Searching Research (Meta-)Data using Semantic Web Technologies

Sarah Bensberg and Marius Politze

IT Center, RWTH Aachen University

Research data can be described with RDF based metadata following discipline specific application profiles. To make the best use of this data stocks according to the FAIR principles, users should be able to search within these metadata sets. Among different approaches it is investigated whether mapping the RDF data into a search index affects the quality of search. The evaluation indicates that approaches are either easy to use but slow or ineffective, or fast and effective but difficult to use. With the transformation of RDF data into a search index, a solution was found that combines benefits of the approaches.

1 Introduction

Digital transformation affects all areas of society: More data is produced and workflows rely on data analysis leading to new challenges in data management. Within the context of research data management, the National Research Data Infrastructure aims to systematize research data stocks and make them accessible. RWTH Aachen University supports this effort with the development of the research data management platform Collaborative Scientific Integration Environment (Coscine): Research data is made accessible independent of the actual storage location and described with metadata that allows a structured search. The goal is to make the data Findable, Accessible, Interoperable and Reusable (FAIR) [1].

1.1 FAIR Metadata

An important aspect is the diversity and heterogeneity of the various scientific disciplines as well as the decentralized nature of data and IT Services [2]. Discipline-specific application profiles are required and should be available to researchers early in the Research Data Life Cycle (RDLC) since implicit knowledge is not passed on and often lost [3]. These application profiles are typically based on existing metadata standards.

The FAIR Digital Object provides a framework for the layout and interleaving of identifiers, metadata and data using a describing record as part of a data identifier [4] in such

Publiziert in: Vincent Heuveline, Nina Bisheh (Hg.): E-Science-Tage 2021. Share Your Research Data. Heidelberg: heiBOOKS, 2022.
DOI: https://doi.org/10.11588/heibooks.979.c13730 (CC BY-SA 4.0)

a distributed system: Several infrastructure components are available for this purpose: (i) the Persistent Identifier (PID) for registration, permanent and unique referencing, (ii) external "linked" metadata records stored with the PID, which enables the use of discipline-specific systems, (iii) a storage system in which researchers can store the actual bit sequences of their research data, and (iv) the means to publish the results in a suitable repository.

To simplify data exchange and machine processing, the World Wide Web Consortium specifies Semantic Web technologies like Resource Description Framework (RDF) [5] to provide the means to describe digital resources. RDF is suitable because its schema-lessness provides flexibility for changes and different disciplines. In order to follow defined metadata schemas, RDF needs to be restricted by application profiles that allow validation as offered by Shapes Constraint Language (SHACL) [6]. While this makes metadata available as a RDF-based knowledge graph, it presents some general challenges described by Arnaout and Elbassuoni [7]:

- *Data incompleteness*, since RDF triples contain a lot of information, but most knowledge is represented as free text in literal nodes.
- *Inflexible querying*, because triple-pattern queries are highly expressive, but at the same time restrictive, because they have to follow a certain structure.
- *Missing result ranking*, queries may yield results in arbitrary order; a ranking is not provided by SPARQL Protocol And RDF Query Language (SPARQL) and has to be added explicitly.
- *Result diversity*, because it is important that the topmost results give a broad overview of the results of a query and do not all contain similar aspects.

The searchability and thus the sub-goal reusability of the FAIR Guiding Principles is achieved by an associated query language SPARQL [8], requiring considerable technical knowledge. There are two main approaches to overcome this obstacle: the systematic generation of a query or the mapping of RDF (meta-)data into a search index for use in a search engine.

1.2 Hypothesis

Based on the state of research on RDF searches or general Information Retrieval (IR) and taking previous approaches into account, the work is oriented towards the following hypothesis, which is to be tested:

Mapping RDF-based metadata records into a search index for use in a search engine improves the quality of the search and results for the user compared to using generated SPARQL queries.

To validate this hypothesis, different approaches to create an efficient semantic search engine for metadata in RDF-based knowledge graphs are evaluated concerning the resulting quality of the search.

1.3 Assessment of the Quality of Search

In this context, "quality" is regarded in multiple dimensions ranging from user experience when entering their search intention to the results returned and the technical abilities of the system.

