This is a postprint version of the following published document:

Sánchez-Cervantes, J.L., Colombo-Mendoza, L.O.,
Alor-Hernández, G. et al. LINDASearch: a faceted
search system for linked open datasets. *Wireless Netw*
26, 5645–5663 (2020).

DOI: <https://doi.org/10.1007/s11276-019-02029-z>

© 2019, Springer Science Business Media, LLC, part of Springer
Nature

José Luis Sánchez-Cervantes¹, Luis Omar Colombo-Mendoza², Giner Alor-Hernández^{2*}, Jorge Luis García-Alcaráz³, José María Álvarez Rodríguez⁴, Alejandro Rodríguez-González⁵

¹CONACYT-Instituto Tecnológico de Orizaba, Av. Oriente 9 852, Col. Emiliano Zapata, C.P. 94320, Orizaba, Mexico.

²Tecnológico Nacional de México/ I. T. Orizaba, Av. Oriente 9 852, Col. Emiliano Zapata, C.P. 94320, Orizaba, Mexico.

³Universidad Autónoma de Ciudad Juárez, Av. Plutarco Elias Calles 1210, Col. Foviste Chamizal, C.P. 31310, Ciudad Juárez, Mexico.

⁴Computer Science Department, Universidad Carlos III de Madrid, Av. Universidad 30, Leganés, 28911 Madrid, Spain.

⁵ETS de Ingenieros Informáticos, Universidad Politécnica de Madrid Campus de Montegancedo, Boadilla del Monte, 28660, Madrid, Spain.

E-mail addresses: jlsanchez@conacyt.mx (JL Sánchez), lcolombom@ito-depi.edu.mx (LO, Colombo), jorge.garcia@uacj.mx (JL García), joalvare@inf.uc3m.es (JM. Álvarez), alejandro.rg@upm.es (A. Rodríguez)

*Corresponding Author: G. Alor-Hernández

e-mail: galor@ito-depi.edu.mx

Tel: +52 (272) 72 57056

Abstract The importance of Linked Data lies on the fact that its practices and principles have been adopted by an increasing number of data providers, resulting in the creation of a data space on the Web containing billions of RDF Triples and accessible worldwide throughout the Internet. RDF datasets can be queried by tools and applications for searching and gathering information. However and due to the huge amount and types of dataset, the selection and reuse of data resources is not easy task. Therefore, a metasearch system for open Linked Data projects called LINDASearch (LINDASearch stands for Linked Data Search) is introduced. LINDASearch provides a middleware architecture in order to provide infor-

mation about the most known Open Linked Data Projects such as DBpedia, The GeoNames geographical database, LinkedGeoData, FOAF profiles, Global Health Observatory, Linked Movie DataBase (LinkedMDB) and World Bank Linked Data. This paper describes the LINDASearch's architecture as well as its functionality through one case study divided in two scenarios in order to show the architecture's functionality and present the results obtained from each scenario.

Keywords: Faceted Browser, Linked Open Data, Links Discovery, Unification datasets

1. Introduction

The transformation of traditional Web into the Web of Data consists in shifting the paradigm from a web of documents to a web of data [1]. The Semantic Web is not a kind of separate Web but an extension in which information is adequately defined and published, enabling computers and people to collaborate sharing and exchanging data [2], [3].

As a part of the Semantic Web, the adoption of the Linked Data principles [4], has led to the creation of a global data space, connecting data from diverse domains such as personal information management, medicine, social networks, bibliography or scientific data. The web of data enables the development of new types of applications and opens new possibilities for domain-specific applications that requires a complete view of the knowledge graph created by the linked resources [5].

However, one of the well-known issues in the consumption of Linked Data is that each dataset has specific characteristics (classes, properties and relationships). This situation implies a challenge [6] when intra and Cross-domain queries must be performed to exploit the underlying knowledge graph. Other challenges have been also identified as barriers to enable a proper reuse of data [7]:

- 1) Data modelling: data sources use different RDF vocabularies;
- 2) Identification: same resources are published under different URIs within different data sources;
- 3) Data inconsistency: data from different data sources may contain conflicting values; and
- 4) Cataloguing: data providers use different techniques to catalogue their data into categories.

Furthermore, it is necessary to consider aspects regarding the usability and semantics of the datasets [8]:

- a) Expressivity: it is the ability to query datasets by referencing elements in the data model structure and operate over the data;
- b) Usability: it allows for an easy-to-operate, intuitive, and task-efficient query interface;
- c) Vocabulary-level semantic matching: it is the ability to semantically match user query terms to dataset vocabulary-level terms;

- d) Entity reconciliation: it matches entities expressed in the query to semantically equivalent dataset entities; and
- e) Semantic tractability mechanisms to improve on the ability to answer queries not supported by explicit dataset statements.

To tackle these challenges, the present paper introduces a Web-based system called LINDASearch that eases users the consumption of Linked Data improving the search capabilities of a data management platform. LINDASearch is designed as a middleware architecture to gather and store information about the most known Open Linked Data Projects, such as DBpedia, The GeoNames geographical database, LinkedGeoData, FOAF profiles, Global Health Observatory, Linked Movie DataBase (LinkedMDB), and World Bank Linked Data, serving as a gateway to look up data resources.

This paper is structured as follows: Section 2 presents recent advances in the state of the art of semantic search systems, while Section 3 presents LINDASearch's internals and functionality. On the other hand, Section 4 describes the unification process for multiple Linked Datasets; Section 5 describes the LINDASearch's evaluation. Finally, some conclusions and future directions are outlined in Section 7.

2. State of the Art

This section reviews the main initiatives to browse and ease the access to linked data resources. To do so, works have been classified into two main groups: 1) Data unification under the Linked Data paradigm and 2) Navigation on Linked Data.

The possibility of building indexing data structures for semantic search, which consists of the Keyword, Entity and Type Indices to support the process of retrieving both keyword-based and semantic information for annotated queries, was researched by [9]. In order to find an efficient indexing data structure, authors have explored both homogeneous and heterogeneous index types that use similar as well as dissimilar data structures for each of the index types, respectively. In addition, they integrated the results of the various index types using two approaches: 1) List-based and, 2) Non-list-based approaches. Authors have found that the list-based approach is more efficient compared to the non-list-based approach for heterogeneous indices. In [10] a solution based on 4 phases covering an important number of heterogeneities from RDF datasets, in order to reduce the user configuration effort to integrate this information type was proposed. Each phase consists of: 1) Property filtering, or automatic data cleaning of "problematic" attributes; 2) Instance profiling allowing to represent each resource by a sub-graph considered relevant for the comparison task; 3) Instance vector representation allowing to compare resources, and 4) A postprocessing data based on hierarchical clustering and key ranking techniques to disambiguate highly similar, though not identical instances. The Open PHACTS consortium developed the Open PHACTS

