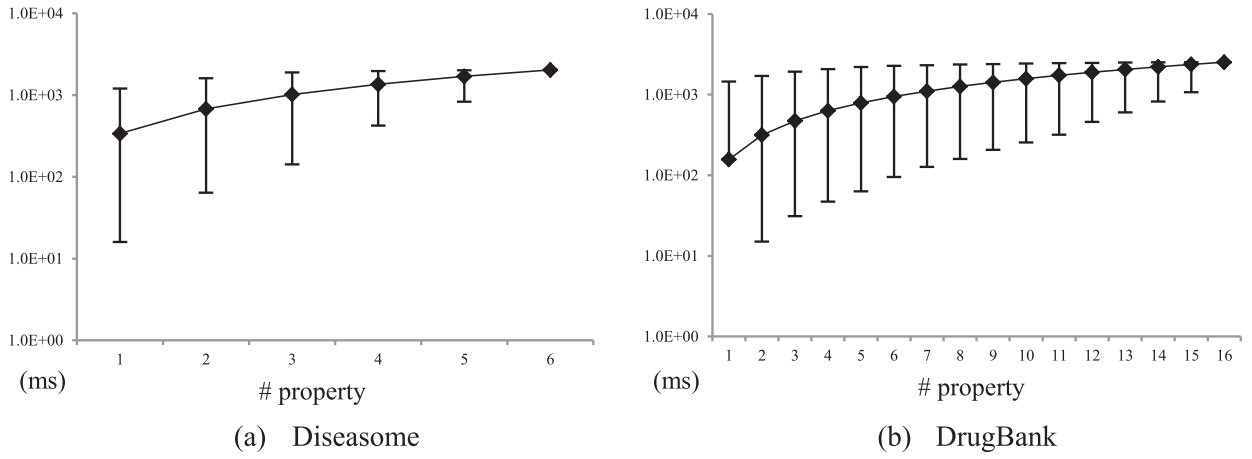



**Fig. 7.** Average running time of building faceted taxonomies with different facets (properties).

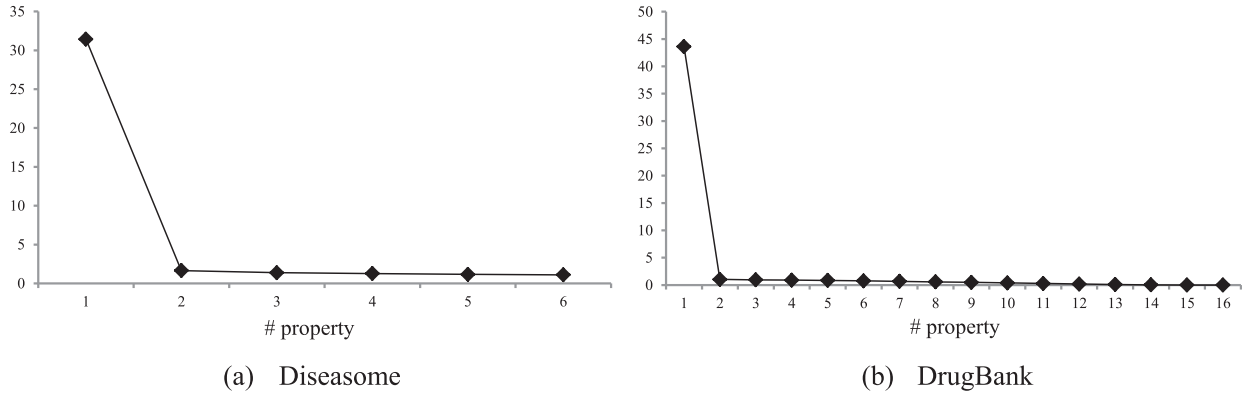



**Fig. 8.** Maximum Resolution scores with different facets (properties).

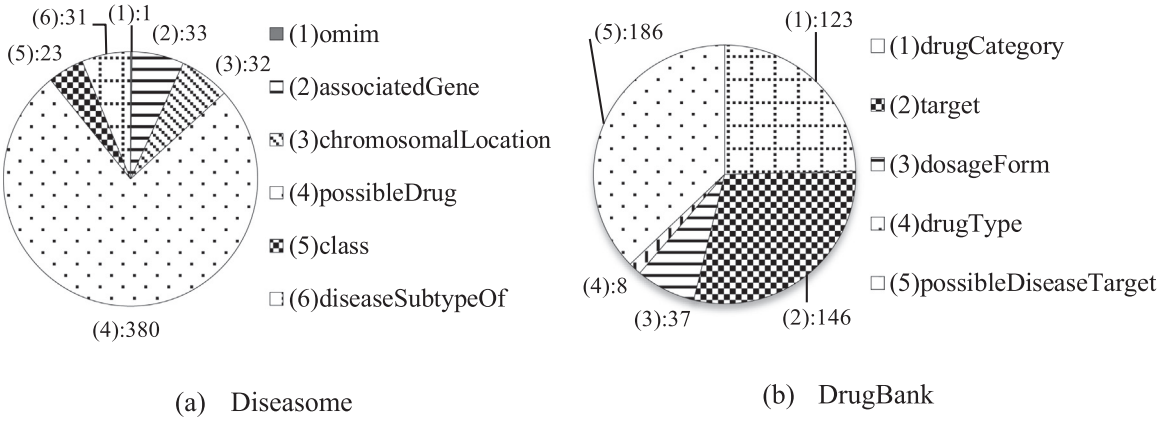



**Fig. 9.** Number of top 500 important concepts in each sub-taxonomy in a faceted taxonomy.

of important concepts of each property in Fig. 9. As Fig. 9(a) shows, "*Diseasome:possibleDrug*" contains the 380 out of 500 important concepts in Diseasome. "*Diseasome:associateGene*" (33), "*Diseasome:chromosomalLocation*" (32), "*Diseasome:class*" (23), and "*Diseasome:diseaseSubtypeOf*" (31) contain almost the same number of important concepts. Also, "*Drugbank:possibleDiseaseTarget*" (186), "*Drugbank:target*" (146), and "*Drugbank:drugCategory*" (123) have the largest number of important concepts in Drugbank. "*Drugbank:dosageForm*" (37) and "*Drugbank:drugType*" (4) contain relatively small numbers of important concepts. Fig. 9 illustrates the most important sub-taxonomies of the facets "*possibleDiseaseTarget*" (or "*possibleDrug*"), "*target*", and "*associateGene*", which are recognized as important properties adapted for the applications [50–53]. These sub-taxonomies, which cover a large number of instances, are recommended to the user who is unfamiliar with the data sets but wants to get the most information.

## 7. Discussion and conclusion

The increasing popularity of publishing LD sets warrants the construction of concept taxonomies from the datasets without ontologies. Instead of building a taxonomy to classify instances from a specific perspective, this study proposes a faceted taxonomy that classifies instances from multiple perspectives based on object properties. However, though the proposed method achieves encouraging results in terms of efficiency and effectiveness with two experiments, there are still some limitations that need to be addressed.