Effectiveness According to International Organization for Standardization (ISO) [9]. Referring to search results, a system should find correct data records and satisfy the search intention.

Usability According to ISO [10]. For search, it is therefore primarily a matter of the User Interface (UI) to be appealing and understandable for the user.

Complexity of the creation of a search query is for the user and expected preliminary knowledge.

Efficiency According to ISO [9]. For search, this refers to the effort formulating a query compared to the generated results.

Response Time Nielsen [11] defines two limits: 0.1s for a direct response to an interaction and 1.0s to notice the delay but remain uninterrupted.

Scalability It is expected, that the data stock is growing over time. Hence, the system should be able to handle growing quantities of data.

Additional Effort This requirement assesses how much additional storage or computing resources are necessary to realize the search.

2 Related Work

To generate a better overview, related approaches are divided into different categories, whereby a clear separation is not always possible or ambiguous.

2.1 Retrieval of Entities in the Semantic Web Using Keywords

Swoogle is a crawler-based indexing and retrieval system for RDF documents [12]: an IR system is applied, which uses either character *n-grams* or Uniform Resource Identifiers (URIs) as terms to find documents given the search terms and computes a rank indicating the documents' importance. *Sindice* indexes RDF documents based on resource URIs, Inverse Functional Propertys (IFPs) and keywords [13]. The focus of the approach is retrieval of ontologies and documents in the Semantic Web. Similarly, *SWSE* is a search engine that allows finding and navigating in an object-oriented search space by keyword-based input [14]. *Falcons* provides keyword-based search for concepts and objects on the Semantic Web by indexing all the terms of an entity's virtual document consisting of its names, associated literals, and names of neighboring entities [15]. While in the

context at hand, metadata records are directly persisted in an RDF triplestore and do not need to be indexed, the approaches demonstrate the distinction between object-based and document-based search.

2.2 Generating a SPARQL Query

A very basic approach is to generate a SPARQL query from the keyword-based search of a user where predefined basic graph pattern templates and a mapping of keywords to possible URIs and background information about the existing data structure are used [16]. This shows that background knowledge and structures are helful to generate adequate queries.

SPARQL query builders allow a step-by-step construction of SPARQL queries using cleverly chosen user interfaces. Kuric et al. have compared the best-known query builders regarding their usability for laypeople and divided them into three different categories [17]: (i) *Form-Based Query Builders* use input fields to build the SPARQL query step by step. Classes and objects must either be advised or made available for selection by the user. *ExConQuer* [18], the *Linked Data Query Wizard* [19], *VizQuery* [20], and the *Wikidata Query Service (WQS)* [21] are examples of such approaches. (ii) *Graph-Based Query Builders* make use of a visual and graphical user interface that guides the user in creating valid SPARQL queries. Some examples of these approaches are *iSPARQL* [22], *NITELIGHT* [23], *OptiqueVQS* [24], and *QueryVOWL* [25]. (iii) *Natural Language-Based Query Builders* build the query based on the natural language of the user. *NLP-Reduce* [26], *SPARKLIS* [27], *DEANNA* [28], *AskNow* [29], *Swip* [30] and *ScoQAS* [31] are examples for a Natural Language (NL)-based approaches that also allow generation of SPARQL queries. However, the apporaches focus on enhancing NL approaches and not so much on the quality of search so these are excluded from further consideration.

Faceted Search is used to allow flexible navigation through a large search space and to limit this space by so-called facets [32]. Faceted search can also be applied to RDF knowledge graphs by offering properties (predicates) of entities (subjects) as facets with corresponding values (objects). *OSCAR* [33], the OpenCitations RDF Search Application, and *SemFacet* [34] are examples of such an approach.

The *Assisted SPARQL Editor* leverages the graph summary to support the user in effectively formulating complex SPARQL queries [35]. For this purpose, suggestions for classes, predicates, relationships between variables, and named graphs are provided to the user. *SemSearch* is a search engine that translates user input from a simple syntax into a formal SPARQL query [36]. While loosing much of the expressiveness of SPARQL. *SINA*, a keyword-based search system, uses a hidden markov model to transform input into a SPARQL query [37]. *AutoSPARQL* uses active supervised machine learning to formulate a SPARQL query [38].

