State of the Art Exploration Systems for Linked Data: A Review

Karwan Jacksi
Computer Science Department
University of Zakho
Zakho, Iraq

Nazife Dimililer
Information Technology Department
Eastern Mediterranean University
Gazimagusa, N. Cyprus

Subhi R. M. Zeebaree
Department of IT, Akre Technical College
Duhok Polytechnic University
Akre, Iraq

Abstract—**The ever-increasing amount of data available on the web is the result of the simplicity of sharing data over the current Web. To retrieve relevant information efficiently from this huge dataspace, a sophisticated search technology, which is further complicated due to the various data formats used, is crucial. Semantic Web (SW) technology has a prominent role in search engines to alleviate this issue by providing a way to understand the contextual meaning of data so as to retrieve relevant, high-quality results. An Exploratory Search System (ESS), is a featured data looking and search approach which helps searchers learn and explore their unclear topics and seeking goals through a set of actions. To retrieve high-quality retrievals for ESSs, Linked Open Data (LOD) is the optimal choice. In this paper, SW technology is reviewed, an overview of the search strategies is provided, and followed by a survey of the state of the art Linked Data Browsers (LDBs) and ESSs based on LOD. Finally, each of the LDBs and ESSs is compared with respect to several features such as algorithms, data presentations, and explanations.**

Keywords—Exploratory Search System; Linked Data; Linked Data Browser; Semantic Web

I. INTRODUCTION

Over the time, World Wide Web (WWW) has made data sharing online an easy task for all users which created an immense amount of data available and transformed the Web to a massive semi-structured database [1]. Hence, retrieving relevant information efficiently from this huge dataspace is a huge challenge. Usually, search engines are used to retrieve data from the current Web, however, to get accurate retrievals, very efficient indexing and seeking strategies and increasingly more complicated heuristics must be employed.

The search approaches have two main categories: lookup (or keyword-based) and exploratory search [2]. Lookup search approaches, typically, have database systems in the background and information is retrieved based on the keywords given. In this widely used search approach, the data is mainly textual documents, and the search items are known [3].

Exploratory search category is a specific information searching approach where the user targets and objectives are not necessarily known during the searching process. The users of this category concentrate on learning and investigation rather than fact retrievals and query answering. They compare, analyze and discover new concepts for the retrieved information [4].

Finding information on the existing Web, also known as the syntactic Web, is based on keyword search, which has limited recall and precision due to synonyms, homonyms…etc. of the keywords. Therefore, the quality of obtained results is rather poor. To enhance it, annotations are added to the contents of the syntactic Web forming the Semantic Web (SW) [1] [5].

The SW is an extension of the syntactic Web through standards by the W3C[1], where data is given well-defined meaning, can be understood by machines, thereby allowing machines and people to work in collaboration [6]. It uses W3C standards such as Resource Description Framework (RDF) to support unified data formats and exchange protocols over the Web so that the meanings of data are understood by machines. This technology increases the efficiency of search engines by enabling machine-driven data processing [5]. The RDF is known as the HTML of the SW and it is purely an XML language. It is the W3C standard used to represent information in the Web and describes Web resources as *<subject, predicate, object>* components, which is known as triples. The structure of interlinking resources of the Web is extended by RDF to use Uniform Resource Identifiers (URIs) to indicate the connections among resources forming a directed and labeled graph of structured and semi-structured data.

Ontologies are essential units of the SW infrastructure and more often known as the backbone of SW [7]. Web Ontology Language (OWL) and RDF Schema (RDFS) are the knowledge representation languages and data models recommended by W3C where fundamental elements for the description of ontologies are given[2].

In the SW, the Web contents enhanced with data annotation to be linked with each other forming the Web of data which makes it possible for relevant data to be found once only a subset is given. The terms SW and Linked Data (LD) have been coined by Berners-Lee and describes the LD as "the SW done right"[3] [8].

LD term indicates a couple of stages or rules to publish and link structured data on the Web. The stages have been identified by Lee in his notes about design issues on Web architecture and shortly became the principles of LD [9]. These stages are 1) things should be identified with URIs; 2) HTTP URIs should be used for these URIs so that people can reuse them; 3) use standards such as RDF and SPARQL[4] when providing information for users looking up URIs; 4) link to other URIs so that further things are discoverable [10].

In fact, LD is to publish data on the Web wherein the data is understandable and processable by machines, its meaning is defined clearly, and is linked to/from other external data sets. Typically, untyped hyperlinks are used to connect HTML data with one another in the Hypertext Web, whereas LD relies on RDF data so to build typed links to connect things worldwide forming Web of Data[5] [11]. When the LD is available under an open license, it is called Linked Open Data (LOD).

The main target of this work is to present the current state of the art LD based exploration systems and to compare them regarding their features and underlying technologies so as to identify their strengths and weaknesses. A broad purpose of our study is to show to what extent the subject of exploration systems for LD has arrived, what kind of algorithms and technologies are used, and which Linked Datasets are most utilized. This is of great importance for the researchers in this field to effortlessly find and compare the above points.

To the best of our knowledge, there is only one other survey study concerning to the topic of exploration systems based on LD presented by [12], where Linked Data Browsers (LDBs), Linked Data Recommenders (LDRs) and Exploratory Search Systems (ESSs) are to some extent reviewed. However, none of the LDBs examined in our paper are in the found study, since in this paper only LDBs from 2012 to this point of time are considered, while the LDBs studied in the found research are up to the year 2011. Conversely, LDRs are explained in the found paper but not in our paper.

This research is conducted by searching and collecting articles from leading sources which are either found from databases such as Web of Science (WoS) or Scopus, or from SW Conferences and Challenges. 28 systems were selected and reviewed in details, 16 systems were chosen to be included in this paper. The selection of papers to be included in this paper based mainly on the authoritative paper sources (publishers) and on the publication date (most recent systems are included).