Firstly, multiple inheritance could exist in sub-taxonomies. The proposed method builds sub-taxonomies based on the concept definition and validation rules in Section 3, which assumes that the object property values of the instances are distributed following the transitive inheritance [54]. Hence, the method can search a possible number of objects to find the best sub-concept in each iteration. However, the object properties for instances may not follow such distribution (e.g., multiple instantiation) and cause multiple inheritance, which was observed in Task 1 with YAGO2. To solve this problem, one possible solution is to add a greedy equation that only selects the most possible sub-concept with the highest score for Eq. 4. However, this will increase the level of discrepancy in real instances distribution in the data and the hypothetical instances distribution in the generated taxonomy. Currently, this trade-off is controlled by $\varphi$ in Eq. 4, however, a more efficient way of automatically setting such parameter by balancing the rate of multiple inheritance and the difference of two distribution is desired for the future.

Secondly, the difficulty in understanding concept hierarchies: a concept hierarchy is constructed based on instance value

distribution, and thus, the subsumption relation in the taxonomy sometimes cannot be interpreted as "*is_A*" relation. For example, in the taxonomy with the facet "*Diseasome:possibleDrug*" for Diseasome in Task 2, it is hard to understand the concept "*Drug:DB00898*" (Ethanol) that is the super concept of "*Drug:DB03929*" (D-Serine). However, the two concepts have a subsumption relation in extension, since "*Drug:DB03929*" can treat "*Disease:2666*" (Hyperekplexia and spastic paraparesis), and "*Drug:DB00898*" can treat "*Disease:2666*", "*Disease:372*"(Epilepsy), and "*Disease:2312*" (Epilepsy, juvenile myoclonic, 606904). While there exists a difficulty in understanding the concept hierarchy, such organization can efficiently reduce the search space in a navigation [12]. For example in a faceted search, if a user wants to find diseases cured by "*Drug:DB03929*", zooming "*Drug:DB00898*" (i.e., zoom-in point [12]) into "*Drug:DB03929*" can reduce three diseases into only one disease. Therefore, the generated taxonomy is considered to be organized more for entity navigation or search than as an ontology for a knowledge base.

