Pre-print of:
G. Kadilierakis, C. Nikas, P. Fafalios, P. Papadakos, Y. Tzitzikas,
Elas4RDF: Multi-perspective Triple-centered Keyword Search over RDF using Elasticsearch,
Demo Paper, Proceedings of the 17th Extended Semantic Web Conference (ESWC'2020),
June 2020, Heraklion, Crete

Elas4RDF: Multi-perspective Triple-centered Keyword Search over RDF using Elasticsearch

Giorgos Kadilierakis[1,2], Christos Nikas[1,2], Pavlos Fafalios[2], Panagiotis Papadakos[1,2], and Yannis Tzitzikas[1,2]

[1] Information Systems Laboratory, FORTH-ICS, Heraklion, Greece,
[2] Computer Science Department, University of Crete, Heraklion, Greece
kadilier@csd.uoc.gr, {cnikas, fafalios, papadako, tzitzik}@ics.forth.gr

Abstract. The task of accessing knowledge graphs through structured query languages like SPARQL is rather demanding for ordinary users. Consequently, there are various approaches that attempt to exploit the simpler and widely used keyword-based search paradigm, either by translating keyword queries to structured queries, or by adopting classical information retrieval (IR) techniques. This paper demonstrates `Elas4RDF`, a keyword search system over RDF that is based on `Elasticsearch`, an out-of-the-box document-centric IR system. `Elas4RDF` indexes and retrieves triples, and thus yields more refined and informative results, that can be viewed through different perspectives. In the presentation we will demonstrate the performance of `Elas4RDF` system in queries of various types, and we will showcase the benefits from offering different approaches for aggregating and visualising the search results.

1 Motivation and Novelty

The Web of Data contains thousands of RDF datasets available online, including cross-domain KBs (e.g., DBpedia and Wikidata), domain specific repositories (e.g., DrugBank and MarineTLO), as well as Markup data through schema.org (see [4] for a recent survey). These datasets are queried through complex structured query languages, like SPARQL. Faceted Search is a user-friendlier paradigm for interactive query formulation, however the systems that support it (see [6] for a survey) need a keyword search engine as an entry point to the information space. Consequently, and since plain users are acquainted with web search engines, an effective method for keyword search over RDF is indispensable.

At the same time we observe a widespread use of out-of-the-box IR systems (e.g., `Elasticsearch`) in different contexts. To this end we investigate how these, document-centric Information Retrieval Systems (IRSs), can be used for enabling keyword search over arbitrary RDF datasets. This endeavor raises various questions revolving around: (a) how to index an RDF dataset, (b) what to rank and how, and (c) how the search results should be presented.

We will demonstrate `Elas4RDF`, a keyword search system over RDF that is based on the popular IR system `Elasticsearch`. Our main research question, as elaborated in the conference paper [3], was: "can `Elasticsearch` be configured to offer a retrieval performance comparable to that of dedicated keyword search systems for RDF?". Here, we describe and demonstrate a system that is based on that approach, that additionally focuses on the presentation / aggregation of the search results. Specifically, the retrieved RDF triples are displayed through various visualisation methods named "tabs", that provide different perspectives of the search space and can satisfy different information needs.

The most relevant work to ours is the LOTUS system [2], a keyword search system over RDF data that is also based on `Elasticsearch`. However, its main focus is on scalability, while we focus on effectiveness (see [3]) and the support of various types of search through different views. With respect to user-friendly interfaces, there are systems focusing on particular aspects (e.g., faceted search). To the best of our knowledge though, there are no available prototypes that offer keyword access and multiple methods for inspecting the search results.

2 Indexing, Retrieval, and Evaluation

As detailed in the conference paper [3], we opt for high flexibility and thus consider *triple* as the retrieval unit. A triple is more informative than an entity. It can be viewed as the simplest representation of a fact that verifies the correctness of a piece of information for Q&A tasks. Furthermore, it offers flexibility on how to structure and present the final results, which is the focus of this work.

For *indexing*, we evaluated variations of two main approaches on what data to consider for each *virtual document*. The *baseline* approach considers only data from the triple itself (i.e., text extracted from the subject, object and predicate). The *extended* approach exploits information in the neighbourhood of the triple's resource elements, like one or more descriptive properties such as *rdfs:label* and *rdfs:comment*. Regarding the *retrieval process* we have experimented with various *query types*, *weighting methods* and *similarity models* offered by `Elasticsearch`.

We have *evaluated* the above using the DBpedia-Entity test collection[3], which is based on a DBpedia dump of 2015-10. The collection contains a set of heterogeneous keyword queries that cover four categories: i) named-entity queries (e.g., "Brooklyn bridge"), ii) IR-style keyword queries (e.g., "electronic music genre"), iii) natural language questions (e.g., "Who is the mayor of Berlin?"), and iv) entity-list queries (e.g., "professional sports teams in New York)". In total, over 49K query-entity pairs are labelled using a three-point scale; 0 for irrelevant, 1 for relevant, and 2 for highly relevant.

