Federated Search Engine for Open Educational Linked Data

Maedeh Mosharraf and Fattaneh Taghiyareh

***Abstract*—Driven by the success of linked data, interlinked web of data is a best infrastructure for distributing open educational resources (OERs). Hand in hand with this structure, immense technological challenges are arising in every phase of their publishing, maintaining, discovering, and accessing. Utilizing OERs licenses and principals of linked data, this paper proposes a federated search engine to retrieve OERs published on web of data. Transforming natural language queries to the SPARQL form using a desirable interface and several pre-prepared queries, parsing each query and broadcasting the sub-queries to several appropriate open repositories, merging retrieved results, and presenting to users in an appropriate user friendly interface are processes of this system. It has successfully passed primary tests and started to be used in a technology enhanced learning laboratory; however, its development is still on-going.**

***Index Terms*—Federated search engine, linked data, OER, SPARQL**

I. Introduction

Introducing the concept of Open Educational Resources (OERs) and subsequently changing the approach of educational service providers leads to positive steps to eliminate economic, demographic, and geographic barriers of education [1]. The appearance of MOOCs, which is a paradigm in learning process, and promoting lifelong learning programs are propounded by presenting this concept [2].

Like other learning resources, OERs are distributed on the web and can be accessed through different mechanisms adjusted by providers. It is essential that learning resources, storages, providers, and services be organized in such a way that instead of being single-use or accumulating worthless, have the ability to be re-used. Such a possibility has also been achieved for some open source software [3]. In addition, according to the importance of OERs and especially documents in knowledge transferring and self-regulated learning [4], their management and retrieval are so important. On the other hand, the thought behind OERs is maximum sharing. However, inconsistency in content management approaches applied in various repositories leads to lack of resources interactivity. The lack of interactivity is the origin of other problems such as losing resource discoverability, reusability, remixability, and adaptability [5].

Electrical and Computer Engineering Department, University of Tehran, Iran. m.mosharraf@ut.ac.ir,ftaghiyar@ut.ac.ir

Providing a basic recipe for connecting and publishing data on the web, linked data helps to solve these problems. Linked data principles proposed by Berners-Lee are [6]:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information
4. Include links to other URIs, so that they can discover more things

While the primary units of the hypertext web are HTML documents connected by un-typed hyperlinks, linked data relies on data described in RDF format. Using several vocabularies, each entity can be described and related to other entities in the world of linked data. Each vocabulary is a collection of classes and properties, which are described by RDF, models domains of interest in various degrees of expressivity [7].

Although resources are structurally described on the web of data, different providers use various vocabularies for modeling and publishing them on linked data. For this reason, using resources originated from various repositories requires to know several vocabularies. Also, each provider describes different properties of their data and models them using different features. Utilizing these resources can be accomplished through either web interfaces which have many limitations or SPARQL endpoints. Obtaining data through SPARQL queries also needs to know this language.

Using search engines can facilitate the problem of accessing open educational linked data. However, the retrieval mechanism of existing search engines is based on document indexing. In addition, they cannot benefit from the features of linked data and answer some complicated questions. In the scope of eLearning two of these queries are:

- Open educational slides in the domain of eLearning which have been produced by Iranian professors.
- Learning contents related to personalization which have been produced in 2010 and has been re-edited.

Retrieving resources which are described in these questions is possible only with the use of linked data and navigating through several links between different repositories. In addition, linked data provides a monolithic index that must contain all the content and relationships for an extremely expansive body of resources. So, it comes with some indexed based discovery engines limits. Fig. 1 shows a brief overview of metadata and relations of different resources published in linked data.

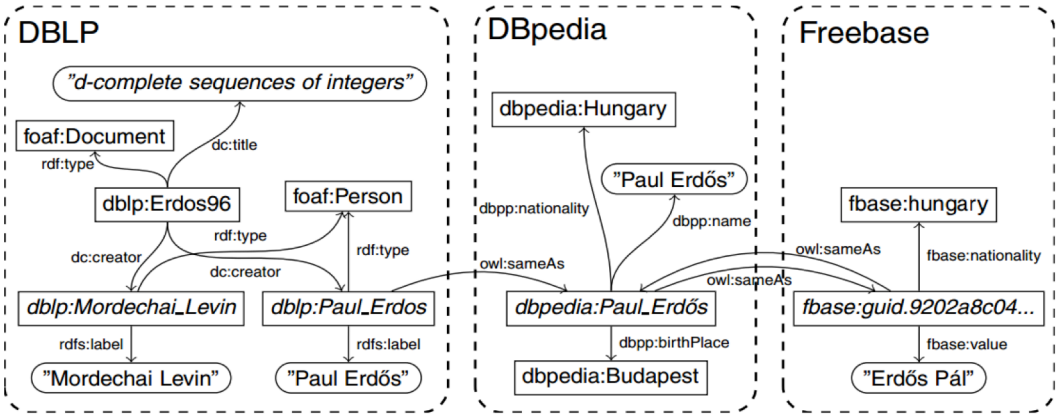

Fig. 1. RDF triples from three Linked Open Data sources [9]