This paper is organized as follows; Section 2 reviews each of the LDBs in details; Section 3 presents a comprehensive review for the existing LD based ESSs; extensive comparisons of the systems are discussed in section 4, and finally we draw some conclusion in Section 5.

II.   LINKED DATA BROWSER (LDB)

In the recent years, the use of LOD has notably increased on the Web. Nevertheless, it remains challenging to be used by users, particularly lay-users. Since the SW foundation, the interaction with LD and its visualization have been documented as issues [13]. Since then, a plethora of LDBs allowing users to understand, explore and interact with the massive LOD have been developed. Some of which, such as Tabulator[6] and Explorator[7], present data as pairs in tables and others, such as Graphity[8] and RelFinder[9], are graph-based browsers while other approaches, such LODmilla, combine features from both. In this section, we will review some of the states of the art LDBs to give an overview of their functionalities and features.

A. LodLive

LodLive is an LDB which uses standards of the LD to navigate RDF resources with the aim of spreading the fundamentals of LD in dynamic visual graphs through a user-friendly interface [14]. Within the application, resources located in different endpoints can be linked so as to discover unexpected connections. Also, inverse relations can be navigated even for different endpoints.

LodLive, via Sesame Framework, is able to parse RDF data even when they do not reside in an SPARQL endpoint. This can be done by generating a graph remotely to store temporarily the requested resources for making queries. During the initial stages of ontology definition, LodLive is a useful tool to verify the validity of an RDF schema and visually pick a solution among several.

JavaScript application layer has been used in the system without the necessity of any application server by parsing the JSON formatted retrievals of JSONP (JSON with Padding) calls from endpoints, and presenting them in an HTML5 web page.

B. CubeViz

CubeViz is a faceted navigation browser and an extension of OntoWiki[10] tool to visualize statistical data represented in RDF [15]. Usually, statistical data sets known as data cubes or basically cubes are distributed as spreadsheets or bi-dimensional matrices. To extract triples from these spreadsheets, further tools such as *CSV2DataCube*[11] (a plug-in extension in OntoWiki) are needed. These tools, usually, use RDF Data Cube vocabulary[12] which is an *SDMX*[13] standard and state of the art in representing statistical data in RDF. Thus, tools distributing multidimensional statistics on the web often use RDF Data Cube vocabulary so that they are able to be linked to related RDF datasets.

CubeViz is built so as to conceal the complication of RDF Data Cube vocabulary for users and to encourage the navigation and investigation of cubes. It utilizes the RDF Data Cube vocabulary to produce a faceted browsing for statistical data that is able to visualize and filter interactive explanations

---

[4] http://www.w3.org/TR/rdf-sparql-query
[5] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210

[6] http://www.w3.org/2005/ajar/tab
[7] http://www.tecweb.inf.puc-rio.br/explorator
[8] http://graphity.org
[9] www.visualdataweb.org/relfinder.php
[10] OntoWiki: a tool provides cooperative support in creating RDF knowledge bases and their maintenance and publication.
[11] https://github.com/AKSW/csvimport.ontowiki
[12] http://purl.org/linked-data/cube
[13] https://sdmx.org

in diverse graph types such as charts and scatter plots. The JavaScript library *Highcharts*[14] has been used for rendering visualized charts for client side users, while on the server side, a PHP class called *PHPlot*[15] is used to render charts.

To use CubeViz, a preferred SPARQL endpoint and data structure have to be chosen. Later, aggregated modules which are defined as instances have to be selected. These instances reference various types of module properties.

C. Linked Data Visualization Model (LDVM)

LDVM can be used to quickly make representations of RDF data visually [16]. It permits users to connect to and extract data from different datasets with different visualization techniques. The conceptual framework of the model is based on Data State Reference Model (*DSRM*) model offered by [17] as a result of adopting its stages' operations, names and transformations so as to fit the LD environment.

The LDVM model may well be seen as a pipeline and it consists of four stages to process data, and three transformation operations in between these stages. The stages are *RDF data:* the raw data, *Analytical extraction*: to extract data from a previous stage, *Visual abstraction*: the visualizable data and *View*: present the data in different views. While the operations in between these stages are: *Data transformation*, *Visualization transformation, and Visual mapping transformation*. So, the model can be sectioned into two core sections: data space and visual space.

To proof the LDVM concept, a prototype and a useful RDF data browser is built based on the LDVM model called *LODVisualization*[16]. It gives multiple visualizations while browsing RDF datasets. Any endpoint supporting JSON and SPARQL 1.1 is compatible with this prototype. Several visualizations, such as charts, tables, treemaps, can be shown.

The server side of *LODVisualization* is written in Python, whereas the client-side built using HTML, CSS, and JavaScript mostly using *D3.js* library[17] and *InfoVis Toolkit*[18]. Google App Engine (GAE)[19] is used as a cloud computing platform. To avoid duplicated queries when users choose different visualization views, a cache system using GAE *Datastore* and *Blobstore* is used on the server so the performance and scalability are improved.

D. LODmilla

LODmilla is a generic LDB to discover and edit LOD with the ability to combine the features of textual and graph based LOD browsers [18]. The web application provides viewing and searching LOD graphs as well as other browsing services such as editing and reorganization utilities for data located in various linked datasets. The system uses a dedicated server for its search processes so as to support caching and rapid triple loading.

To load triple data, the approach switches through two methods employed: the SPARQL-based query and actionable URIs. Through the Jena toolkit at the server side, RDF data is able to get parsed into several serializations including JSON. As a result, a wide range of LOD datasets is possible to be used concurrently despite the configurations of datasets at the client side, hence the graph data is fetched using the actionable URIs method.