Discovery Platform that leveraged Linked Data, to provide an integrated access to pharmacology databases in chemistry or biology [11]. Similarly, a tool that produces structured interoperable data from product features, i.e., attribute name–value pairs on the Web was presented by [12]. The tool extracts the product features using a Web site-specific template created by users. Data is then enriched using the GoodRelations vocabulary producing an RDFa or microdata view of the product. Such snippets can be embedded into existing static and dynamic Web pages to allow search engines like Google, Bing and Yahoo to easily access and index data products [13]. In the financial context, the FLORA [14] platform offers an integrated linked data view of XBRL documents to enable extended financial data analysis. A web-based application for collecting and exploring the Linked Open Data cloud was presented by [15]. On the other hand, [16] introduced the SPARQL2XQuery Framework to create an interoperable environment, where SPARQL queries are automatically translated into XQuery queries to access XML data across the Web. The framework supports both manual and automatic mapping specification between ontologies and XML Schemas. The SWGET portal, presented by [17], is a Web-based portal that allows users to train software modules to (virtually) move across data sources, interpret knowledge, take decisions, and trigger actions. In this sense, the SWGET portal facilitates the data exploration in DBpedia through SPARQL-based queries execution. Such instructions are given via the navigational language NautiLOD [18], since it facilitates the definition of navigational expressions. Similarly, SWOWS (Semantic Web Open dataflow System), a platform for declarative specification of applications using Linked Data, was presented by [19]. The main contribution of this platform is the possibility of automatically generating SPARQL queries. [8] focused on the research of a generic type of query mechanism for Linked Data, the provision of an independent vocabulary and a natural query language for Linked Data. According to the authors, this type of queries allows users to expressively gather the contents of distributed linked datasets without the need of a prior knowledge of the underlying vocabularies. PatEx, a system designed to explore RDF(S)/OWL datasets using two exploration strategies, allowing the user to interactively switch between them was introduced in [20]. These two approaches rely on the definition of patterns to formalize users' requirements during the exploration process. The carried out experiments on RDF(S)/OWL real datasets, demonstrated the effectiveness of approach proposed by authors. The Linked Open Graph (LOG), presented by [21], is a Web-based application for collaborative browsing and navigating several SPARQL endpoints. The main contributions of this approach were: 1) management of multiple SPARQL endpoints; 2) storing and sharing RDF graphs via the web; and 3) learning and inspection of RDF graphs. [22] proposed an automatic faceted navigation mechanism that allows semantic units from texts, build a hierarchy of semantic facets, and navigate the hierarchy. Such faceted navigation mechanism incorporates

the simulation of the human reading process and it allows users to read multiple facets of one text or more texts without any prior input. Previously, [23] had proposed a faceted navigation approach through keyword interaction. However, their most recent method does not need any prior input, and it transforms one or more texts into a hierarchy of semantic facets for navigation. It also provides a new reading model for users in order to read semantically related content. The new mechanism and its previous mechanism can be integrated to form a powerful faceted navigation mechanism. Similarly, the initiative of [24] focused on a dynamic categorization called faceted navigation in literature. The topic of the paper was particularly facet selection/ordering, using the total gain ratio, applied to a movie search engine working with an ontology. The study of [25] focused on implementing a dynamic faceted navigation system as a decision support system, which grouped results based on search context using the Semantic Web technology. Discovery Hub, a novel application that processes DBpedia for an exploratory search purpose was presented by [26]. It implements on-the-fly semantic spreading activation and sampling over Linked Data sources to suggest ranked topics of interest to the user. Discovery Hub uses DBpedia to perform the semantic spreading activation and organize the results space (e.g. filters and facets). It also offers various explanations about the results based on DBpedia and Wikipedia. [27] proposed an exploratory strategy for patent search that exploit metadata already available in patents. Authors proposed the exploitation of Linked Data for specifying the entities of interest and providing further information about the identified entities. From an information-seeking process perspective, they also presented the tight integration of different search tools for a) Faceted Search using existing metadata; b) Entity extraction; and c) Textual clustering. [28] carried out a user study with a unifocal semantic browsing interface for exploratory search through several datasets linked via domain ontologies. The study was qualitative and exploratory in nature and used music as an illustrative domain. It examined: a) Obstacles and challenges related to user exploratory search in LOD cloud and b) The effect of serendipitous learning and the role that Semantic Web plays in this type of learning. [29] presented Facete, a Web-based exploration and visualization application enabling the faceted browsing of data with a spatial dimension. Facete implements a novel spatial data exploration paradigm based on three key components: 1) A domain independent faceted filtering module, which directly operates on SPARQL and supports nested facets; 2) An algorithm that efficiently detects spatial information related to those resources that satisfy the facet selection; and 3) A workflow for making the map display interact with data sources that contain large amounts of geometric information. In a previous work [30], [31], proposed the Query-based Faceted Search (QFS), as a way to reconcile the expressivity of formal languages and the usability of faceted searches. In [32], presented a further reconciled QFS with scalability and portability by building queries over SPARQL endpoints. The expressivity

and readability were improved. With Ferré's initiative, many SPARQL features are now covered. A faceted search platform that providing the following main functionalities: 1) Computation of facet graphs from an ontology; 2) Interface generation from facet graphs; and 3) Interface update in response to user actions, was developed by [33]. Authors carried out the implementation of their platform through a proof-of-concept system called SemFacet. A customizable framework named LD Viewer was presented by [34]. LD Viewer can easily fit in different Linked datasets. Also, LD Viewer further improves upon the Triple Action Framework (TAF) from DBpedia Viewer and it allows for both easy high-level interface customization and adding interactivity to the presented triples. [35] presented an initiative that consisted in to research easy-to-use interfaces to access Linked Data. The author looked into ways of turning search keywords into URIs and applied the technique in a simplified end-user interface for accessing Linked Data [36]. This approach was extended into CAF-SIAL, a proof-of-concept application described by [37]. CAF-SIAL helped users search information about concepts in the LOD cloud without having to know any of the mechanics of the Semantic Web. The second phase of CAF-SIAL was closely related to the CODE project [38]. The goal of CODE is to establish a sophisticated ecosystem for Linked Data [39]. Under this perspective, the focus of Hoefler's research was the Linked Data Query Wizard as a part of CODE's Visual Analytics work package.

Faceted Search is a technique to access document collections that combines text search and faceted navigation applied to the documents' metadata. With the latter, users can narrow down search results by incrementally applying multiple filters called facets [40]. It can be stated that Faceted Search is a prominent approach for end-user data access, and several RDF-based Faceted Search systems have been developed. In this context, GraFa: A faceted browsing system for heterogeneous large-scale RDF graphs based on a materialization strategy that performs an offline analysis of the input graph in order to identify a subset of the exponential number of possible facet combinations that are candidates for indexing was proposed by [41]. In experiments over Wikidata, authors demonstrated that materialization allows for displaying (exact) faceted views over millions of diverse results in under a second while keeping index sizes relatively small. In [42] a solution to effectively address the challenges of real-time analysis of data from Social media using a configurable search engine based on Elasticsearch and Kibana was demonstrated. Authors used a distributed database architecture, they pre-build indexing and standardizing the Elasticsearch framework for large scale text mining. The obtained results from the query engine are visualized in almost real-time. Finally, a detailed review of previous studies on using data mining techniques in indoor navigation systems for the Internet of Things (IoT) scenarios was presented by [43]. The aim of authors initiative consists in understand what type of navigation problems exist in different IoT scenarios

with a focus on indoor environments and later they will investigate how data mining solutions can provide solutions on those challenges.