2.3 Generating a Search Index to use IR Techniques

Linked swissbib converts bibliographic data into a RDF-based data model and divides it into six different concepts [39]. By using the *JSON-LD* serialization of RDF, they are indexed in the search engine *Elasticsearch (ES)*. *Open Semantic Search* stores the associated labels of the RDF annotation to URIs for indexing data in Solr and allows a faceted search without generting a SPARQL query. *Semplore* also uses a hybrid query option that allows structured queries and faceted searches [40]. Rocha et al. presented a hybrid approach to keyword-based search in the Semantic Web, where the focus is also on finding concepts using an instance graph for each concept, which is searched with traditional IR techniques [41].

3 Approaches for Searching in RDF Metadata

If RDF data is stored in a triple store, the easiest way to access it is the SPARQL Endpoint that allows querying and filtering data. A user must have some knowledge to do this: (i) RDF and SPARQL, (ii) the structure of the stored data, (iii) used vocabularies and terms as well as their corresponding URIs or Internationalized Resource Identifiers (IRIs)[1]. Even if a user would have these information, the use of SPARQL is very expressive and therefore challenging. It can thus be assumed that the interface can only be used by more experienced Data Curators. Unfortunately, many of the presented approaches are outdated, no longer available or there is no code or exact description for their implementation and realization[2], are domain-specific and would have to be widely adapted for use in the domain independent scenario.

3.1 Full-Text Search

SPARQL offers the possibility to filter data using regular expressions. However, a SPARQL query must be generated internally, which applies the user input to all possible occurrences of a literal somewhere in the graph. Virtuoso also offers the option to do a case insensitive full-text search with the `bif:contains` function [42]. Since only indexed literals can be searched with `bif:contains`, it is not possible to define additional rules which compound the literals on the fly in the constructed SPARQL query. Therefore, not all literals in the graph can be searched without onstructed literals and additional triples directly in the triplestore. However, a strategy would have to be considered how to separate the original metadata record, which is why this variant is not further considered.

[1] An IRI is a generalization of an URI. Since both are used very interchangeably in general linguistic usage, only the term URI is used in the following, even if a IRI would be allowed.

[2] Ironically, this is exactly one of the phenomenons, that should be tackeled by an Research Data Management (RDM) system.

3.2 Faceted Search

The approach of a faceted search is especially interesting because it allows for an incremental refinement of data. At the core, this approach generates a SPARQL query that, by selecting facets and values, directly using URIs and data structures of the knowledge graph. Implementations should consider further aspects like the selection of facts and their representation in the UI. In the case of Coscine these could be metadata fields from application profiles, but the question arises how to select the currently visible application profiles. In the scope of this work, however, the focus is on the result which can be achieved with such an approach rather than the generation of an adequate UI.

3.3 Query Builder

The related work shows a lot of different query builders, which all work a bit differently or have a different focus. They will be evaluated in the work because they can provide the most optimal and direct representation in the knowledge graph by gradually assembling a SPARQL query. Various such approaches have already been extensively evaluated [43, 17, 44]. In general, the generated SPARQL queries are to be considered for the evaluation.

3.4 Elastic Search: Index and Search Engine

Creating a search index allows to benefit from functionalities and optimizations for user input and search features like auto-completions, search suggestions, stemming or tolerance towards spelling errors. Positively evaluating this approach would therefore support the core hypothesis.

A conceptual challenge remains how to map the entities from RDF to documents. The mapping offered by Semplore [40] was enhanced to allow custom inference rules, and the possibility to use ES to benefit from advanced search types like value range queries. The resulting model considers resources, the metadata records, as documents in the search index. In principle this defines an entity-object-mapping as basis for the conversion from RDF to the search index.

To define queries, ES provides a Domain Specific Language (DSL) based on JSON. ES offers two different search syntaxes to directly convert user inputs into a corresponding search query. A simpler variant is ignoring syntactically incorrect input while the advanced syntax is offers more search features while being stricter in terms of syntax.