Graph traversal techniques have been applied in the system to find paths between notes. It starts from both sides of nodes using basic heuristics excluding connections having huge connections so as to discover next level paths quickly.

LODmilla can be used as an LD editor since it provides utilities to add/remove nodes and/or draw/cut out edges between the nodes in the graph. This editing facility is useful for finding quick solutions for incomplete graphs.

E. SView

Smart View, or SView, is a system that allows users to navigate entity descriptions for LD perceptively [19]. It uses Lenses, a group of organized features, to cluster and organizes entity descriptions so as to support users find related information easily. Moreover, it provides different methods to discover related elements, such as discoveries depending on link pattern and discoveries based on entity similarity.

Through SView, registered users are able to create their personal lenses so that they personalize how the features of LD are grouped and ordered. Then, they are able to share the created lenses globally so that other users can use or customize them for their needs. Accordingly, the system is able to provide a set of already created lenses for all users through leveraging these personalized lenses of the registered users and offer them to other users globally.

In the system, if a value of the current entity feature is a new entity, this is called a link. While discovering an entity description over lenses, users follow the link to the entity to discover related entities. Along with this method, two other mechanisms for discovering related entities are offered by the system: link pattern discovery and similarity based entity recommendation.

In the link pattern discovery method, having more than one link from an entity to another entity is possible in LD. The set of all links from between them is called entity's link pattern. Linked entities having a shared link pattern are clustered for exploration. Consequently, descriptions of entities having shared link patterns can be navigated by users. As a result, the system is able to provide pivoting from an entity to linked set of entities by transforming lenses to tables where the rows in the table are entities and the columns are features.

In SView, similar entities can be recommended using several approaches. The system uses surfing history of entire users to recommend similar entities. One of the approaches is to discover entities having mutual types to the entity in use. Thus, more mutual types, more recommended entity.

---

[14] http://www.highcharts.com
[15] http://sourceforge.net/projects/phplot_`
[16] http://lodvisualization.appspot.com
[17] http://d3js.org
[18] http://philogb.github.io/jit
[19] https://en.wikipedia.org/wiki/Google_App_Engine

F. LD Viewer

Reference [20] implemented an adjustable framework and LDB that is integrating several tools to generally provide an ease of use LD explorations. The framework objective is to deliver a unified interface with a set of powerful features so it can without difficulty be adopted by several linked datasets.

The important component of the system is the property table which presents all the obtainable properties of the explored entity. It proposes forward and reverses properties for each explored entity together with pagination facility for reverse properties that have a huge amount of values.

Labels are presented rather than resource URIs so as to get a better legibility of the property table. Live previews for each link is also available in the system, by showing a brief preview of links in the property table. OpenStreetMap API[20] is used to generate maps for entities having location information.

For each triple in the property table, there is a clickable action (or set of actions) related to it when integrated conditions are met. These actions differ from each other due to the nature of the triple. For instance, using one of the actions, the user can make annotations to DBpedia dataset if the action is applicable for such triple.

The architecture of the application takes advantages from Model View Controller (MVC) software architectural pattern, it has been built primarily with AngularJS[21] framework, and modules of *Jassa library* [22] (JAvascript Suite for Sparql Access) are reused. To adopt the interface for datasets, there is no need to understand the core layer of the system.

Theoretically, to set up the framework to a dataset, three levels have to be concerned: 1) the triple store, which is reachable via SPARQL query language; 2) the server side to let the JavaScript implementation to work; and 3) the client side, which is implemented purely in JavaScript.

G. DBpedia Mobile Explorer

DBpedia Mobile Explorer is a mobile application to explore and visualize LD [21]. The framework is implemented to operate as a generic and a domain specific explorer for DBpedia dataset. In the generic exploration case, concise details of a resource been explored are given, a graph is formed for the connected nodes to the resource, and a table is created to present all important materials. In addition, all of the related categories to the resource are fetched from the DBpedia and listed to the user so that they can explore these categories and find other resources contained within that category. The specific domain case drives users to only explore resources of a specific domain such as music or films. Therefore, it is infeasible to explore resources beyond the specified domain rather than presenting concise details of them.

MVC software pattern is considered in the framework. The model in MVC contains classes with reference to resources, and a parser to fetch the classes from the resources received by DBpedia. While controller and view, which are the primary units of the framework, have been designed independently of models so it can be reused with diverse modules.

For the domain specific exploration, RDF:type property is examined prior resource visualization so as to exclude out of domain resources and to only generate short details for them. The domain to be specified is not fixed and it can be specified by domain experienced users. The application is considered for DBpedia dataset, but it is able to be applied to other linked datasets as well as it depends on RDF and SPARQL.

H. DBpedia Atlas

DBpedia Atlas is a web application that allows users to browse classes, instances, and relationships of DBpedia dataset in an interactive and map-like visualization. The application is built by adopting the efforts of [22] on Gosper Treemaps to Linked datasets with a structure of the hierarchical ontology. Therefore, it transforms entities of the DBpedia dataset to be shown as maps creating a hierarchy of areas along with their ontological class. A group of thematic maps and accessory charts are then created on top of these maps creating an atlas to describe diverse features of the dataset. The aim of the project is to give inexpert SW users a way to explore the dataset and understand its fundamental features.

System interface includes three main modules: map, search box, and infobox modules. The complete instances and classes of DBpedia are delivered to users through the map. Hence, the DBpedia root, e.g. *owl:Thing*, forms the primary map and the connected entities to the root form the regions inside the map. A minor island (separated from the primary map) is also created for untyped instances. For each region, there is a label, but smaller regions have to be zoomed so as to make the labels visible. However, to give users a further identification of basic categories, a manual identification for some of the instances are set with permanently visible labels. When an instance is selected from the map, all its details are presented in the infobox, and all its linked instances are distributed by way of red dots on the map. Several thematic maps are also available within the application, for instance, the map can illustrate the depth of the classes in the ontology hierarchy by notions of the darker colors the deeper in the hierarchy [23].