Although there are multiple related researches, these initiatives pose several limitations: a) most of them are based on only one specific domain and one specific dataset; b) a multi-domain semantic search system has not been yet reported; c) several of these works do not include the faceted navigation among their features and are also limited to searching through SPARQL-based queries, which are complex for non-experienced users; d) the analyzed works not include the discovering of additional information over DBpedia for complement the obtained results, and e) no related work include the functionality for recommending the search queries most frequently performed. However, the main contribution of LINDASearch regarding the state of the art, consists in provides an alternative to solve the afore mentioned deficiencies which can be improved by: a) defining a metasearch system that implements queries for multi-domain Linked Data datasets; b) designing a system that allows users to avoid the search for information via SPARQL endpoints and write SPARQL queries; c) implementing a metasearch system to facilitate and expedite the search of user information through the faceted navigation; d) in a holistic sense, help to complement the obtained results through the discovering of additional information that are related with de user's search; e) generate search recommendations associated to search queries that are performed by active users, and f) generate search recommendations syntactically similar to the stored in the history navigation.

3. LINDASearch: A Meta Search System for Linked Open Datasets

LINDASearch is a multi-domain semantic metasearch developed by using JavaServer Faces-PrimeFaces, and Jena technology. LINDASearch implements five datasets by applying the SPARQL-based queries needed, depending on the users search. These datasets are: DBpedia, Geo WGS84, Global Health Observatory, LinkedMDB, and World Bank Linked Data.

By using LINDASearch, users can perform semantic search to obtain information and restrict the overall set of results through multiple criteria or facets. Thus, the results are more accurate since they are based on the semantic properties or attributes for certain Datasets. When the results of the search contain hyperlinks, these will be displayed to allow users to navigate through the related web pages.

3.1 Architecture and functionality

The architecture of LINDASearch is based on four main layers: (1) one layer that represents the interface between the user and the remainder of the application, (2) one layer that is in charge of receiving and

sending data through the user interface, (3) one layer that interprets the queries that come from the previous layer and extracts the results obtained from the LOD cloud and (4) a set of millions RDF triples with information on several fields that are published and maintained by the respective provider. Each layer contains components that are also divided into several subcomponents and there are different levels of interdependence among them during the framework operation. This architecture has a layered-design that allows for scalability and easy maintenance, since its tasks and responsibilities are distributed.

3.2 Components description

The LINDASearch architecture is presented in Fig. 1, in order to help describe in general way the structural components, the relationships among them and their behavior. Such description is the follows:

Fig. 1. LINDASearch Architecture

- *Dataset Selector*: This component provides users the option of selecting one of the Linked Open Datasets available in LINDASearch through which they can perform semantic searches.
 - *Endpoint Selector*: It obtains and sends the namespace for each dataset to the Data Selector Component.
- *Faceted Query Builder*: This component uses the XML-based document built by the Response Builder component in order to build the facets corresponding to the results obtained after the information extraction process. The facets built in this component are shown in the visual user interface, which is a JSF-PrimeFaces Web Application.
- *Navigation History Builder*: This component is responsible for keeping temporally the navigation history of users to expedite the searches performed by other users, including the links discovered into the LOD cloud by the Links Discoverer component.
- *Recommendation Engine*: This component is responsible for recommending the search queries that are most frequently associated to an active user's search query in the navigation histories of the k most similar users to the active user; it is also responsible for recommending those search queries among all the search queries in the system which are related (i. e., syntactically similar) to the active user's search query. This component has three main modules:
 - *Preference Learner*: This module is responsible for calculating ratings for all the search queries in the system based on the duration of the associated search sessions and the weights (TF-IDF scores) of the search queries to the navigation histories in the whole collection of navigation histories.

- *Association Rule Miner*: This module uses the Apriori algorithm to learn association rules from the search query data in the dataset of the navigation histories owned by the k most similar users to the active user and to recommend those search queries that are most frequently associated to the active user's search query.
 - *Doc2vec Builder*: This module uses the doc2vec model to compute vectors that hold numeric representations of documents for the documents represented by the search queries available in the system and to infer the corresponding vectors for those documents represented by unknown search queries. These document vectors are the starting point for the recommendation of other search queries that are related (i.e., syntactically similar) to the active user's search query.
- *Response Builder*: this component receives the responses coming from SPARQL-based queries. This module retrieves useful information from responses and builds a XML-based document with information from a Linked Open dataset. This XML-based document is presented in HTML-based format to display the results to the user.
- *Semantic Layer Bridge (SLB)*: The aim of *Semantic Layer Bridge* subcomponent is to interpret all the queries coming from *Integration Layer* in order to infer whether the knowledge that will be provided to these modules could come from the *Directory* layer or whether some extra information could be provided by some of the components which have been added to this layer. This component has two main modules:
 - *Data Source Selector*: It analyzes the information and selects the appropriate data source.
 - *Data Query Manager*: depending on the source selected in the Data Source Selector module, Data Query Manager will throw a process to retrieve the information stored in the datasets, including their properties (predicates) that will enable to navigate among more datasets through the links discovery on the Web. However, when it is unable to obtain properties from the datasets, Data Query Manager can obtain them from its list of properties, which are organized according to their use in different categories [44]. Then, the Semantic (LD) Querier executes the process to obtain information stored in the datasets.
- *Links Discoverer (Silk Framework)*: Once the search is performed, this component allows the user to discover links of information related to the key word(s). The links discovery is possible

thanks to the Silk Framework, which shows an emergent interface with a list of discovered links.

The process of discovering links is described in detail in following section [45].

- *Linked Open Data Extractor (LODE)*: The aim of LODE component is to analyze the information stored in the Linked Open Data Cloud and if this information is necessary at some moment, then it is extracted. This component has two main modules:
 - *Data Finder/Analyzer*: This module is invoked when the *Data Source Selector* module of SLB component tries to know if the information that it is searching can be found in the Linked Open Datasets. The aim of this module is to navigate in the LOD cloud to search specific information.
 - *Data Extractor*: Once the *Data Source Selector* module has determined that the information should come from Linked Data information source and the *Semantic (LD) Querier* module has been thrown, this module extracts the appropriate information.

4. Unification of Linked Datasets

When Linked Open Data is consumed, the need for matching different vocabularies of the consumed (internal and external) datasets often occurs. This task is relevant to ensure smooth data integration by vocabulary alignment [46]. Considering this premise, LINDASearch tries to solve the matching process of different vocabularies corresponding to datasets queried by the users. In order to do this task, LINDASearch tries to unify different datasets by following the phases of the process that are briefly described below:

- A) During the first phase, while a process of parameterized search or faceted navigation is done in any dataset (dataset A), the *Data Query Manager* component obtains at the same time the properties of that dataset and sends them with the search value (keyword) to the Dataset Selector component. When this occurs, task A1) is executed. Otherwise, if it does not obtain the properties from the dataset, then task A2) is executed.

A1) It launches a crawl to retrieve the properties of that dataset and it sends them with the search value (keyword) to *Dataset Selector* component (e.g. the data interlinking between Dataset A and DBpedia).

A2) If the *Data Query Manager* did not find any results after crawling the properties from the dataset, then it will obtain the required properties from its properties table (see Table 1). The twelve properties contained in Table 1 are organized according to their primary usage in the LOD cloud under the following categories: Social Web, Publications, User-Generated Con-

tent, Media, Life Sciences, Government, Geographic, and Cross-domain [44]. Thus, the *owl:sameAs* property will be used in first place for the links discovery (phase D). If the links are not discovered with this property, then the Data Query Manager tries to obtain results with the next property: *rdfs:seeAlso*, and so on. This operation is repeated until one of the properties allows for the discovery of links in Phase 4. If none of the properties enables to discover links, LINDASearch will notify the user that there are no more facets to navigate.