According to [8], Linked Open Data offers new solutions for educational resources, partly solving some of the OER limits. (1) Open data are interlinked by definition, so a semantic layer is created to describe OERs, (2) federated search to find resources belonging to distinct datasets are the most important outcome of linked data.

In this paper, we propose a system to answer different queries about OERs published in linked data. In addition to search on educational resources which are modeled in the web of data, this system can federate queries on separated resources and answer to complicated questions. The implemented system can also be beneficial in these situations:

- Reducing users confusion in searching different repositories
- Reducing time of search
- Facilitating access to resources distributed in different repositories
- Providing different options related to users' queries from different repositories and specifying type of relations
- Focusing on metadata and full text from open access repositories
- Offering appropriate resources which are not in the query language of users. (Sometimes inadequacy or lack of quality of the retrieved resources makes users to fetch resources provided in other languages. Doing it can be possible through navigating links between multilingual data published on the web of data.)

The implemented system has many applications. Integrating with Learning and Content Management Systems (LMSs and CMSs) in order to provide free access to various resources, with authoring systems in order to help to reuse and generate new contents, with library systems in order to retrieve related resources to entered quires can be useful. In addition, it can be a step towards the expanding universe of open linked data.

The rest of the paper is organized as follows. Section 2 presents the architecture of our proposed federated search engine and functions of its components. Section 3 describes the implementation details and repositories utilized in the primary evaluation. Finally, section 4 concludes the paper and outlines areas for future research.

II. PROPOSED SYSTEM

Our system is being developed to provide an efficient solution for federated query processing on open educational linked data. The word "federate" means "to join together" or "to unite". The federated search system allows a user to submit a single query and receive results from multiple sources, without having to query each of the sources individually. This system consists of (1) mapping a user query to one or several SPARQL queries, (2) broadcasting the SPARQL query to a group of disparate repositories with the appropriate format, (3) merging the results collected from different repositories, and (4) presenting the results in a unified and user-friendly format with minimal duplication. Fig. 2 shows the architecture of the implemented system.

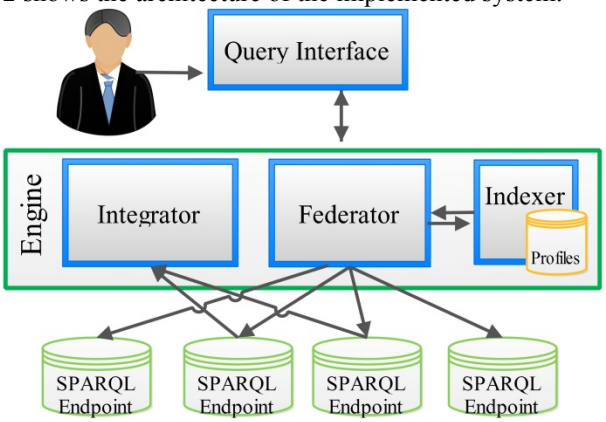

Fig. 2. Architecture of the federted query search engine

In the following each part of this engine is presented.

A. *Mapping the input query in the SPARQL format*

Transforming users' queries in natural language to SPARQL is a complicated task and requires knowing complexities, rules, terms, phrases, and exceptions of natural language. Natural language processing techniques, which our system should be enriched to, use large corpuses contained examples of natural language queries as well as logic rules. To avoid natural language processing, users' queries are entered into the system using the specific fields provided in the user

interface. In an interview conducted among different users, we ask them to determine all the information may need in the learning or research activities through searching the internet. Accumulating demands of these survey participants, including 5 high school students, 5 undergraduate students, 5 graduate students, 5 teachers, and 5 researchers, all the educational questions can be categorized in 5 groups (or combination of them): Requests resources or data by specifying a set of specifications, Requests information about a/several persons by determining a set of characteristics, Questions about dates, Questions about publishers, and Questions about subjects.