The key results from the evaluation are: i) all triple components contribute to the system's performance; ii) object keywords seem to be more important than subject keywords, thus giving higher weight to the object fields can improve performance; iii) extending the index with additional descriptive information about the triple URIs improves performance; however, including all available information (all outgoing properties) introduces noise and drops performance; iv) the default similarity model of `Elasticsearch` (BM25) performs satisfactory; v) using `Elasticsearch` for keyword search over RDF data is almost as effective as task- and dataset-oriented systems built from scratch.

The approach and the evaluation results are presented in detail in [3].

[3] `https://iai-group.github.io/DBpedia-Entity/`

3 The Elas4RDF Search System

3.1 Indexing Service and Search REST API

For enabling the community and other interested parties to use our approach over arbitrary RDF datasets, we have made publicly available two dedicated Elas4RDF services:

Elas4RDF-index Service.[4] Creates an index of an RDF dataset based on a configuration that states the data to index (i.e., baseline and/or extended approaches described in [3]). The index can then be queried by the **Elas4RDF-search** service.

Elas4RDF-search Service.[5] Service that exploits an `Elas4RDF-index` and initialises a REST API which accepts keyword queries and returns results in JSON format. Apart from the *query*, the list of parameters optionally includes: i) the *size* of the answer, ii) the name of the *index* to consider (from `Elas4RDF-index`), iii) the *type* of the answer (triples, entities, both), iv) the index *field* over which to evaluate the query (e.g., only over the subject), and v) a *body* parameter through which one can express a complicated DSL query.[6]

The `Elas4RDF-search` service is used by the `Elas4RDF` search system for retrieving the results of a keyword query and presenting them to the user through different visualisation methods (more details below). Thus, one can easily configure it to use a search service over another dataset. A demo of the `Elas4RDF` system over DBpedia is available at: `https://demos.isl.ics.forth.gr/elas4rdf/`

3.2 Multi-Perspective Presentation of Search Results

The presentation and visualisation of RDF data is challenging due to the complex, interlinked, and multi-dimensional nature of this type of data [1]. An established method on how to present RDF results for arbitrary query types does not exist yet, and it seems that a single approach can not suit all possible requirements. For this reason we offer a *multi-perspective* approach. *Multiple tabs* are used for the presentation of keyword search results, where each one stresses a different aspect of the hits. The user can easily inspect all tabs and get a better overview and understanding of the search results. Figure 1 shows the search results for the query "`El Greco paintings`", as presented in each of the four currently-supported tabs. Below, we give more details for each tab.

Triples Tab. A ranked list of triples is displayed to the user, where each triple is shown in a different row. For visualising a triple, we create a snippet for each triple element (subject, predicate, object). The snippet is composed of: i) a title (the text indexed by the baseline method), ii) a description (the text indexed by the extended index; if any), and iii) the URI of the resource (if the element is a

[4] `https://github.com/SemanticAccessAndRetrieval/Elas4RDF-index`
[5] `https://github.com/SemanticAccessAndRetrieval/Elas4RDF-search`
[6] `https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html`

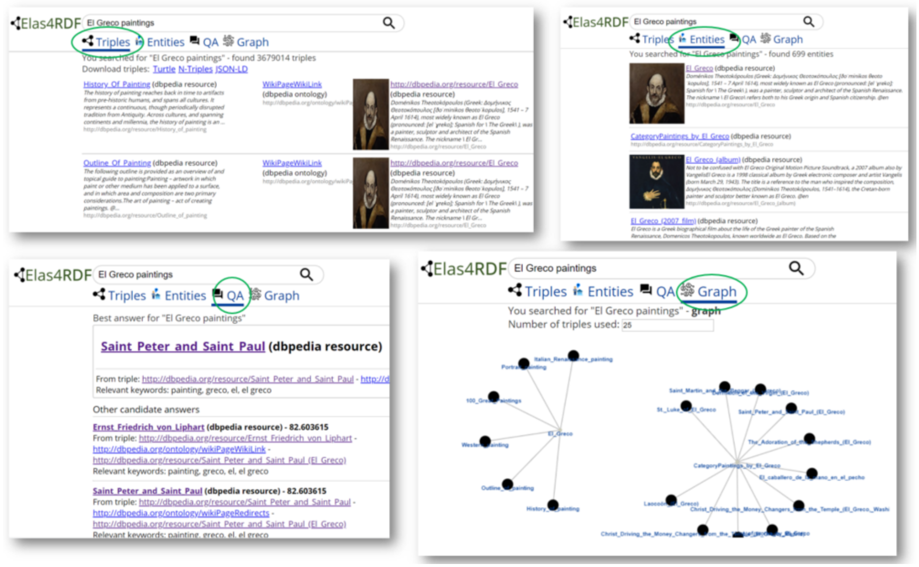

Fig. 1. Search results for the query "El Greco paintings".

resource). If the triple element is a resource, its title is displayed as a hyperlink, allowing the user to further explore it. We also retrieve and show an image of the corresponding entity (if any), which is usually provided in DBpedia.