- B) In the second phase, the Dataset Selector component sends the URIS of the datasets where the links discovery will take place through the Endpoint selector.
- C) In this phase, when the URIs are obtained, the Endpoint Selector component obtains the appropriate SPARQL endpoint to carry out the discovery of links. In this phase, the properties (e.g. *owl:sameAs* and *rdfs:seeAlso*), the keyword, and the SPARQL endpoint (namespaces) are sent as input parameters to the Links Discoverer component.
- D) In next phase, the Links Discoverer component receives the parameters and configures the Silk Framework [45] (XML configuration file) with these parameters to subsequently execute the Silk Framework on the SPARQL endpoint established in the Silk configuration. Fig. 2 exemplifies how the properties are obtained from dataset A and how the *owl:sameAs* property is selected to discover links on DBpedia (A1). Moreover, the same Figure shows the case where properties are not obtained from Dataset A, but from the properties table in order to discover links on the Dataset B (A2). In this case, links were not found with the *owl:sameAs* property on the Dataset B; therefore, the Links Discoverer continues the search with the following property: *rdfs:seeAlso* to discover links on the same dataset: B (A2). In both cases, the discovered links are sent to the Response Builder component.
- E) In the last phase, once the links are discovered, the Response Builder component builds a XML-based document with these links, which are then formatted and shown in HTML-based format to enable the navigation among datasets (e.g. DBpedia and Dataset B) following a style of facets through the visual interface.

It is noteworthy that in order to expedite the navigation among datasets, the Navigation History Builder temporarily keeps each of the links browsed. In addition, for each link discovered and accessed, the Data Query Manager obtains the value of the search (keyword) and tries to retrieve the properties from their corresponding dataset in order to facilitate the search of more information through the discovery of links available on other datasets. In essence, the process of discovering links returns to phase A.

This process seeks to demonstrate that LINDASearch here unifies three different linked datasets, although it actually allows for the unification of more datasets. It is suitable to run the Silk Framework over the SPARQL endpoint of DBpedia since it is the core of the LODcloud [47], [48], and the *owl:sameAs* property is used since it is a property commonly used in the data interlinking of several categories [44].

This process seeks to demonstrate that LINDASearch here unifies three different linked datasets, although it actually allows for the unification of more datasets. It is suitable to run the Silk Framework over the SPARQL endpoint of DBpedia since it is the core of the LODcloud [47], [48], and the *owl:sameAs* property is used since it is a property commonly used in the data interlinking of several categories.

Table 1. Priority of the Properties in LINDASearch [44]

PREDICATE	CATEGORY	USAGE	PREDICATE	CATEGORY	USAGE
owl:sameAs	Social Web	32.20 %	foaf:knows	Social Web*	60.27 %
	Publications	53.13 %	foaf:based_near	Media	18.75 %
	User-Generated	81.25 %		Social Web	35.69%
	Life Sciences	52.17 %		Total	54.44 %
	Government	24.27 %	dct:creator	Life Sciences	21.74 %
	Geographic	80.00 %		Cross-domain	20.00 %
	Total	323.02 %		Total	41.74 %
rdfs:seeAlso	Publications	23.73 %	sioc:follows	Social Web*	34.34 %
	User-Generated	21.88 %	dct:spatial	Government*	30.10 %
	Media	18.75 %	dct:language	Publications*	25.42 %
	Life Sciences	48.48 %	skos:exactMatch	Geographic*	21.43 %
	Cross-domain	52.00 %	skos:closeMatch	Geographic*	21.43 %
	Total	164.84 %	dct:source	User-Generated*	18.75 %
dct:publisher	Life Sciences	47.57 %	* = Total		
	Government	47.57 %			
	Total	95.14 %			

Finally, LINDASearch uses four of its components to try to unify multiple datasets through several of the most used properties in order to discover and navigate in the links of multiple datasets and, in a holistic sense, provide complementary information for the performed searches while keeping the browsing history. Fig. 2 depicts the process for unification of different datasets that was described in this section.

Fig. 2. Unification Datasets Process of LINDASearch

5. Evaluation

A broad range of evaluation methods have been proposed in software development for qualitative, quantitative, and hybrid evaluation methods can evaluate software and tools, each one differently [49]. Quantitative evaluation methods are based on the assumption that a software product has at least one measurable property expected to change as a result of using the methods/tools to be evaluated. On the other hand, feature analysis is usually hold as a qualitative evaluation method. More specifically, in feature analysis, the researcher identifies the user requirements for a particular task or activity and maps such requirements to the features that a method/tool aimed at supporting that task or activity should possess. Finally, hybrid evaluation methods may provide more comprehensive assessments, since they combine both qualitative and quantitative elements. Considering Kitchenham, Linkman, & Law's classification of software evaluation methods, we conducted a hybrid analysis of LINDASearch to identify its main features and quantitatively measure the results obtained by its Links Discoverer module to obtain additional data from DBpedia.

5.1 Qualitative Evaluation

Faceted browsing is a popular technique in the Semantic Web community, since faceted browsers provide a convenient, user-friendly way of navigating through a wide range of data collections [50]. For the qualitative evaluation of LINDASearch, we selected a set of features typically present in faceted Web browsers. These features are classified and briefly described below:

1. Visual user interface

- **Friendly Interface (FI):** This type of visual interface is friendlier to non-experienced users than command-line interfaces, which require formal query expressions.
- **Customization (CU):** This means that users can customize their operations. For instance, since menus are simply data collections, users can employ collection management tools to add operations and rearrange or create menus as they wish.
- **Language Filtering (LF):** This function allows the user to choose a preferred display language, although English is usually chosen by default.

2. Search and navigation [40]

- **Parametric Search (PS):** It is an interface for faceted content allowing users to query by specifying constraints on facet values. Basically, this type of interface requires the

user to express a piece of information needed as a query in one shot, making selections across all facets of interest.

- **Faceted Navigation (FN):** It allows the user to progressively elaborate a query, seeing the effect of each choice in one facet on the available choices in other facets. Faceted navigation fills in the missing piece with results from the parametric search.
- **Discovering Links (DL):** It facilitates the exploration and collection of additional information through other datasets by using a property (predicate), *such as owl: sameAs, rdf: type, skos: exactMatch, foaf: based_near*. It basically allows expanding the exploration of the RDF graph to internal and external URIs that belong to other SPARQL entry points.
- **Navigation History and Recommendations (NHR):** It involves the exploitation of recorded navigation histories to help users perform searches; it specifically allows users to discover past search queries that are frequently associated with their active queries and past search queries that are related (i.e., syntactically similar) to their active queries.

3. Presentation results and geolocation

- **Pagination (PG):** It consists of a process that divides the content of results into pages.
- **Maps (MP):** This function is available for entities having location information; a map is shown with its coordinates. There are some geolocation systems such as Google Maps, GeoNames and OpenStreetMap that can be used for the map display.
- **Export Results (ER):** It consists in exporting the results obtained in additional formats, such as a spreadsheet, JSON, XML, RDF, N3, and JavaScript, among others.

Several works in the literature, which were also described in the State of the Art, have used these selected features. Among these faceted browsers we find the SWGET portal [17], SWOWS [19], Linked Open Graph (LOG) [21], LD Viewer [34], Linked Data Query Wizard [35], Discovery Hub [51], and Facete [29].