Considering some specified fields in the system user interface, all the simple and complex queries can be constructed. Knowing educational repositories and sets of vocabularies which model OERs, several SPARQL queries which have not been initialized are determined as the system default. Entering each field of a user request, the proportional variable in the SPARQL queries is set. Fig. 3 illustrates part of a not initialized query.

```
PREFIX ...
SELECT ?title ?date ?name ?subject ?abstract ?link
WHERE{
        ?author foaf:name ?name.
        ?author dbo:notableWork ?work.
        ?work rdfs:label ?title.
        FILTER (regex(str(?name), "", "i"))
        FILTER (regex(str(?title), "", "i"))
        ?work dbp:releaseDate ?date;
        dcterms:subject ?subject;
        dbo:wikiPageExternalLink ?link;
        dbo:abstract ?abstract.
        FILTER (regex(str(?suject), "", "i"))
        FILTER (regex(str(?abstract), "", "i"))
}
```

Fig. 3. Part of a SPARQL primary query which is not initialized

B. Broadcasting the SPARQL query to different SPARQL endpoints

Each primary query can be parsed into sub-queries that can be answered by individual repositories. Indexer and Federator, as the two main modules of our system, are the responsibility of determining suitable repositories and parsing the query in accordance to their metadata, respectively.

Profiling each repository, Indexer determines types of resources and other information which can be accessed through it. Types of metadata that can be harvested using vocabularies are other fields of repositories profiles stored in Indexer. This data is defined manually after checking each repository metadata. Using repositories profiles and both variables and vocabularies of each query, Indexer determines repositories that the query should be sent to.

Considering vocabularies applied in the determined repositories, Federator parses the primary query. For each repository, Federator filters parts of the query that contains vocabularies not being used in it. The primary query consists of several triples. Each triple utilizes one, two, or rarely tree vocabularies. Each triple remains in the sub-queries, if all of their vocabularies are applied in the repository.

Let's explain this phase with an example. Checking the profile of DBLP shows that vocabularies used for describing resources of this repository include “Swrc”, “rdf”, “owl”, “d2r”, “xsd”, “dcterms”, “rdfs” ,”map”, “foaf”, and “dc”. Therefore, after initializing the primary query using user requested values, each triple that consists of another vocabulary is removed before sending the query to DBLP. “dbo” and “dbp” are vocabularies used in the primary query illustrated in Fig.3 and not engaged in DBLP profile.

C. Merging the results obtained from each SPARQL endpoint

The results obtained from the distributed sub-queries are merged in Integrator and finally returned in an aggregated form. In this respect, all the results retrieved from each repository are saved in the temporary database of Integrator. These results are in the form of RDF and with the understandable meta-information for machines. Navigating these typed triples, distributed results are linked using “owl:sameAs” and “rdfs:seeAlso” relations. At best, a connected graph is created by integrating all the distributed results. Applying the primary query in the generated graph, this module prunes all the extra information, which is non-relevant to the input query.

D. Presenting the results

The retrieved results of query execution should be represented to the user in a uniform and understandable format. It is possible that contrary to the usual representation method, the user wants to see the RDF graph of the results to obtain other information. Therefore, the results are represented to the user in two interfaces:

- The results are displayed in a typical way as other search engines. In our approach, all the main information is displayed in one phrase. Clicking each phrase, all the other meta-information is provided for the user. The user can adjust the displaying order by subject, date, and alphabetical.
- The RDF graph of the results is displayed. This view is basically an adequate way to establish an abstraction for the underlying data schema. Its’ benefit is providing the possibility of easy reuse and exchange of sub-graphs, recursive view definitions, and applications for data integration from distributed repositories.

III. IMPLEMENTATION AND EVALUATION

For implementing the federated search engine on linked data, Microsoft ASP.NET MVC 3 framework and Microsoft Visual Studio 2012 as IDE was chosen. In addition to default libraries and packages of ASP.NET MVC 3, many other free and open source java and JavaScript libraries such as jena, jQuery, jQuery UI, and jScrollPane are were used. The main reason for choosing ASP.NET MVC 3 framework was its MVC based (Model-View-Controller) pattern and object-oriented aspect of C# programming language which makes it reasonably easy to build standard and scalable web applications.