4 Evaluation

In order to evaluate the discussed approaches for searching RDF metadata a two step approach is followed. Firstly they are used generate SPARQL queries that match 13

different search intentions, assuming the user has all necessary knowledge to produce an optimal result. Secondly the generated queries are used for evaluation against a sample dataset.

4.1 Dataset and Evaluation Environment

For the following evaluation of the different approaches, 10.000 exemplary metadata records were generated based on Records from DBPedia. They are based on an application profile based on the EngMeta metadata schema [45].

In the context of the evaluation and for testing the quality of the search results a set of search intentions were considered. The intentions were selected in a way to cover many different search types encountered in the context of RDM e.g.:

- Records with version number 10
- Records about computer science
- Records which are available since 03.07.2020
- Records about political left
- Records which contain a variable with the value 2 meters
- Records about object-oriented software which are published before 2015

Respective search queries are constructed for the approach, that resemble each search intention as close as possible assuming the user not to make any mistakes in this process and achieves a close to optimal result. For the ES approaches, RDF data is first mapped to JSON objects and then inserted into the search index.

4.2 Comparison of the Effectiveness

Based on the sample search queries, confusion matrices were constructed. Table 1 summarizes the results by calculating average precision, recall, and F1 score of the total confusion matrices of the respective approaches.

Table 1: Precision, recall, and F1 score (rounded to two digits)

	Regex	Bif:contains	Query Builder	Faceted Search	ES Simple	ES Advanced
Preciscion	0.6	0.77	0.96	0.88	0.66	0.96
Recall	1	0.67	1	0.9	0.98	1
F1 Score	0.75	0.72	0.98	0.89	0.79	0.98

4.3 Comparison of the Usability

For comparison, the *task execution time* estimated by means of the Keystroke-level Model (KLM), the simplest of the Goals, Operators, Methods, and Selection rules (GOMS) family [46] was considered. The KLM [47] describes various actions with associated physical

operations and specifies an execution time for them. Tasks are converted into operations and the execution time is added up. Table 2 shows average execution times across the approaches.

Table 2: Average task execution time in s based on the KLM

	Regex	Bif:contains	Query Builder	Faceted Search	ES Simple	ES Advanced
⌀	7.79	7.12	34.15	2.86	8.51	17.7

For Faceted Search only the time of the second step is calculated. For the first step, the same time is valid as for the respective search intention in the regex approach.

4.4 Comparison of Complexity

Sepecific preliminary knowledge to express search intentions is quite different: Regex requires egular expression syntax, `Bif:contains` requires Specific Bif:contains operators, Query Builder requires SPARQL Syntax (optional) and knowledge graph structure, Faceted Search requires Regular expressions for preliminary full text search (optional), ES Simple] requires ES simple search syntax and ES Advanced requires ES advanced search syntax.

Individual search queries were considered per approach and classified into categories. The most often selected category was used as an indicator for a final comparison of the complexity as shown in Table 3.

Table 3: Complexity for user to perform search requests ($\surd$ = easy, $-$ = complex)

Regex	Bif:contains	Query Builder	Faceted Search	ES Simple	ES Advanced
$\surd$	$\surd$	$-$	$\surd$	$\surd$	$\surd$

4.5 Compasrison of Efficiency

The categories of effectiveness, usability, and complexity are put in relation to each other. The ranking for the efficiency results from the ratio output/input, where output is the effectiveness and input is the combination of usability and complexity. Table 4 gives a ranking in the categories and overall efficiency.

4.6 Comparison of Response Time

The average wall clock times from sending the request to returning the result set were measured for a small dataset. The standard deviation was calculated to illustrate the inaccuracies of measurements as shown int Table 5.