III. EXPLORATORY SEARCH SYSTEM (ESS)

ESS forms a special category of seeking information on the Web with the purpose of revealing related information to the searcher along with retrievals of what have been searched for. With this search category, the final targets of the search are not known, and the goal itself is not defined. Therefore a set of additional activities, for instance, learning, exploration and evaluation, are accessible through this category [24]. The history of this category has begun with the Exploratory Search Interface XSI 2005 Workshop [25]. The machine data processing nature of Semantic Web (SW) proposes overwhelming possibilities for search engines specifically for ESS [26]. In this section, a review of the existing ESSs based on the SW technologies is presented. A more detailed survey is presented by [27].

---

[20] http://wiki.openstreetmap.org/wiki/API
[21] https://angularjs.org
[22] http://aksw.org/Projects/Jassa.html

A. Yovisto

The authors of this system address the issue of deploying explorative search for video data using SW technology and LD [28]. They confirm how LOD can be used to enable an ESS for video data. Yovisto is a search engine dedicated to academic lecture and conference videos. It provides an exploratory search property based on LD. Through Yovisto's time-dependent index, the ability to search in video contents is viable. This feature makes Yovisto different from other video search systems. Machine analysis methods e.g. smart character recognition and scene detection approaches are used to generate metadata. Additionally, annotations and time-based tags can be given by users within their comments. The index of Yovisto is generated through fine granular time-based metadata

The system utilizes LOD resources while providing search results; it presents additional materials that are relevant to searchers' query semantically. DBpedia dataset is used to provide the relevant information. Since DBpedia dataset is enormous and there is a huge amount of data for each entity, allocating all the related information for each entity is a very heavy process. Therefore, statistically based heuristics are created to select the best relevant retrievals for the searchers' query. Running online queries against DBpedia is a tedious task and consequently falls in performance issues particularly if many related resources need to be returned. Therefore, an offline processing has been set up to process every term in advance.

B. Semantic Wonder Cloud (SWOC)

SWOC is an ESS that uses DBpedia as its dataset allowing surfers to navigate through the dataset using its semantic connections [3]. On top of utilizing the DBpedia semantic properties, the system takes advantage of external resources, like search systems and social tagging approaches, to find and present the DBpedia resources. So, this hybrid feature makes the system distinctive among other ESSs to rank the resources. The system consists of two main components: back-end, where the links between pairs of DBpedia resources are computed, and a front-end, where the attained data from the back-end is presented in a flash based interface.

For the ranking of the results of DBpedia resources, DBpediaRanker is used, which is an algorithm to compute the similarity of the resources compared to the initially queried resource. Thus, when the DBpedia resources are explored the DBpediaRanker uses external sources to compute and find the value of similarity for each pair of resources founded in the graph exploration.

In the front-end, the topic is selected, using the DBpedia lookup service[23], from a drop-down list of resource labels. The selected resource is returned together with its 10 most related resources forming a star network topology like a graph. The central node is the selected resource, while the surrounding nodes are the most related resources to the central one. The surrounding resources have different sizes depending on the similarity value; the bigger, the most similar to the central node. An info box to the right of the graph is presented showing a short description of the central node.

C. Lookup Explore Discover (LED)

LED is an ESS aiming to expand the search progress for users through providing an accurate exploration related to their queries [29]. DBpedia dataset is used in the application so that semantically related data are delivered. Users select the concept of interest from an autocomplete dropdown list of resources using DBpedia Lookup Service. Then, the system proposes a collection of concepts that are related to the user query forming a cloud of tags. Improving the results are possible through the system, by adding the proposed tags to the query bar so a combination of the two concepts is created and is sent as a new query to the system. Relevant information to each of the individual concepts and the combined concepts are returned and presented at separate tabs. As soon as a new tag is added to the query, a new search is performed and new tabs are added to the tab bar. All of the concepts in the tag cloud are associated with each other semantically.

LED is also presenting results of the user query from external search and microblogging systems, which can help users get more general results of their queries, or where the results cannot be found in the dataset so a broader search is given. DBpediaRanker is utilized by the system for ranking resources for the user query. A RESTful JSON API is provided by LED so that any system with HTTP requests can access the LED. This is useful for building web applications featuring new methods and it is suitable for various algorithm comparisons.

D. Aemoo

Aemoo provides exploratory search over the Web [30]. It exploits Encyclopedic Knowledge Pattern (EKP) to deliver its ESS. The system uses DBpedia and external sources to resolve user queries. The combined information is derived from Wikipedia, Twitter and Google News sources. Information combination is attained based on cognitively-sound principles by using knowledge patterns, hypertext link structure and SW technology.

Result presentation is based on EKP[24] filtration so that only relevant results from the returned information are presented. Additionally, the reason for presenting only those results is provided. The system proposes a further utility called *curiosity* so as to present additional knowledge which has been filtered by EKP.

E. Seevl

Seevl is an application for exploring musical data based on SW technology [31]. The application mines music connections to collect desired information with the aim of making context, search and discovery possible to be brought to users like music. A linked dataset of musical entities is built from collecting several resources on the Web. These entities contain musical data for instance Bands, Artists, and etc. so that services, such as recommendations, on top of the dataset are given.