5.1.1 Qualitative evaluation design

This section describes the main features of LINDASearch as a faceted browser to conduct its qualitative evaluation. The evaluation consisted of three stages, and we asked for the assistance of experts in Web design, Semantic Web (Linked Data), and software engineering to perform it. Similarly, the stages comprised in the evaluation of LINDASearch were: 1) Identify the essential features of faceted search; 2) Create the weighted matrix according to the selected features and; 3) Classify the most important features

of a faceted search system, specifically for Linked Data; then, validate whether LINDASearch provides these features. Additionally, the third step included an exploratory search to complement the assessment.

For the first stage, we searched and identified the essential features of faceted search in terms of visual user interface, search and navigation, and presentation results and geolocations (see section 5.1). Then, for the second stage, we employed the Analytic Hierarchy Process (AHP) [52]. AHP is defined in [53] as the theory of relative measurement of intangible criteria. This approach uses paired comparisons, unlike the traditional measurement where some scale is applied to measure any element, and the elements are measured one by one, not by comparing them with one another. The AHP is useful for evaluating processes [54], selecting technology [55], or selecting product features [56]. In this work, we used AHP to both classify the most important features of Faceted Search and validate whether LINDASearch provided these features.

To conduct the AHP-based evaluation, the following activities had to be covered: 1) Select an expert panel; 2) Run a pair comparison between features; 3) Normalize values, and 4) Derive conclusions. A five-point Likert scale was used to run the comparison, where 1 stood for “equally important”, 2 for “slightly important”, 3 meant “more important”, 4 meant “considerably more important,” and 5 meant “extremely important.” Three experts, including a Web designer, a Semantic Web (Linked Data) specialist, and a software engineer met for this evaluation to generate the pair comparisons matrix as shown in Table 2. Table 3 shows the percentages calculated for the feature values once normalized.

Table 2. Essential Features for Faceted Browsers

FEATURE/VALUE	FI	CU	LF	PS	FN	DL	NH	PG	MP	ER
FI	1.00	7.00	2.00	0.50	1.00	0.30	3.00	1.00	0.50	2.00
CU	0.14	1.00	4.00	1.00	0.30	0.50	4.00	2.00	4.00	0.50
LF	0.50	0.25	1.00	0.50	0.50	1.00	2.00	2.00	1.00	5.00
PS	2.00	1.00	2.00	1.00	0.50	3.00	0.50	1.00	5.00	0.50
FN	1.00	3.33	2.00	2.00	1.00	1.00	2.00	0.20	3.00	2.00
DL	3.33	2.00	1.00	0.33	1.00	1.00	0.50	2.00	0.33	4.00
NHR	0.33	0.25	0.50	2.00	0.50	2.00	1.00	0.30	2.00	1.00
PG	1.00	0.50	0.50	1.00	5.00	0.50	3.33	1.00	0.30	5.00
MP	2.00	0.25	1.00	0.20	0.33	3.03	0.50	3.33	1.00	3.00
ER	0.50	2.00	0.20	2.00	0.50	0.25	1.00	0.20	0.33	1.00
TOTAL	11.81	17.58	14.20	10.53	10.63	12.58	17.83	13.03	17.46	24.00

Table 3. Percentages of Most Important Features Values (Normalized)

FEATURE	%	FEATURE	%
PG	12.42%	CU	11.41%
FN	11.61%	DL	10.39%
PS	11.59%	MP	10.11%
FI	11.46%		

We compared five well-known Faceted Search browsers with LINDASearch: Gnos, Elastic lists demo, Buzzillions, Forrester Research, and Air New Zealand. According to our panel of experts, the selected faceted browsers do possess the essential features of a faceted browser. For instance, Gnos uses semantic Web technologies and standards to structure and link all types of content among themselves and with other Linked Open Data (Linked Data). Similarly, Elastic lists demo is a demonstration of the elastic list principle [57] to browse multi-faceted data structures. On the other hand, Buzzillions is a product-review site that uses social tag-based facets in its navigation; it allows customers to refine results based on tags grouped as “Pros” or “Cons.” Forrester Research is similar to Buzzillions, but it focuses on business and technology research. It allows users to narrow their search by region, industry, and topic. However, in this faceted browser the values of each facet are not shown; visitors instead must open the facet category first to access them. Air New Zealand consists of an animated map that facilitates purchasing flight tickets by using facets.

The aim of evaluating the faceted browsers was to demonstrate that LINDASearch possesses the essential features of a Linked-Data-based Faceted Search system (see Table 3). More specifically, we only conducted this assessment to quantify the quality of our proposed design, and comparisons made with other Faceted Browsers were merely performed to demonstrate LINDASearch benefits.

- 1) **Friendly Interface:** LINDASearch provides a friendly, intuitive visual interface, even for the most unexperienced user (see Fig. 3).

Fig. 3. Screenshot of the Search Element

- 2) **Parametric Search:** With this functionality, users can query information specifying constraints to obtain the values that they want to obtain. These constraints can be provided through some input component, e.g. combo box, list box, text box, or even a facet (or set of facets) that permits gathering more information related to the user’s search. This is depicted in Fig. 3.

- 3) **Faceted Navigation:** This navigation type complements the parametric search. Once the parametric search is performed, the user can progressively collect information by using options provided by the facets, which are shown in the section of Additional Information (see Fig. 4).

Fig. 4. Screenshot of the search and faceted navigation elements

- 4) **Discovering Links (DL):** With LINDASearch, the ability to discover links and navigate within the LOD cloud to obtain additional information related the search is performed by the implementation of Silk Framework [45]. LINDASearch provides the Parametric search through the text box, combo box, and radio button components. Furthermore, the faceted navigation is provided by links (and discovered links) that allow the user to collect and navigate through information related to his/her search. This is depicted in Fig. 4.

- 5) **Pagination:** LINDASearch organizes the results in 30 rows by page. The aim of this organization is to display the content on multiple pages. The pagination buttons are located at the top and bottom of the results section (see Fig. 5).

Fig. 5. Screenshot of the pagination, maps and export results elements

- 6) **Maps:** when necessary, LINDASearch shows a Map within the Additional information section as part of the results obtained. This is depicted in Fig. 5.
- 7) **Export the Results:** LINDASearch allows users to export the results in XML and RDF formats.

5.1.2 Qualitative evaluation results

Table 4 shows the compared elements contained in some of the commercial faceted Web browsers previously described. For practical purposes, only the features selected as the most important for the qualitative evaluation (see Table 3) are shown in Table 4. As can be observed, LINDASearch and the compared faceted browsers contain these features. In detail, the Links Discoverer function is not present in all faceted browsers, since most of them only allow users to perform searches within a dataset and do not permit unifying multiple datasets to explore and obtain more information. Also, only LINDASearch allows for exporting results, namely in XML and RDF formats. Finally, we found that the Navigation History and Recommendations function is partially available for Buzillions, whereas the Language Filtering is only present in Gnos. None of the faceted browsers allows for interface customization.