Thirdly, entity recognition and mapping needs to be considered for taxonomy generation. The proposed method uses shared common entities to build a taxonomy, thus requires unique entities to be distinguished and identical entities to be mapped. Blank nodes are internal anonymous resources without identifiers and are hard to be distinguished [55]. Even though bank nodes are suggested to be avoided for data publisher, only 44.1% of LD datasets do not publish blank nodes [55]. The process to identity two blank nodes is based on the comparison of the incoming and out-coming links of the two nodes. Such comparison is very expensive for large dataset, and remains an open issue. While our method does not provide a solution for dealing with blank nodes and we did not observe any errors caused by the blank nodes in our experiment, this issue needs to be studied for the proposed framework. Similar to distinguishing the different nodes, identical entities from heterogeneous data resources need to be mapped. This is an important issue recognized as entity mapping in linked data [7]. We will endeavor to strengthen the framework in Section 3.3 by dealing with above two issues in our future work.

Lastly, a facet can be defined in a more flexible way that provides a more powerful navigation for entities. In our method, a facet is defined in relation to one object property instead of multiple properties, since we considered that it is more helpful in practice to support faceted search with a sub-taxonomy from one perspective. We gained this understanding from our experiments in which we created a misleading organization of instances for a facet defined by two contradictive properties (e.g., "*sideEffect*" and "*possibleDiseaseTarget*"). However, we also acknowledge that some facets generated from combinations of properties may provide more information than from a singular property. One possible solution might be to define a facet by a sematic meaning instead (e.g., "activeIngredient" and "activeMoiety"). Such definition may require a system that pre-clusters semantic meanings of object properties. Another practical solution is to allow users to define a personalized facet and generate the taxonomies on-the-fly [56].

The faceted taxonomy proposed in this study supports multiple perspectives to organize instances, and this is a useful feature for browsing instances with multiple properties, such as biomedical linked data sets. Linked Life Data[4] already provides faceted search with limited categories as filters. One possible application of the proposed method is to enrich such categories to improve search. In addition, the implementation issues of the faceted navigation, such as how to dynamically visualize the taxonomies as the user moves along in the structure, should be further investigated in the future.

## Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2017R1C1B1009595). This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and future Planning (NRF-2014R1A2A1A11049728).

The authors appreciate the valuable comments from Jinhyun Ahn, Hyunwhan Joe, Sungin Lee, Nadarajan Gaya Rachael Sze Nga Wong, and Victoria Ngo.

## References

[1] D. Blum, S. Cohen, Generating RDF for application testing, in: Proceedings of the 9th International Semantic Web Conference ISWC 2010, Citeseerpp. 105, 2010.
[2] L. Ding, D. DiFranzo, A. Graves, J.R. Michaelis, X. Li, D.L. McGuinness, J.A. Hendler, TWC data-gov corpus: incrementally generating linked government data from data.gov, in: Proceedings of the 19th international conference on World wide web, ACM, Raleigh, North Carolina, USApp. 1383–1386, 2010.
[3] M. Zhu, Z. Gao, J.Z. Pan, Y. Zhao, Y. Xu, Z. Quan, TBox learning from incomplete data by inference in BelNet+, Knowl.-Based Syst 75 (2015) 30–40.
[4] I. Horrocks, Ontologies and the semantic web, Commun. ACM 51 (2008) 58–67.
[5] F. Baader, The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, New York, NY, USA, 2003.
[6] B. Chandrasekaran, J.R. Josephson, V.R. Benjamins, What Are Ontologies, and Why Do We Need Them?, IEEE Intell. Syst. 14 (1999) 20–26.
[7] C. Bizer, T. Heath, T. Berners-Lee, Linked data-the story so far, Int. J. Sem. Web Inf. Syst. 5 (2009) 1–22.
[8] R. Parundekar, C.A. Knoblock, J.L. Ambite, Linking and building ontologies of linked data, in: Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I, Springer-Verlag, Shanghai, Chinapp, 2010, pp. 598–614.
[9] J. Völker, M. Niepert, Statistical Schema Induction, in: G. Antoniou, M. Grobelnik, E. Simperl, B. Parsia, D. Plexousakis, P. De Leenheer, J. Pan (Eds.), The Semantic Web: Research and Applications, Springer, Berlin Heidelberg, 2011, pp. 124–138.
[10] C. Brewster, Y. Wilks, Ontologies, Taxonomies, Thesauri Learning from Texts, in: M. Deegan (Ed.)The Keyword Project: Unlocking Content through Computational Linguistics (2004), 2004.
[11] Y. Tzitzikas, N. Manolis, P. Papadakos, Faceted exploration of RDF/S datasets: a survey, J. Intell. Inf. Syst. 48 (2017) 329–364.
[12] G.M. Sacco, Y. Tzitzikas, Dynamic Taxonomies and Faceted Search: theory, Practice, and Experience (Incorporated), Springer Publishing Company, New York, NY, USA, 2009.
[13] E. Oren, R. Delbru, S. Decker, Extending faceted navigation for RDF data, in: Proceedings of the 5th international conference on The Semantic Web, Springer-