---

[23] http://dbpedia.org/projects/dbpedia-lookup

[24] www.ontologydesignpatterns.org/ekp

A Virtuoso powered RDF store is created for the data being collected, and then it is hosted on an Elastic Compute Cloud (EC2) so as to get the EC2 architecture advantages like elastic cache and load balancing. Moreover, an LDB is built for this data so that it can be browsed by the user and also provides recommendations for the available artists.

F. Discovery Hub

Discovery Hub is an ESS based on the SW technology [32]. The system uses DBpedia data to fetch data from. DBpedia lookup service is used to permit users to select their topic of interest. While selecting the topic a stack is created so that new topics can be selected in case of finding relationships among them. The results of the query are ranked and categorized based on their similarity. Entity labels are used to rank and present retrievals, and explanations for how the results are presented are provided by the system in a graph form. The engine queries the DBpedia SPARQL endpoint on the fly without the need of preprocessing necessities.

Semantic spreading activation is extended on typed graphs of the LD formality and integrated with a graph sampling technique so the results are computed and returned to the frontend.

G. Linked Jazz

Linked Jazz is an ESS to reveal the relationships among jazz musicians based on Linked Jazz dataset [33]. It exploits the technology of LOD to improve the exploration of cultural heritage materials and to enhance the semantics describing them and to reveal the relationships among musicians and expose their society networks from the resources based on transcripts of interviews from jazz archives. Therefore, an RDF store describing these relationships is created as LOD.

The fundamental mechanism for the LOD to be built is that a unique ID (URI) has to be assigned to each entity in the directory. Thus, a directory of jazz musician's names combined with their URIs is constructed. The directory is called Linked Jazz Name Directory. Mapping tool, the dataset foundation application, ingested by extracted files from DBpedia and bibliographic name authority creating the Name Directory. Later on, this directory is refined by Curation tool. The process of Mapping and Curation tools is achieved by automated processing and crowdsourced activities.

The created dataset is then explored via Linked Jazz Network Visualization tool. The tool offers several visualization models such as Dynamic, Fixed, Free and Similar models so that users can use the appropriate model for their exploration of the jazz musician's network.

H. inWalk

inWalk is a Web application to explore LD based on inCloud and thematic walk ideas [34]. The concept of inClould refers to a high-level thematic graph where the vertices of the graph are clusters of relevant LD, and edges are associations of proximity amongst graph nodes. Hierarchical clustering algorithm HC^{f+} is used to construct inClould [35]. The application is built with targets to 1) defeat inflexible LDBs through a presentation of a thematic and a high-level data views generated from similarity based combination techniques; 2) provide common querying methods for inexperienced users in RDF query languages.

The system features can be summarized as 1) Abstraction by aggregation: by providing a conceptual view of data via inCloud notion; 2) exploration by walks: by providing the LD exploration through thematic and inside walks; 3) filtering by patterns: by enabling users to only explore a portion of inClould that satisfies them through filtering actions over the inCloud structure.

There are two main components of the system; the engine, which transforms retrievals from linked datasets, like Freebase and DBpedia, to a similar inCloud cluster; and the front-end, which is an HTML5 based interactive interface, to deliver discovery walks on inCloud clusters.

IV. DISCUSSION

TABLE I presents a summary to LDBs reviewed in the previous section. From there, it is observable that most of the browsers have the ability to connect to more than a specific dataset and present their data. DBpedia has been used in all of them as a default dataset except for CubeViz where it depends on the datasets having statistical contents. The systems present the data in different visualization techniques. LodLive, LODmilla, and DBpedia Mobile Explorer present the retrieved data in graph forms. Infobox-like navigation has been used in LodLive, LODmilla, and DBpedia Atlas in addition to their primary presentation of data so as to explore detailed information about the selected node. CubeViz presents data in statistical forms such as different charts and scatterplots. Faceted navigation is offered by LodLive, CubeViz, and LODmilla so as to let users easily filter the retrieved data and discover their interest. All but CubeViz and LODVisualization use lookup as for query paradigm, while the remaining use dataset existing dimension elements and manual selection of entities respectively. Query suggestion feature is only proposed by LodLive browser, this feature gives users a new opportunity to figure out how the retrievals come from the datasets through SPARQL queries.

Four out of eight systems propose extra utilities so they can be used as LD editors, this can be seen in LodViz, CubeViz, LODmilla and LD Viewer.

The specifications of this feature vary from a system to another. However, this is a great feature for users to be available for both lay users and SW professionals. Lay users can benefit from this feature by obtaining a better understanding and learning for how LD are constructed, and how to add/edit new/existing graph properties or nodes, and for professionals to construct a new knowledge or make advanced practices in the dataset.

TABLE I.   LDBS SUMMERY

System Name	LodLive	CubeViz	LODVisualiz ation	LODmilla	SView	LD Viewer	DBpedia Mobile Explorer	DBpedia Atlas
Release Date	2012	2012	2013	2014	2014	2014	2015	2015
Web Address	http://en.lo dlive.it	http://cubeviz.aksw.o rg	http://lodvisual ization.appspo t.com	http://munkapad. sztaki.hu/lodmill a	http://ws.nju.edu. cn/sview	http://ldv.dbpedi a.org	Mobile app	http://wafi.iit.cnr. it/lod/dbpedia/atl as
Dataset(s)	Multiple	Any dataset contains statistics	Multiple	Multiple	DBpedia	Multiple	DBpedia	DBpedia
Visualization Technique	Graph & infobox	Charts & scatter plots	Treemap, tree, charts	Graph & infobox	Tables via Lenses	Property table	Graph and text	Map-like visualization & infobox
Faceted Navigation	Yes, in graph	Yes	No	Yes	No	No	No	No
Query Model	Lookup	Dimension element selection	Manual selection	Lookup	Lookup	Lookup	Lookup	Lookup
Query suggestion	Yes	No	No	No	No	No	No	No
LD editing utilities	Yes, Ontology verification	Yes, add/edit/delete properties to data, create/edit/import knowledge bases	No	Yes, add/edit/delete nodes/properties	No	Yes, Annotate using DBpedia Spotlight	No	No
LD View Save	No	No	No	Yes	No	No	No	No
Breadcrumb	No	Yes, sessions	Yes, sessions	Yes	No	No	No	No
Live Endpoints	Yes, Multiple	Yes	Yes	Yes	Yes	Yes	Yes	Yes
API	No	No	No	No	No	No	No	No

ESSs are more advanced systems compared to LDBs due to their advanced features. TABLE II presents a summary to ESSs studied in the previous section with additional features summarized from [12].