Table 4. Comparison of LINDASearch with other Faceted Web Browsers

Faceted Web	Gnos	Elastic	Buzillions	Forrester	Air	LINDASearch
-------------	------	---------	------------	-----------	-----	-------------

Browser Features		lists		Research	New Zealand	
Friendly Interface	1	1	1	1	1	1
Parametric Search	1	0	1	1	1	1
Faceted Navigation	1	1	1	1	1	1
Discovering Links	0	0	0	0	0	1
Pagination	1	0	1	1	0	1
Maps	0	0	0	0	1	1
Export Results	0	0	0	0	0	1

5.2 Quantitative Evaluation

Information available in the LOD cloud can be exploited by the discovery of information relevant to user searches through LINDASearch. Hence, the quantitative evaluation presented in this section focused on the Links Discoverer module of LINDASearch, and it was performed following three steps: 1) Define links discovery experiments with and without the Levenshtein [58] and a stoplist (empty words); 2) Execute defined experiments using the Links Discoverer module; and 3) Perform an analysis of discovered links by applying precision and recall metrics.

5.2.1 Quantitative evaluation design

To map the search parameters to DBpedia, we used the keywords “Godzilla vs. the Sea Monster”.

As shown in Fig. 6, the procedure to map the keywords was as follows: 1) Generate a list of common stop words and abbreviations in English, including: *the, do, by, and, at, Ent.* (Entertainment), *VS* (versus), *Inc.* (Incorporated), *Ltd.* (Limited), and *Co.* (Company)”, to mention but a few; 2) Use the Levenshtein distance (minimum number of single-character changes) to transform one string of characters into another. The editing operations to perform these transformations are insertions, deletions, or substitutions; and 3) In the Links Discoverer module, select the property *owl:sameAs* applied to keywords “Godzilla vs. the Sea Monster.” Then, the module internally generates tokens “Godzilla,” “Sea,” and “Monster,” which are sent to DBpedia’s SPARQL endpoint (dbpedia.org), as shown in Fig. 6 and detailed in Table 5.

When the Levenshtein distance value was 0 and no stop words were added, it meant that synonyms were not used for the keywords. That is, only the complete keywords were used in experiments 1 and 3. Also, note that in addition to *owl:sameAs*, other LINDASearch properties can be used, such as *rdfs:seeAlso*, *dct:publisher*, *foaf:knows*, and *dct:source*, among others, yet we selected *owl:sameAs* be-

cause this property is useful for constructing relations between a word or phrases and a representative name of some entity. Moreover, *owl:sameAs* allows for searching information, it interlinks datasets [59], [60], preprocesses data for data mining, and makes the links discovery in the LOD cloud [61], [62]. However, in *owl:sameAs*, sometimes two entities are considered the same, but this is not applicable to all contexts [61], [63]. Furthermore, we assigned values 0 and 1 to the Levenshtein distance, since a distance greater than 1 decreases the possibility of discovering significant links. More specifically, as the changes for transforming one string of characters into another increase, we are more likely to obtain ambiguous links.

The LINDASearch Links Discoverer module shows the user the significant discovered links and the tokens. To achieve this, the module sums the highest values obtained for the key words and the tokens. Because LINDASearch does not know whether the user is interested in additional information related to the search, if the keyword is only a set of characters, the Links Discoverer module shows the links discovered. In addition, if the search includes several keywords, the Links Discoverer module divides the keywords into tokens, and for each token, it tries to find related links, yet such discovered links may not be directly related to the original search. In this evaluation, tokens “Sea” and “Monster” may not be related to the movie “Godzilla vs. the Sea Monster,” but token “Godzilla” surely does (see Fig. 6).

In this quantitative evaluation, the total number of links discovered by the Links Discoverer module was 216, whereas the number of discovered links directly related to the keywords was 80. This number resulted from adding the highest values for each keyword (token). However, after analyzing these links we removed those corresponding to token “Sea”, because this keyword was the least relevant to the search. In the end, 64 discovered links were relevant to “Godzilla vs the Sea Monster.”

Fig. 6. Screenshots of the obtained results from LINDASearch Links discoverer module

Table 5. Experiments for discovering links in LOD cloud

Keywords		No. Experiment	Levenshtein Distance	Stop words	Discovered Links
1	Godzilla vs. the Sea Monster	1	0	No	4
		2	0	Yes	19
		3	1	No	9
		4	1	Yes	12
2	Godzilla	1	0	No	16
		2	0	Yes	31
		3	1	No	15

		4	1	Yes	21
3	Sea	1	0	No	16
		2	0	Yes	9
		3	1	No	15
		4	1	Yes	11
4	Monster	1	0	No	5
		2	0	Yes	14
		3	1	No	7
		4	1	Yes	12
Total of discovered Links					216
Number of links discovered with information related to the search keywords					80
Number of relevant Links					64

5.2.2 Quantitative evaluation results

According to results shown in Table 5, a simple strings comparison (experiments 1 and 3) provided few discovered links for keywords 1, 2, and 4, whereas keyword 3 – a short word – reported the largest number of links discovered in experiment 1. These results revealed that when the stop words list was added (experiments 2 and 4), the number of discovered links increased with not too short keywords. For example, the largest number of discovered links were obtained with keyword 1 “Godzilla vs. the Sea Monster.” In addition, experiments with value 0 in the Levenshtein distance did not require changes to transform the keywords into their analogues with the filtered stop words, whereas experiments with value 1 in the Levenshtein distance did require to change one character to do the transformation.

Results from the links discovery experiments allowed us to estimate the number of discovered links relevant to the keyword and its tokens (see Table 5). However, we also performed a complementary validation to set precision percentages in the discovered links to identify which links truly contained information related to the search. For this validation stage, we adopted precision and recall metrics [64]. Precision refers to the proportion of retrieved cases that are classified as relevant. We obtained this score by dividing the number of links relevant to the search by the total number of discovered links. On the other hand, recall refers to the proportion of relevant cases retrieved. We obtained this score by dividing the number of links relevant to the search by the total number of discovered links (see Table 6).

Table 6. Quantitative evaluation results

Precision	0.80
Recall	0.29

When precision and recall metrics are applied, the ideal result is one in which both metrics have a high value (i.e., a value very close to 1). In simple terms, high precision means that an algorithm returns substantially more relevant than irrelevant results, while high sensitivity means that an algorithm returns most of the relevant results. In this sense, results shown in Table 6 indicate that of 216 links discovered by the Links Discoverer module, 80 were related to the search (i.e., “Godzilla vs. the Sea Monster,” tokens included), with a relatively high Precision of 80% but with a low Recall of 20%. This means that only 64 links discovered within DBpedia contained information substantially relevant and related to the keywords.

6. Conclusions and Future Work

In recent years, the Web evolution allowed Linked Data principles and practices to be adopted by an increasing number of data providers, which lead to the creation of a global data space on the Web. This Web evolution also brought new changes in the way data is accessed and utilized; and Web APIs have shown its impact on the creation of content from different Web of data sources. In addition, applications can operate on top of an unbounded set of data sources, via standardized access mechanisms through Linked Data. In this sense, tools and applications that help localize and search for specific information into Linked Datasets are of great importance to find and acquire relevant Semantically information to user.