Table 4: Ranking of effectiveness, usability, complexity, and efficiency whereby the approaches in a higher row are better ranked

Rank	Effectiveness	Usability	Complexity	Efficiency
1	Query Builder ES Advanced	Regex Bif:contains ES Simple	Regex Bif:contains Faceted Search ES Simple ES Advanced	ES Simple ES Advanced
2	Faceted Search	Faceted Search ES Advanced	Query Builder	Faceted Search
3	ES Simple	Query Builder		RegEx
4	Regex			Quiery Builder
5	Bif:contains			Bif:contains

Table 5: Average response time in ms (rounded to a full number) of user requests across all intentions.

	Regex	Bif:contains	Query Builder	Faceted Search	ES Simple	ES Advanced
⌀	401±26	346±24	311±20	327±24	553±30	555±28

For Faceted Search only the time of the second step is calculated. For the first step, the same time is valid as for the respective search intention in the regex approach.

4.7 Comparison of Scalability

To measure scalability of the approaches, the previously presented search queries were executed on the entire 10,000 metadata records the same way described the previous section. The results are shown in Table 6

Table 6: Response time in ms (rounded to a full number) of user requests in large search data record.

	Regex	Bif:contains	Query Builder	Faceted Search	ES Simple	ES Advanced
⌀	2604±97	357±21	335±38	1084±37	552±33	552±27

For Faceted Search only the time of the second step is calculated. For the first step, the same time is valid as for the respective search intention in the regex approach.

4.8 Comparison of Additional Effort

For the SPARQL query builder and the faceted search, no further arrangements need to be made. For the *regex* and *bif:contains* approach, only the literals need to be specified and stored. For ES, in addition to this, the mapping of metadata records to the search index is required additional processing times for different actions on the metadata are shown in Table 7.

Table 7: Times in ms for the additional effort for indexing in ES.

(Re)-Indexing	Add	Update	Delete
1646706±53376	837±143	839±129	735±121

5 Conclusion

Based on the evaluation criteria the approaches were ranked. The order is a direct result of the measured times or classifications without normalizing them or considering further gradations (i.e., to classify similar values equally) in the individual categories. To get a better overview of the results, a preference matrix as shown in Table 8, is used to look at how often one approach is better or worse than another.

Table 8: Preference matrix to compare different approaches where the numbers indicate how often the approach of the top beats the one to be compared

	Regex	Bif:contains	Query Builder	Faceted Search	ES Simple	ES Adv.
Regex	-	2	4	3	3	3
Bif:cont.	1	-	4	3	3	3
Query B.	3	3	-	3	4	4
Faceted S.	4	3	3	-	4	4
ES Simp.	2	3	3	2	-	2
ES Adv.	2	2	2	1	2	-
Win/Loss	11/15	13/14	16/17	12/17	16/11	16/10
Diff	-4	-1	-1	-5	+5	+6

The extended ES approach performs best. Next comes the simple variant of ES, and after that the SPARQL query builder together with the *bif:contains* approach. This largely confirms the hypothesis: The mapping of RDF-based metadata records into a search index for use in a search engine improves the quality of the search for the user, compared to the use of generated SPARQL queries. Essentially, the same results can be achieved with less effort for the user.

In conclusion, the main finding of this work is that the use of search engines can also be suitable for searching in RDF-based knowledge graphs if an adequate entity-object-mapping is applied. In the case at hand a mapping was derived from and enhanced.

The consecutive step is the full integration of the search functionality in the research data management platform Coscine. In addition, an appropriate user interface for displaying the found metadata records must be considered. Exemplary ES specific search interfaces also include ES features like auto-complete or search suggestions [48], which reduce the error rate and help the user to enter the most optimal search query.

Additionally, some tests or considerations for the optimal configuration of ES could still be done. Here, for example, the indexing and search analyzers, tokenizers, synonyms, and ranking functions could be considered to enhance results and rankings.

Supplementary Information

The datasets and applications used for the evaluation are available:

- "Search Engine Evaluation for Research Data in an RDF-based Knowledge Graph", DOI: https://doi.org/10.18154/RWTH-2020-09885.
- "Sample Dataset for Search Engine Evaluation for Research Data in an RDF-based Knowledge Graph",https://doi.org/10.18154/RWTH-2020-09886.
- "RDF-based Knowledge Graph Mapping for Elastic Search", https://doi.org/10.18154/RWTH-2020-09884.