From the table, it can be noticed that the DBpedia dataset is used as a data resource by all but inWalk. In the case of Aemoo, SWOC, and LED, external sources are used to compute relations among dataset nodes. Search engines, microblogging systems, and social tagging methods are examples of external source. As for query model, the DBpedia lookup service provided by DBpedia SPARQL endpoint is utilized by the majority of ESSs.

The breadcrumb facility, also known as browsing history, is available in all systems through using browser memory sessions. This facility clearly decreases the server requests and supports users the ability to analyze and compare results with previous states. Result explanation feature support searchers to figure out how the results are presented, this utility is offered by Linked Jazz, Seevl, Aemoo and Discovery Hub in different visualizations such as graph and textual forms.

Results are usually ranked according to the similarity measures. inWalk, Linked Jazz, and Aemoo present their results as graphs so the rankings are not available, while the remaining systems present them in different ways. In SWOC, the node size of the graph shows the similarity to the queried concept, the bigger node, the most similar is. The top thumbnail sketch in a list is the most relevant result to the query; this can be seen in rankings of Seevl and Yovisto. LED uses the tag font size as the relevance to the query, and Discovery Hub categorizes the results based on the labels of resources in a list of thumbnail sketches.

TABLE II.   OVERVIEW OF ESSS

System Name	Yovisto	SWOC	LED	Aemoo	Seevl	Discovery Hub	Linked Jazz	inWalk
Release Date	2009	2010	2010	2012	2012	2013	2013	2014
Web Address	www.yovisto.com	sisinflab.poliba.it/semantic-wonder-cloud	sisinflab.poliba.it/led	wit.istc.cnr.it/aemoo	play.seevl.fm	discoveryhub.co	linkedjazz.org	islab.di.unimi.it/inwalk
Main Data	DBpedia EN+DE	DBpedia EN	DBpedia EN	DBpedia EN	DBpedia, Freebase & MusicBrainz	DBpedia EN, FR, IT	DBpedia -> Linked Jazz DB	Freebase & Twitter
Auxiliary Data	No	Search Engines & Tagging Systems	Search Engines & Tagging Systems	External services	No	No	No	No
Query Model	Keyword search	DBpedia lookup	Keyword search	Lookup	Lookup	DBpedia Lookup	Manual selection from a list	Lookup & selection
Matching	String-match	Direct match (lookup)	String matching	Direct match (lookup)	Direct match (lookup)	Direct match (lookup)	Selection	Direct match (lookup)
Domain in (Purpose)	Academic Videos	IT Domain	ICT	General	Music	General	Jazz Musicians	Athletes or Twitter News
Database Method	Freebase Parallax	SWOC Storage	LED Storage	Knowledge Pattern (KP) Repository Manager	Scalable RDF/NoSQL storage by OpenLink Virtuoso	Virtuoso, MySQL	Linked Jazz Name Directory	inWalk repository
Database Purpose	map queries to entities	Stores popularity & similarity values for pairs of resources	Stores results computed by Ranker	responsible for the storage, indexing & fetching of KPs	Scalability	User account & follows	Stores individuals represented by literal triples	Provides a high-level view of relevant LD
Principal Layout	Query suggestions	Graph	Tags cloud	Graph	List	List	Graph	Graph
Results Explanations	No	No	No	Wikipedia-based	Shared properties	Yes, Text & Graph	Yes, Interview Transcripts	No
Breadcrumb	Sessions & registration	Sessions	Sessions	Sessions	Sessions & registration	Sessions & registration	Sessions	Sessions
Algorithms	Set of heuristics	DBpedia Ranker	DBpedia Ranker	EKP filtered view	LDSD, DBrec algorithm	Semantic spreading activation	Mapping & Curator Tool & the Transcript Analyzer	HCf^{I+} clustering algorithm
Ranking	Yes	Yes, graph size	Yes	No	Yes	Yes	No	No
Offline Processing	Yes	Yes, similarity of pairs	Yes	Yes , EKP part	Yes	No, on the fly	Yes, Linked Jazz DB	Yes
API	RDF triple-store	No	Yes, RESTful	Yes, RESTful	Yes, Content negotiation JSON-LD	No	Yes, JSON, RDF & GEXF files	No
Faceted Navigation	Yes	No	No	No	Yes	Yes	No	No
User Interface	HTML	Flash based	HTML	HTML	HTML, ajax	HTML	HTML5 + jQuery	HTML5 + JS

Different algorithms are used in the ESSs to calculate the similarity values. SWOC and LED use DBpediaRanker to calculate the similarities between DBpedia resources. A collection of heuristics to decide the best similar entity in the dataset is utilized in Yovisto. EKP has been used in Aemoo to determine the related type of resources and to state the representative classes to clarify entities of a particular type. Seevl employs LDSD and DBrec algorithms to bring musical recommendations. A semantic-sensitive traversal algorithm combined with a graph sampling technique is the base algorithm of Discovery Hub. In Linked Jazz, Mapping and Curator tools were used to construct its dataset and refine its

data respectively. The inWalk employs HC^{I+} clustering algorithm for the construction of inCloud.