This work presented LINDASearch, a system for semantic search, faceted navigation, data unification, discovering, and generation of search recommendation over information contained in the Open Linked datasets available in the Web of data. Hence, LINDASearch is an alternative focused to solve several limitations as: 1) limited searches to datasets of a single domain; 2) lack of faceted navigation and data unification from datasets of multiple domain; 3) lack of capacity to scale up the obtained results; 4) limitations to optimize the time of search information, to mention but a few. Also, this research involves the management, reuse, and development issues of new Linked Data applications with multiple data sets. In addition, the paper includes information on Linked Data, Linked Open Datasets, and a comprehensive State of the Art that presents several initiatives related to LINDASearch (including some commercial faceted browsers). Moreover, this article describes LINDASearch’s architecture, and the proposed architecture contributes to the foundations for querying multiple datasets in order to provide a simple form of integrat-

ing multiple data sources of the Linked Open Data (public available data). Therefore, LINDASearch permits querying and navigating multiple datasets, as well as reducing time in the development of new Linked Data applications. Finally, for the evaluation of LINDASearch, we followed a hybrid method consisting of an AHP-based qualitative evaluation to compare LINDASearch with other faceted browsers, and a quantitative evaluation based on precision and recall metrics to obtain the effectivity percentage for the discovery of links in the LOD cloud using the LINDASearch Links Discoverer module. As future work, it would be suitable to integrate additional modules to complement and improve the functionalities of LINDASearch. Among the functionalities to be incorporated are: 1) New datasets to cater for more parametric searches and navigate with facets through their data using more properties to facilitate the collection of additional information; 2) A customizable interface to provides users a friendlier and more visual interface; 3) A language switcher, although English is the language considered by default to provide users with the ability to search information in other languages; 4) The export of the results in more formats, such as N3 (N-Triples), JSON, and JavaScript; and 5) NLP (Natural Processing Language) support and multimodality to develop a LINDASearch mobile version.

Acknowledgments

Authors are grateful to the National Technological of Mexico for supporting this work. This research paper was also sponsored by the National Council of Science and Technology (CONACYT) and the Secretariat of Public Education (SEP) through PRODEP.

References

1. D. Allemang and J. Hendler, *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011.
2. T. Berners-Lee, J. Hendler, O. Lassila, and others, "The semantic web," *Sci. Am.*, vol. 284, no. 5, pp. 28–37, 2001.
3. T. Berners-Lee, "Linked Data," *Linked Data*, 2009. [Online]. Available: <https://www.w3.org/DesignIssues/LinkedData.html>. [Accessed: 16-Mar-2019].
4. T. Heath and C. Bizer, "Linked data: Evolving the web into a global data space," *Synth. Lect. Semant. web theory Technol.*, vol. 1, no. 1, pp. 1–136, 2011.
5. C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," in *Semantic services, interoperability and web applications: emerging concepts*, IGI Global, 2011, pp. 205–227.
6. S. Anam, B. H. Kang, Y. S. Kim, and Q. Liu, "Linked data provenance: State of the art and challenges," in *3rd Australasian Web Conference (AWC 2015)*, 2015, vol. 166, pp. 19–28.
7. G. Rodríguez-Enríquez, C.A., Alor-Hernández, G., Cortés-Robles, "LiDIA: A integration architecture to query Linked Open Data from multiple datasets," in *Workshops Proceedings in the Mexican International Conference on Computer Science (ENC 2013)*, 2013, pp. 64–68.
8. A. Freitas, J. G. Oliveira, S. O’Riain, J. C. P. da Silva, and E. Curry, "Querying linked data graphs using semantic relatedness: A vocabulary independent approach," *Data Knowl. Eng.*, vol. 88, pp. 126–141, 2013.
9. F. Lashkari, F. Ensan, E. Bagheri, and A. A. Ghorbani, "Efficient indexing for semantic search," *Expert Syst. Appl.*, vol. 73, pp. 92–114, 2017.
10. M. Achichi, Z. Bellahsene, M. Ben Ellefi, and K. Todorov, "Linking and disambiguating entities

- across heterogeneous RDF graphs,” *J. Web Semant.*, 2019.
11. P. Groth, A. Loizou, A. J. G. Gray, C. Goble, L. Harland, and S. Pettifer, “API-centric Linked Data integration: The Open PHACTS Discovery Platform case study,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 29, pp. 12–18, 2014.
 12. T. Özacar, “A tool for producing structured interoperable data from product features on the web,” *Inf. Syst.*, vol. 56, pp. 36–54, 2016.
 13. M. König, J. Dirnbek, and V. Stankovski, “Architecture of an open knowledge base for sustainable buildings based on Linked Data technologies,” *Autom. Constr.*, vol. 35, pp. 542–550, 2013.
 14. M. Radzinski, J. L. Sánchez-Cervantes, A. Garcia-Crespo, and I. Temiño-Aguirre, “Intelligent Architecture for Comparative Analysis of Public Companies Using Semantics and XBRL Data,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 24, no. 05, pp. 801–823, 2014.
 15. I. Ermilov, M. Martin, J. Lehmann, and S. Auer, “Linked Open Data Statistics: Collection and Exploitation,” in *Knowledge Engineering and the Semantic Web SE - 19*, vol. 394, P. Klinov and D. Mourmoutsev, Eds. Springer Berlin Heidelberg, 2013, pp. 242–249.
 16. N. Bikakis, C. Tsinaraki, I. Stavrakantonakis, N. Gioldasis, and S. Christodoulakis, “The SPARQL2XQuery interoperability framework,” *World Wide Web*, vol. 18, no. 2, pp. 403–490, Mar. 2015.
 17. V. Fionda, C. Gutierrez, and G. Pirrò, “The swget portal: Navigating and acting on the web of linked data,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 26, pp. 29–35, May 2014.
 18. V. Fionda, C. Gutierrez, and G. Pirrò, “Semantic Navigation on the Web of Data: Specification of Routes, Web Fragments and Actions,” in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 281–290.
 19. P. Bottoni and M. Ceriani, “SWOWS and dynamic queries to build browsing applications on linked data,” *J. Vis. Lang. Comput.*, vol. 25, no. 6, pp. 738–744, 2014.
 20. H. Ouksili, Z. Kedad, S. Lopes, and S. Nugier, “Pattern oriented RDF graphs exploration,” *Data Knowl. Eng.*, vol. 113, pp. 171–183, 2018.
 21. P. Bellini, P. Nesi, and A. Venturi, “Linked open graph: Browsing multiple SPARQL entry points to build your own LOD views,” *J. Vis. Lang. Comput.*, vol. 25, no. 6, pp. 703–716, 2014.
 22. B. Xu and H. Zhuge, “Automatic faceted navigation,” *Futur. Gener. Comput. Syst.*, vol. 32, pp. 187–197, 2014.
 23. B. Xu and H. Zhuge, “Faceted navigation through keyword interaction,” *World Wide Web*, vol. 17, no. 4, pp. 671–689, Jul. 2014.
 24. H.-J. Kim, Y. Zhu, W. Kim, and T. Sun, “Dynamic faceted navigation in decision making using Semantic Web technology,” *Decis. Support Syst.*, vol. 61, no. 0, pp. 59–68, 2014.
 25. Y. Zhu, D. Jeon, W. Kim, J. S. Hong, M. Lee, Z. Wen, and Y. Cai, “The Dynamic Generation of Refining Categories in Ontology-Based Search,” in *Semantic Technology*, 2013, pp. 146–158.
 26. N. Marie, F. Gandon, and M. Ribière, “Exploratory Search on the Top of DBpedia Chapters with the Discovery Hub Application,” in *The Semantic Web: ESWC 2013 Satellite Events SE - 45*, vol. 7955, P. Cimiano, M. Fernández, V. Lopez, S. Schlobach, and J. Völker, Eds. Springer Berlin Heidelberg, 2013, pp. 287–288.
 27. P. Fafalios, M. Salampasis, and Y. Tzitzikas, “Exploratory patent search with faceted search and configurable entity mining,” *Proc. Integr. IR Technol. Prof. Search (in conjunction with ECIR 2013)*, 2013.
 28. 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*, 2013, p. 2:1----2:8.
 29. C. Stadler, M. Martin, and S. Auer, “Exploring the Web of Spatial Data with Facete,” in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, 2014, pp. 175–178.
 30. S. Ferré and A. Hermann, “Semantic Search: Reconciling Expressive Querying and Exploratory Search,” in *The Semantic Web -- ISWC 2011*, 2011, pp. 177–192.
 31. S. Ferré and A. Hermann, “Reconciling faceted search and query languages for the Semantic Web,” *Int. J. Metadata, Semant. Ontol.*, vol. 7, no. 1, pp. 37–54, 2012.
 32. S. Ferré, “Expressive and Scalable Query-Based Faceted Search over SPARQL Endpoints,” in *The Semantic Web – ISWC 2014 SE - 28*, vol. 8797, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, Eds. Springer International Publishing, 2014, pp. 438–453.
 33. M. Arenas, B. Cuenca Grau, E. Kharlamov, S. Marciuska, and D. Zheleznyakov, “Faceted Search over Ontology-Enhanced RDF Data,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 939–948.
 34. D. Lukovnikov, C. Stadler, and J. Lehmann, “LD Viewer - Linked Data Presentation Framework,” in *Proceedings of the 10th International Conference on Semantic Systems*, 2014,