[4] http://linkedlifedata.com/

Verlag, Athens, GApp. 559–572, 2006.
[14] N. Zong, S. Nam, J.-H. Eom, J. Ahn, H. Joe, H.-G. Kim, Aligning ontologies with subsumption and equivalence relations in Linked Data, Knowl.-Based Syst. 76 (2015) 30–41.
[15] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: a nucleus for a web of open data, Sem. Web (2007) 722–735.
[16] J. Hoffart, F.M. Suchanek, K. Berberich, G. Weikum, YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia, Artif. Intell. 194 (2013) 28–61.
[17] K.-I. Goh, M.E. Cusick, D. Valle, B. Childs, M. Vidal, A.L. Barabási, The human disease network, Proc. Natl. Acad. Sci. 10 (4) (2007) 8685–8690.
[18] D.S. Wishart, C. Knox, A.C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, M. Hassanali, DrugBank: a knowledgebase for drugs, drug actions and drug targets, Nucleic Acids Res. 36 (2007) D901–D906.
[19] Resource Description Framework (RDF) Model and Syntax Specification, in, 1999.
[20] M. Hazman, S.R. El-Beltagy, A. Rafea, A survey of ontology learning approaches, Database 7 (2011) 6.
[21] G. Piateski, W. Frawley, Knowledge Discovery in Databases, MIT Press, Cambridge, MA, USA, 1991.
[22] J. Han, Y. Cai, N. Cercone, Knowledge discovery in databases: an attribute-oriented approach, in: Proceedings of the 18th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Incpp, 1992, pp. 547–559.
[23] I. Bedini, C. Matheus, P.F. Patel-Schneider, A. Boran, B. Nguyen, Transforming XML Schema to OWL Using Patterns, in: IEEE Proceedings of the Fifth International Conference on Semantic Computing, IEEE Computer Society, 2011, pp. pp. 102–109.
[24] N. Lammari, I. Comyn-Wattiau, J. Akoka, Extracting generalization hierarchies from relational databases: a reverse engineering approach, Data Knowl. Eng. 63 (2007) 568–589.
[25] S. Lee, S.-Y. Huh, R.D. McNiel, Automatic generation of concept hierarchies using WordNet, Expert Syst. Appl. 35 (2008) 1132–1144.
[26] E. Drymonas, K. Zervanou, E.G.M. Petrakis, Unsupervised ontology acquisition from plain texts: the OntoGain system, in: Proceedings of the 15th international conference on Applications of natural language to information systems, Springer-Verlag, Cardiff, UK, 2010, pp. 277–287.
[27] J.D. Knijff, F. Frasincar, F. Hogenboom, Domain taxonomy learning from text: the subsumption method versus hierarchical clustering, Data Knowl. Eng. 83 (2013) 54–69.
[28] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, R. Krishnapuram, A hierarchical monothetic document clustering algorithm for summarization and browsing search results, in: Proceedings of the 13th international conference on World Wide Web, ACM, New York, NY, USA, 2004,. pp. 658–665.
[29] P. Cimiano, A. Hotho, S. Staab, Learning concept hierarchies from text corpora using formal concept analysis, J. Artif. Int. Res. 24 (2005) 305–339.
[30] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc, Upper Saddle River, NJ, USA, 1988.
[31] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: KDD workshop on text mining, Bostonpp, 2000, pp. 525–526.
[32] N. Zong, D.-H. Im, S. Yang, H. Namgoon, H.-G. Kim, Dynamic generation of concepts hierarchies for knowledge discovering in bio-medical linked data sets, in: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ACM, Kuala Lumpur, Malaysiapp, 2012, pp. 1–5.
[33] M.-S. Paukkeri, A.P. García-Plaza, V. Fresno, R.M. Unanue, T. Honkela, Learning a taxonomy from a set of text documents, Appl. Soft Comput. 12 (2012) 1138–1148.
[34] M. Sanderson, B. Croft, Deriving concept hierarchies from text, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Berkeley, California, USA, 1999, pp. 206–213.