Faceted browsing feature, which allows users to filter the results, is proposed by Seevl, Yovisto and Discovery Hub.

Database technology has been used in most of the systems with different aims. Freebase Parallax is utilized in Yovisto so as to get a collection of entities first, and retrieve videos related to entities. Both systems SWOC and LED benefit from the database technology by using a Storage to keep the similarity values among pairs of nodes in a DBMS so the retrievals of the system are more efficient at runtime. Knowledge Pattern Repository Manager is utilized by Aemoo

to store, index and fetch retrievals from Knowledge Patterns. OpenLink Virtuoso platform for RDF/NoSQL data is used by Seevl for the aim of scalability.

Regarding the application program interface (API) of the systems, five of the systems, Yovisto, LED, Aemoo, Seevl and Linked Jazz, provide facilities for mashup Web applications. Yovisto published its metadata in RDF format, embedded in web pages as RDFa and reachable via an RDF triple store. LED tag cloud generation is publically available as RESTful web service. The server-side section of Aemoo is released based on a REST Web service in Java, while its client side interface interacts with third party sections by REST interfaces via AJAX. For the sake of building applications on top of Seevl, Content Negotiation[25] on Seevl server is enabled and provides all the data as JSON-LD. The outputs of the Linked Jazz API are JSON, RDF Triples, and Gephi GEXF files. Although the default return is JSON, some of the ESSs allow the data return in other formats.

Fig. 1 describes statistics of the features used by all of the systems reviewed in this paper. As it can be noticed, DBpedia has been used as the dataset by 88% of the systems, which indicates the importance of this dataset and its wide usage. DBpedia Lookup Service has been used as query model by 63% of the systems, however, all of the systems that use DBpedia as their main dataset, use DBpedia Lookup Service as their query model. The presentation of the faceted navigation, which helps the users to realize and comprehend the information space, is still poor, this is due to the heavy processing of large-sized datasets.
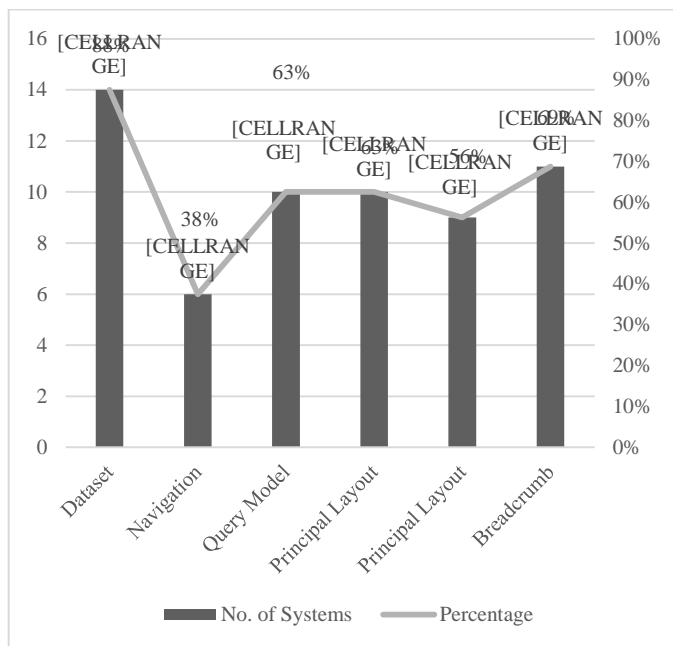

Fig. 1. Common features used by the systems

V. CONCLUSION

The effortless process of sharing data online created a massive volume of data which is increasing continuously.

Consequently, the information retrieval systems are facing new challenges in order to return relevant data. The growth of the syntactic Web to the SW technology where the information is understandable by machines has raised expectations. This technology improves the efficiency of searching systems through automatic processing of data.

This paper has given an overview of the SW technology and LD. The concept of LDBs and ESSs are clarified. Afterward, the most recent LDBs and ESSs for LD have been reviewed in details. The motivation behind this study was to provide the reader with a clear explanation of the LDBs and search systems such as the way they use LD, their query paradigm, principal layout and used algorithms.

The future prospects of the exploration systems based on LD are promising. Moreover, the interest in this area will continue to expand constituting a critical enhancement for the future of the search experience and its results.

REFERENCES

[1] G. Madhu, D. A. Govardhan, and D. T. Rajinikanth, "Intelligent Semantic Web Search Engines: A Brief Survey," ArXiv Prepr. ArXiv11020831, 2011.

[2] G. Marchionini, "Exploratory Search: From Finding to Understanding," Commun ACM, vol. 49, no. 4, pp. 41–46, Apr. 2006.

[3] R. Mirizzi, A. Ragone, T. D. Noia, and E. D. Sciascio, "Semantic Wonder Cloud: Exploratory Search in DBpedia," in Current Trends in Web Engineering, F. Daniel and F. M. Facca, Eds. Springer Berlin Heidelberg, 2010, pp. 138–149.

[4] T. Jiang, "Exploratory Search: A Critical Analysis of the Theoretical Foundations, System Features, and Research Trends," in Library and Information Sciences, Springer, 2014, pp. 79–103.

[5] J. A. R and M. Kurian, "A Survey on Tools essential for Semantic web Research," Int. J. Comput. Appl., vol. 62, no. 9, pp. 26–29, Jan. 2013.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," Sci. Am., vol. 284, no. 5, pp. 28–37, 2001.

[7] D. Fensel, Spinning the Semantic Web: bringing the World Wide Web to its full potential. MIT Press, 2005.

[8] K. Krieger and D. Rosner, "Linked Data in E-Learning: A Survey," Semantic Web 0, pp. 1–9, 2011.