A more extensive report on the background, especially the entity-object-mapping is available in the master thesis by Sarah Bensberg: "An Efficient Semantic Search Engine for Research Data in an RDF-based Knowledge Graph". 2020. DOI: https://doi.org/10.18154/RWTH-2020-09883.

Acknowledgments

The work was partially funded with resources granted by NFDI4Ing and Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 432233186 – AIMS.

The authors would like to thank Prof. Stefan Decker and Prof. Matthias S. Müller for supervising the master thesis that served as a basis for this report.

Bibliography

[1] Marius Politze, Florian Claus, Bela Brenger, M. Amin Yazdi, Benedikt Heinrichs, and Annett Schwarz. How to manage it resources in research projects? towards a collaborative scientific integration environment. In Yves Epelboin, Michele Mennielli, Anna Pacholak, Pekka Kähkipuro, Gill Ferell, Carmen Diaz, Ligia M. Riberio, Johan Bergström, Thierry Koscielnieak, Elsa Cardoso, Raimund Vogl, Bas Cordewener, Noel Wilson, Carla Vilarrasa, Nicole Harris, and Outi Tasala, editors, *European Journal of Higher Education IT 2020-2*. 2020.

[2] Dominik Schmitz and Marius Politze. Forschungsdaten managen – bausteine für eine dezentrale, forschungsnahe unterstützung. *o-bib. Das offene Bibliotheksjournal / Herausgeber VDB*, 5(3):76–91, 2018. https://doi.org/10.5282/o-bib/2018H3S76-91

[3] Marius Politze, Sarah Bensberg, and Matthias Müller. Managing discipline-specific metadata within an integrated research data management system. In Joaquim Filipe, editor, *Proceedings of the 21st International Conference on Enterprise Information Systems ICEIS 2019, Heraklion, Crete - Greece, May 3 - 5, 2019*, ICEIS (Setúbal). SciTePress, 2019. https://doi.org/10.5220/0007725002530260

[4] Koenraad de Smedt, Dimitris Koureas, and Peter Wittenburg. Fair digital objects for science: From data pieces to actionable knowledge units. *Publications*, 8(2):21, 2020. https://doi.org/10.3390/publications8020021

[5] Richard Cyganiak, David Wood, and Markus Lanthaler, eds. RDF 1.1 Concepts and Ab- stract Syntax. 2014. (Visited on 02/20/2020).

[6] Shapes constraint language (shacl). URL:https://www.w3.org/TR/shacl/.

[7] Hiba Arnaout and Shady Elbassuoni. Effective searching of rdf knowledge graphs. *SSRN Electronic Journal*, 2018. https://doi.org/10.2139/ssrn.3199315

[8] Sparql 1.1 overview. URL:https://www.w3.org/TR/sparql11-overview/.

[9] En iso 9000:2015 d/e quality management systems— fundamentals and vocabulary.

[10] Iso 9241-11:2018(en) ergonomics of human-system interaction — part 11: Usability: Definitions and concepts.

[11] Jakob Nielsen. *Usability engineering*. Academic Press, Boston, 1993.

[12] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle. In David Grossman, Luis Gravano, ChengXiang Zhai, Otthein Herzog, and David A. Evans, editors, *Proceedings of the Thirteenth ACM conference on Information and knowledge management - CIKM '04*, New York, New York, USA, 2004. ACM Press. https://doi.org/10.1145/1031171.1031289

[13] Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata Semantics and Ontologies*, 3(1), 2008. https://doi.org/10.1504/IJMSO.2008.021204

[14] Andreas Harth, Aidan Hogan, Jurgen Umbrich, and Stefan Decker. *SWSE: Objects before documents*. 2012.