- pp. 124–131.
35. P. Hoefler, M. Granitzer, E. Veas, and C. Seifert, “Linked Data Query Wizard: A Novel Interface for Accessing SPARQL Endpoints,” *Proc. Linked Data Web WWW*, 2014.
 36. A. Latif, M. T. Afzal, P. Hoefler, A. U. Saeed, and K. Tochtermann, “Turning Keywords into URIs: Simplified User Interfaces for Exploring Linked Data,” in *Proceedings of the 2Nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, 2009, pp. 76–81.
 37. A. Latif, M. T. Afzal, A. U. Saeed, P. Hoefler, and K. Tochtermann, “CAF-SIAL: Concept aggregation framework for structuring informational aspects of linked open data,” in *2009 First International Conference on Networked Digital Technologies*, 2009, pp. 100–105.
 38. CODE, “Commercially Empowered Linked Open Data Ecosystems in Research,” 2014. [Online]. Available: <http://code-research.eu>. [Accessed: 16-Mar-2019].
 39. F. Stegmaier, C. Seifert, R. Kern, P. Höfler, S. Bayerl, M. Granitzer, H. Kosch, S. Lindstaedt, B. Mutlu, V. Sabol, K. Schlegel, and S. Zwicklbauer, “Unleashing Semantics of Research Data,” in *Specifying Big Data Benchmarks*, 2014, pp. 103–112.
 40. D. Tunkelang, “Faceted Search,” *Synth. Lect. Inf. Concepts, Retrieval, Serv.*, vol. 1, no. 1, pp. 1–80, 2009.
 41. J. Moreno-Vega and A. Hogan, “GraFa: Scalable Faceted Browsing for RDF Graphs,” in *The Semantic Web -- ISWC 2018*, 2018, pp. 301–317.
 42. N. Shah, D. Willick, and V. Mago, “A framework for social media data analytics using Elasticsearch and Kibana,” *Wirel. Networks*, 2018.
 43. M. Sattarian, J. Rezazadeh, R. Farahbakhsh, and A. Bagheri, “Indoor navigation systems based on data mining techniques in internet of things: a survey,” *Wirel. Networks*, vol. 25, no. 3, pp. 1385–1402, 2019.
 44. M. Schmachtenberg, H. Paulheim, and C. Bizer, “Adoption of Linked Data Best Practices in Different Topical Domains,” in *The 13th International Semantic Web Conference (ISWC2014), Lecture Notes in Computer Science*, vol. 8796, Riva del Garda-Trentino, Italy: Springer International Publishing, 2014, pp. 245–260.
 45. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov, “Silk-a link discovery framework for the web of data,” *LDOW*, vol. 538, 2009.
 46. M. Bauer, F., Kaltenböck, *Linked Open Data: The Essentials The Climate Knowledge Brokering Edition*, 1st ed. Vienna, Austria: edition mono/monochrom, 2016.
 47. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “DBpedia: A Nucleus for a Web of Open Data,” in *The Semantic Web*, 2007, pp. 722–735.
 48. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “DBpedia - A crystallization point for the Web of Data,” *J. Web Semant.*, vol. 7, no. 3, pp. 154–165, 2009.
 49. D. Law, “DESMET: a methodology for evaluating software engineering methods and tools,” *Comput. Control Eng. J.*, vol. 8, no. 3, p. 120–126(6), 1997.
 50. M. Hildebrand, J. van Ossenbruggen, and L. Hardman, “/facet: A Browser for Heterogeneous Semantic Web Repositories,” in *The Semantic Web - ISWC 2006*, 2006, pp. 272–285.
 51. 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*, 2013, pp. 17–24.
 52. R. W. Saaty, “The analytic hierarchy process—what it is and how it is used,” *Math. Model.*, vol. 9, no. 3, pp. 161–176, 1987.
 53. T. L. Saaty, “Analytic Hierarchy Process,” in *Encyclopedia of Biostatistics*, American Cancer Society, 2005.
 54. L.-A. Vidal, F. Marle, and J.-C. Bocquet, “Using a Delphi process and the Analytic Hierarchy Process (AHP) to evaluate the complexity of projects,” *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5388–5405, 2011.
 55. S. Lee, W. Kim, Y. M. Kim, and K. J. Oh, “Using AHP to determine intangible priority factors for technology transfer adoption,” *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6388–6395, 2012.
 56. N. Subramanian and R. Ramanathan, “A review of applications of Analytic Hierarchy Process in operations management,” *Int. J. Prod. Econ.*, vol. 138, no. 2, pp. 215–241, 2012.
 57. M. Stefaner and B. Muller, “Elastic lists for facet browsers,” in *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*, 2007, pp. 217–221.
 58. V. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Sov. Phys. Dokl.*, vol. 10, no. 8, pp. 707–710, 1966.
 59. W. Beek, S. Schlobach, and F. van Harmelen, “A Contextualised Semantics for owl:sameAs,” in *The Semantic Web. Latest Advances and New Domains*, 2016, pp. 405–419.
 60. G. Correndo, A. Penta, N. Gibbins, and N. Shadbolt, “Statistical Analysis of the owl:sameAs Network for Aligning Concepts in the Linking Open Data Cloud,” in *Database and Expert*

- Systems Applications*, 2012, pp. 215–230.
61. H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson, “When owl:sameAs Isn’t the Same: An Analysis of Identity in Linked Data,” in *The Semantic Web -- ISWC 2010*, 2010, pp. 305–320.
 62. K. Kagawa, S. Tamagawa, and T. Yamaguchi, “An Automatic sameAs Link Discovery from Wikipedia,” in *Semantic Technology*, 2014, pp. 399–413.
 63. J. L. Sánchez-Cervantes, G. S. Hernández-Chan, M. Radzinski, J. M. Gómez-Berbís, and Á. García-Crespo, “Discovering and Linking Financial Data on the Web,” in *IARIA, 2013, The Second International Conference on Data Analytics (DATA ANALYTICS 2013)*, 2013, pp. 36–40.
 64. K. M. Ting, “Precision and Recall,” in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, p. 781.