[9] T. Heath and C. Bizer, Linked Data: Evolving the Web into a Global Data Space, 1st edition. Morgan & Claypool., 2011.

[10] Le Hors, M. Nally, and S. Speicher, "Using read/write Linked Data for Application Integration-Towards a Linked Data Basic Profile.," presented at the LDOW, 2012.

[11] Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," Int. J. Semantic Web Inf. Syst., vol. 5, no. 3, pp. 1–22, 2009.

[12] N. Marie and F. Gandon, "Survey of linked data based exploration systems," presented at the IESD 2014-Intelligent Exploitation of Semantic Data, 2014.

[13] GEROIMENKO AND C. CHEN, VISUALIZING THE SEMANTIC WEB: XML-BASED INTERNET AND INFORMATION VISUALIZATION. SPRINGER SCIENCE & BUSINESS MEDIA, 2006.

[14] D. V. Camarda, S. Mazzini, and A. Antonuccio, "Lodlive, exploring the web of data," presented at the Proceedings of the 8th International Conference on Semantic Systems, 2012, pp. 197–200.

[15] P. E. R. Salas, M. Martin, F. M. D. Mota, S. Auer, K. Breitman, and M. Casanova, "Publishing statistical data on the web," presented at the Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on, 2012, pp. 285–292.

[16] J. M. Brunetti, S. Auer, R. García, J. Klímek, and M. Nečaský, "Formal linked data visualization model," presented at the Proceedings of International Conference on Information Integration and Web-based Applications & Services, 2013, p. 309.

---

[25] http://en.wikipedia.org/wiki/Content_negotiation

[17] E. H. Chi, "A taxonomy of visualization techniques using the data state reference model," presented at the Information Visualization, 2000. InfoVis 2000. IEEE Symposium on, 2000, pp. 69–75.

[18] Micsik, Z. Tóth, and S. Turbucz, "LODmilla: Shared Visualization of Linked Open Data," presented at the Theory and Practice of Digital Libraries--TPDL 2013 Selected Workshops, 2014, pp. 89–100.

[19] Y. Qu et al., "SView: Smart Views for Browsing Linked Entities," Semantic Web Chall., 2014.

[20] D. Lukovnikov, C. Stadler, and J. Lehmann, "LD viewer-linked data presentation framework," presented at the Proceedings of the 10th International Conference on Semantic Systems, 2014, pp. 124–131.

[21] Vagliano, M. Marengo, and M. Morisio, "DBpedia Mobile Explorer," presented at the Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015 IEEE 1st International Forum on, 2015, pp. 181–185.

[22] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, and A. Saulnier, "GosperMap: Using a gosper curve for laying out hierarchical data," Vis. Comput. Graph. IEEE Trans. On, vol. 19, no. 11, pp. 1820–1832, 2013.

[23] F. Valsecchi, M. Abrate, C. Bacciu, M. Tesconi, and A. Marchetti, "DBpedia Atlas: Mapping the Uncharted Lands of Linked Data," Proc. Workshop Linked Data Web, 2015.

[24] R. W. White and R. A. Roth, Exploratory Search: Beyond the Query-Response Paradigm Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2009.

[25] R. W. White, B. Kules, and B. Bederson, "Exploratory search interfaces: categorization, clustering and beyond: report on the XSI 2005 workshop at the Human-Computer Interaction Laboratory, University of Maryland," presented at the ACM SIGIR Forum, 2005, vol. 39, pp. 52–56.

[26] V. Dimitrova, L. Lau, D. Thakker, F. Yang-Turner, and D. Despotakis, "Exploring Exploratory Search: A User Study with Linked Semantic Data," in Proceedings of the 2Nd International Workshop on Intelligent Exploration of Semantic Data, New York, NY, USA, 2013, p. 2:1–2:8.

[27] K. Jacksi, N. Dimililer, and S. R. Zeebaree, "A SURVEY OF EXPLORATORY SEARCH SYSTEMS BASED ON LOD RESOURCES," Proc. 5th Int. Conf. Comput. Inform. ICOCI 2015, pp. 501–509, 2015.

[28] J. Waitelonis and H. Sack, "Towards Exploratory Video Search Using Linked Data," Multimed. Tools Appl, vol. 59, no. 2, pp. 645–672, Jul. 2012.

[29] R. Mirizzia, A. R. T. Di Noiaa, and E. Di Sciascioa, "Lookup, Explore, Discover: how DBpedia can improve your Web search," 2010.

[30] Musetti et al., "Aemoo: Exploratory search based on knowledge patterns over the semantic web," Semantic Web Chall., 2012.

[31] Passant, "seevl: mining music connections to bring context, search and discovery to the music you like," in Semantic Web Challenge, 2012.

[32] N. Marie, F. Gandon, M. Ribière, and F. Rodio, "Discovery Hub: On-the-fly Linked Data Exploratory Search," in Proceedings of the 9th International Conference on Semantic Systems, New York, NY, USA, 2013, pp. 17–24.

[33] M. C. Pattuelli, M. Miller, L. Lange, S. Fitzell, and C. Li-Madeo, "Crafting Linked Open Data for Cultural Heritage: Mapping and Curation Tools for the Linked Jazz Project," Code{4}lib, no. 21, Jul. 2013.

[34] S. Castano, A. Ferrara, and S. Montanelli, "inWalk: Interactive and Thematic Walks inside the Web of Data.," presented at the EDBT, 2014, pp. 628–631.

[35] A. Ferrara, L. Genta, and S. Montanelli, "Linked Data Classification: A Feature-based Approach," in Proceedings of the Joint EDBT/ICDT 2013 Workshops, New York, NY, USA, 2013, pp. 75–82.