[15] Gong Cheng, Weiyi Ge, and Yuzhong Qu. Falcons: Searching and browsing entities on the semantic web. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, New York, NY, USA, 2008. Association for Computing Machinery. https://doi.org/10.1145/1367497.1367676

[16] Saeedeh Shekarpour, Soren Auer, A.-C. Ngonga Ngomo, Daniel Gerber, and Claus Stadler. Keyword-driven sparql query generation leveraging background knowledge. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2011. https://doi.org/10.1109/WI-IAT.2011.70

[17] Emil Kuric, Javier D. Fernández, and Olha Drozd. Knowledge graph exploration: A usability evaluation of query builders for laypeople. In Maribel Acosta, Philippe Cudré-Mauroux, and Maria Maleshkova, editors, *Semantic Systems. The Power of AI and Knowledge Graphs*, Information Systems and Applications, incl. Internet/Web, and HCI, Cham, 2019. Springer International Publishing. https://doi.org/10.1007/978-3-030-33220-4_24

[18] Judie Attard, Fabrizio Orlandi, and Sören Auer. Exconquer: Lowering barriers to rdf and linked data re-use. *Semantic Web*, 9(2), 2018. https://doi.org/10.3233/SW-170260

[19] Patrick Hoefler, Michael Granitzer, Eduardo Veas, and Christin Seifert. Linked data query wizard: A novel interface for accessing sparql endpoints. *7th Workshop on Linked Data on the Web 2014*, 2014.

[20] Vizquery - hay's tools. URL: https://hay.toolforge.org/vizquery/.

[21] Wikidata query service. URL: https://query.wikidata.org/.

[22] Openlink isparql. URL: http://dbpedia.org/isparql/.

[23] Paul R Smart, Alistair Russell, Dave Braines, Yannis Kalfoglou, and Nigel R. Shadbolt. A visual approach to semantic query design using a web-based graphical query designer. 2008. https://doi.org/10.1007/978-3-540-87696-0_25

[24] Ahmet Soylu, Evgeny Kharlamov, Dmitriy Zheleznyakov, Ernesto Jimenez-Ruiz, Martin Giese, Martin G. Skjæveland, Dag Hovland, Rudolf Schlatte, Sebastian Brandt, Hallstein Lie, and Ian Horrocks. Optiquevqs: A visual query system over ontologies for industry. *Semantic Web*, 9(5), 2018. https://doi.org/10.3233/SW-180293

[25] Florian Haag, Steffen Lohmann, Stephan Siek, and Thomas Ertl. Queryvowl: Visual composition of sparql queries. In Fabien Gandon, Christophe Guéret, Serena Villata, John Breslin, Catherine Faron-Zucker, and Antoine Zimmermann, editors, *The semantic web: ESWC 2015 satellite events*, Lecture notes in computer science Computer communication networks and telecommunications, Cham and Heidelberg and New York, 2015. Springer. https://doi.org/10.1007/978-3-319-25639-9_12

[26] Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. Nlp-reduce: A naive but domainindependent natural language interface for querying ontologies. In *4th European Semantic Web Conference ESWC*, 2007.

[27] Sébastien Ferré. Sparklis: An expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8(3), 2016. https://doi.org/10.3233/SW-150208

[28] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. *Natural Language Questions for the Web of Data*. EMNLP-CoNLL '12. Association for Computational Linguistics, USA, 2012.

[29] Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Höffner, and Jens Lehmann. Asknow: A framework for natural language query formalization in sparql. In Harald Sack, Eva Blomqvist, Mathieu d'Aquin, Simone Paolo Ponzetto, Christoph Lange, and Chiara Ghidini, editors, *The semantic web*, Lecture notes in computer science Theoretical computer science and general issues, Cham and Heidelberg, 2016. Springer. https://doi.org/10.1007/978-3-319-34129-3_19

[30] Camille Pradel, O. Haemmerlé, and N. Hernandez. Demo: Swip, a semantic web interface using patterns. 2013.

[31] Majid Latifi, Horacio rodriguez, and Miquel Sànchez-Marrè. Scoqas: A semantic-based closed and open domain question answering system. *Procesamiento de Lenguaje Natural*, 59, 2017.

[32] Hearst Marti A. Uis for faceted navigation: Recent advances and remaining open problems. 2008.