TABLE 1
SOME SPECIFICATIONS OF SOME OER REPOSITORIES

Repository	Address	#Resources	#People	#Triples
DBpedia	http://dbpedia.org/sparql	12325452	1840596	438038746
DBLP	http://dblp.l3s.de/d2r/snorql/	10038	10036	444184
ACM	http://acm.rkbexplorer.com/sparql/	Not defined	Not defined	Not defined
Charles University in Prague	http://linked.opendata.cz/sparql	Not defined	444536	557106351
University of Southampton	http://sparql.data.southampton.ac.uk/	4812696	341764	6749073
University of Muenster (LODUM)	http://data.uni-muenster.de/sparql/	56034	11657	4248725
Aalto University	http://data.aalto.fi/endpoint	21794	226572	1649357
OxPoints (University of Oxford)	https://data.ox.ac.uk/sparql/	11644	91	631546
ASN:US	http://sparql.jesandco.org:8890/sparql	1358	0	7497309

Up to now, many OERs in different formats have been published in the web of linked data. Datahub[1] as a free, powerful data management platform from the Open Knowledge Foundation provides a tool for accessing many of them. Using this platform, we can find many SPARQL endpoints covering data from different domains and integrate them with our search engine. In addition, our system directly connects to some of open repositories through their SPARQL endpoints. In these repositories, providers of educational data publish their data or its model freely on the web of data on their own. Table 1 shows specifications of these repositories.

The system passes the primary test by applying on the introduced educational linked data. Depending on the internet bandwidth and the availability of resources, time of query execution and retrieved results are different. In eleven different test queries, 1200 milliseconds and 29 seconds are respectively the minimum and maximum times spent from receiving the user queries to representing their results. Although, experiments showed that increasing number of repositories used in this system can increase execution time; the retrieved results are more desirable. This system is being used as a pilot for promoting other research focused on semantic processes enhances educational technologies.

IV. CONCLUSION

The federated search engine offers a scalable mechanism for querying open educational Linked Data. However, there are some challenges in its processes that should carefully be examined.

- There may be some natural language queries which cannot fit in the search fields of the user interface. Considering mechanisms of entering these types of queries and transforming to the SPARQL format are of the future works of the project.
- Increasing open repositories as well as their variance and execution time will increase the time cost of system processing. Grouping resources and changing in the Indexer approach can be beneficial. It is worth stating that the main cost of our system is the query execution time and cost of transmitting query and results between the system and linked data repositories. The worst-case cost of integrating results is defined as complexity times $O(N^2)$, where N is the number of graph nodes.
- The query execution time in each repository and retrieving non-related responses are challenges originated from the simple mechanism of query parsing in Federator. Applying some logics for optimal and efficient execution of sub-queries in each repository can be beneficial.
- Indexer contains profile information of repositories. So, it should be updated synchronously with any changes in each repository.
- Another challenge is SPARQL endpoints availability and timeout. It is hoped that this problem will be resolved with advances in research and applications of linked data.

Considering these challenges in the future works of this project, improving the implemented system may continue in other different ways. Enriching to mechanism for crawling links and finding resources described by specified relations, and promoting by the process of semantic query expansion using different domain ontologies are samples.

[1] https://datahub.io/about

REFERENCES

[1] M. Mosharraf and F. Taghiyareh, "The Role of Open Educational Resources in the eLearning movement," *Knowledge Management & E-Learning*, vol. 8, p. 10-21, 2016.

[2] A. Mesquita and P. Peres, *Furthering Higher Education Possibilities through Massive Open Online Courses*. IGI Global, 2015.

[3] J. Kuriakose and J. Parsons, "An Enhanced Requirements Gathering Interface for Open Source Software Development Environments," in *Proceedings of IEEE 23rd International Requirements Engineering Conference (RE)*, Ottawa, 2015.

[4] N. Hood, A. Littlejohn, and C. Milligan, "Context counts: How learners' contexts influence learning in a MOOC," *Computers & Education*, vol. 91, p. 83–91, 2015.

[5] N. Piedra, J. Chicaiza, J. López, and E. Tovar , "Using linked open data to improve the search of open educational resources for engineering students," in *Proceedings of IEEE Frontiers in Education Conference*, Oklahoma City, 2013.

[6] T. Berners-Lee," Linked Data- Design Issues", 2006, from: http://www.w3.org/DesignIssues/LinkedData.html .

[7] K. Janowicz, P. Hitzler , B. Adams , D. Kolas , and C. Vardeman, "Five Stars of Linked Data Vocabulary Use," *Semantic Web* , vol. 5, pp. 173-176 , 2014.

[8] D. Taibi, G. Fulantelli, S. Dietze, and B. Fetahu, "Educational Linked Data on the Web -Exploring and Analysing the Scope and Coverage," in *Open Data for Education*. Switzerland: Springer International Publishing , 2016, p. 16–37.

[9] O. G¨orlitz and S. Staab, "Federated Data Management and Query Optimization for Linked Open Data", in *New Directions in Web Data Management*. Springer Berlin Heidelberg, 2011, ch. 5, pp. 109-137.