[35] D.J. Lawrie, W.B. Croft, Generating hierarchical summaries for web searches, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, Berkeley, CA, USA, 2003, pp. 457–458.
[36] P. Heymann, H. Garcia-Molina, Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems, Stanford InfoLab, Stanford, CA, USA, 2006.
[37] J.F. Sowa, Knowledge Representation: Logical, Philosophical and Computational Foundations, Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2000.
[38] S. Bechhofer, F.v. Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein, OWL Web Ontology Language Reference, in: M. Dean, G. Schreiber (Eds.), 2004.
[39] A. Taivalsaari, On the notion of inheritance, ACM Comput. Surv. 28 (1996) 438–479.
[40] J. Carroll, I. Herman, P.F. Patel-Schneider, OWL 2 Web Ontology Language RDF-Based Semantics (Second Edition), in: M. Schneider (Ed.), 2012.
[41] K. Dellschaft, S. Staab, On how to perform a gold standard based evaluation of ontology learning, in: Proceedings of the 5th International Conference on the Semantic Web, Springer-Verlag, Athens, GA, 2006, pp. 228-241.
[42] H.A. Santoso, S.-C. Haw, Z.T. Abdul-Mehdi, Ontology extraction from relational database: concept hierarchy as background knowledge, Knowl.-Based Syst. 24 (2011) 457–464.
[43] L. Drumond, R. Girardi, Extracting ontology concept hierarchies from text using markov logic, in: Proceedings of the 2010 ACM Symposium on Applied Computing, ACM, 2008, pp. 1354–1358.
[44] S. Dasgupta, D. Dinakarpandian, Y. Lee, A Panoramic Approach to Integrated Evaluation of Ontologies in the Semantic Web, in: EON, 2007, pp. 31-40.
[45] G. Cheng, W. Ge, H. Wu, Y. Qu, Searching Semantic Web Objects Based on Class Hierarchies, in: LDOW, 2008.
[46] E. Demidova, I. Oelze, W. Nejdl, Aligning freebase with the YAGO ontology, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, ACM, 2013, pp. 579–588.
[47] B. Zheng, W. Zhang, X.F.B. Feng, A survey of faceted search, J. Web Eng. 12 (2013) 041–064.
[48] J. Koren, Y. Zhang, X. Liu, Personalized interactive faceted search, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 477–486.
[49] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, S. Yogev, Beyond basic faceted search, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, ACM, 2008, pp. 33–44.
[50] W. Wang, S. Yang, X. Zhang, J. Li, Drug repositioning by integrating target information through a heterogeneous network model, Bioinformatics 30 (2014) 2923–2930.
[51] X.A. Qu, R.C. Gudivada, A.G. Jegga, E.K. Neumann, B.J. Aronow, Inferring novel disease indications for known drugs by semantically linking drug action and disease mechanism relationships, BMC Bioinform. 10 (2009) S4.
[52] D. Sonntag, P. Wennerberg, S. Zillner, Applications of an ontology engineering methodology accessing linked data for dialogue-based medical image retrieval, AAAI, 2010.
[53] N. Zong, H. Kim, V. Ngo, O. Harismendy, Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations, Bioinformatics (2017) btx160.
[54] C. Rosse, J.L. Mejino, A reference ontology for biomedical informatics: the Foundational Model of Anatomy, J. Biomed. Inform. 36 (2003) 478–500.
[55] A. Mallea, M. Arenas, A. Hogan, A. Polleres, On blank nodes, in: International Semantic Web Conference, Springer, 2011, pp. 421–437.
[56] S. Basu Roy, H. Wang, G. Das, U. Nambiar, M. Mohania, Minimum-effort driven dynamic faceted search in structured databases, in: Proceedings of the 17th ACM conference on Information and knowledge management, ACM, 2008, pp. 13–22.