[33] Ivan Heibi, Silvio Peroni, and David Shotton. Enabling text search on sparql endpoints through oscar. *Data Science*, 2(1-2), 2019. https://doi.org/10.3233/DS-190016

[34] Marcelo Arenas, Bernardo Cuenca Grau, Evgeny Kharlamov, Šarūnas Marciuška, and Dmitriy Zheleznyakov. Faceted search over rdf-based knowledge graphs. *Journal of Web Semantics*, 37-38, 2016. https://doi.org/10.1016/j.websem.2015.12.002

[35] Stéphane Campinas. *Graph summarisation of web data: data-driven generation of structured representations*. PhD thesis, NUI Galway, 2016.

[36] Yuangui Lei, Victoria Uren, and Enrico Motta. Semsearch: A search engine for the semantic web. In Steffen Staab and Vojtěch Svátek, editors, *Managing knowledge in a world of networks*, volume 4248 of *Lecture notes in computer science Lecture notes in artificial intelligence*. Springer, Berlin, 2006. https://doi.org/10.1007/11891451_22

[37] Saeedeh Shekarpour, Edgard Marx, Axel-Cyrille Ngonga Ngomo, and Sören Auer. *SINA: Semantic Interpretation of User Queries for Question Answering on Interlinked Data*, volume 30. 2015. Semantic Search. https://doi.org/10.1016/j.websem.2014.06.002

[38] Jens Lehmann and Lorenz Bühmann. Autosparql: Let users query your knowledge base. In Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter de Leenheer, and Jeff Pan, editors, *The semantic web: research and applications*, Lecture Notes in Computer Science, Berlin, 2011. Springer. https://doi.org/10.1007/978-3-642-21034-1_5

[39] Markus Mandalka. Open semantic search: Your own search engine for documents, images, tables, files, intranet & news. URL: https://www.opensemanticsearch.org//.

[40] Lei Zhang, Qiaoling Liu, Jie Zhang, Haofen Wang, Yue Pan, and Yong Yu. Semplore: An ir approach to scalable hybrid query of semantic web data. In Karl Aberer, editor, *The semantic web*, Lecture Notes in Computer Science, Berlin, 2007. Springer. https://doi.org/10.1007/978-3-540-76298-0_47

[41] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi Aragao. A hybrid approach for searching in the semantic web. In Stuart Feldman, Mike Uretsky, Marc Najork, and

Craig Wills, editors, *13th International Conference on World Wide Web Conference*, New York, N.Y., 2005. ACM Press. https://doi.org/10.1145/988672.988723

[42] 16.3.1.using full text search in sparql. URL: http://docs.openlinksw.com/virtuoso/rdfsparqlrulefulltext/.

[43] Pavel Grafkin, Mikhail Mironov, Michael Fellmann, Birger Lantow, Kurt Sandkuhl, and Alexander V. Smirnov. SPARQL query builders: Overview and comparison. In Björn Johansson and Filip Vencovský, editors, *Joint Proceedings of the BIR 2016 Workshops and Doctoral Consortium co-located with 15th International Conference on Perspectives in Business Informatics Research (BIR 2016), Prague, Czech Republic, September 14 - 16, 2016*, volume 1684 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[44] Adam Styperek, Michal Ciesielczyk, Andrzej Szwabe, and Pawel Misiorek. Evaluation of sparql-compliant semantic search user interfaces. *Vietnam Journal of Computer Science*, 2(3), 2015. https://doi.org/10.1007/s40595-015-0044-y

[45] Engmeta - beschreibung von forschungsdaten | informations- und kommunikationszentrum | universität stuttgart. URL: https://www.izus.uni-stuttgart.de/fokus/engmeta/.

[46] Shiroq Al-Megren, Joharah Khabti, and Hend S. Al-Khalifa. A systematic review of modifications and validation methods for the extension of the keystroke-level model. *Advances in Human-Computer Interaction*, 2018, 2018. URL: https://www.hindawi.com/journals/ahci/2018/7528278/, https://doi.org/10.1155/2018/7528278

[47] David Kieras. *Using the Keystroke-Level Model to Estimate Execution Times*. 2003.

[48] Suggesters | elasticsearch reference [7.9] | elastic. URL: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-suggesters.html.