Entities Tab. Here the retrieved triples are grouped based on entities (subject and object URIs), and the entities are ranked following the approach described in [3] (which considers the weighted gain factor of the ranking order of the triples in which the entities appear). Then, a ranked list of entities is displayed to the user, where each entity is shown in a different row. For visualising an entity, we create the same snippet like previously. The title is displayed as a hyperlink, since the entities are resources, allowing the user to further explore the entity.

Graphs Tab. Here the retrieved triples are visualised as a graph using the JavaScript InfoVis Toolkit.[7] In this way the user can see how the triples are connected. The default size of the graph is 25 nodes, however the user can increase or reduce this number.

Question Answering (QA) Tab. Here we attempt to interpret the user's query as a question and provide a single compact answer. The challenge is to retrieve the most relevant triple(s) and then extract natural language answers from them. QA over structured data is a challenging problem in general and currently only a few kinds of questions are supported by this "under-development" tab. It returns the more probable answer accompanied by a score, plus a list of other possible answers. In our running example, this tab returns the title of one painting of El Greco, while for the query "`Who developed Skype?`" it returns as more probable answer "`Microsoft`" and the next possible answer is "`Skype Technologies`".

[7] `https://philogb.github.io/jit/`

3.3 Demonstration Scenarios

We will demonstrate various kinds of queries for showcasing the added value that each tab brings. Below, we briefly discuss the added value of each tab for the indicative query $q1$ = `Crete and Mars`.
• **Triple's Tab**: This tab is generally the most useful one since the user can inspect all components of each triple, and understand the reason why that triple is returned. The addition of images help to easily understand which triples involve the same entities. For the query $q1$ the user gets more than 600K triples that involve the name Crete (island) and Mars (mythical god, planet, etc.).
• **Entities' Tab**: If the user is interested in entities, and not in particular facts, this view provides the main entities. For the query $q1$ the returned entities include the island of Crete, an area of Mars whose name is related to Crete, Administration Area of Crete, Battle of Crete, and others.
• **Graph's Tab**: This tab allows the user to inspect a large number of triples without having to scroll down. Moreover this view reveals the grouping of triples, and whether there is one or more poles and interesting insights. For example, for $q1$ the user can see the connection of Crete with Mars (mythology), through a resource about the Battle of Crete: Mars was the mythical codename of a group of the Operation Mercury (Nazi's invasion to Crete in WWII).
• **Q&A Tab**: The result of this view for the query $q1$ is "Classical albedo features on Mars", which is related to the naming of an area of Mars, according to a classic albedo feature based on Crete, the island where Icarus lived.

Other queries that will be demonstrated include "`drugs containing aloe`", "`Which cities does the Weser flow through?`", "`Rivers of Greece`", etc.

4 Closing Remarks

`Elas4RDF` is a triple-centric keyword search system over RDF data. It can applied to a plethora of RDF datasets since it is schema agnostic, it can be configured easily, and "inherits" the maturity and scalability features of `Elasticsearch`. The multi-perspective presentation of the search results allow tackling various kinds of information needs and the participants will be able to issue their own queries and explore the information space of DBpedia through the prisms of triple, entities, graph and Q&A tabs. More tabs will be added in the near future, while we also plan to test the system over the domain specific knowledge repositories of GRSF [7] and ClaimsKG [5].

Acknowledgements

This work has received funding from the European Union's Horizon 2020 innovation action BlueCloud (Grant agreement No 862409).

References

1. Dadzie, A.S., Pietriga, E.: Visualisation of linked data–reprise. Semantic Web **8**(1), 1–21 (2017)

2. Ilievski, F., Beek, W., van Erp, M., Rietveld, L., Schlobach, S.: LOTUS: Adaptive text search for big linked data. In: ESWC. pp. 470–485. Springer (2016)
3. Kadilierakis, G., Fafalios, P., Papadakos, P., Tzitzikas, Y.: Keyword Search over RDF using Document-centric Information Retrieval Systems. In: ESWC (2020)
4. Mountantonakis, M., Tzitzikas, Y.: Large-scale Semantic Integration of Linked Data: A Survey. ACM Computing Surveys (CSUR) **52**(5), 103 (2019)
5. Tchechmedjiev, A., Fafalios, P., Boland, K., Gasquet, M., Zloch, M., Zapilko, B., Dietze, S., Todorov, K.: Claimskg: A knowledge graph of fact-checked claims. In: International Semantic Web Conference. pp. 309–324. Springer (2019)
6. Tzitzikas, Y., Manolis, N., Papadakos, P.: Faceted exploration of RDF/S datasets: a survey. Journal of Intelligent Information Systems **48**(2), 329–364 (2017)
7. Tzitzikas, Y., Marketakis, Y., Minadakis, N., Mountantonakis, M., Candela, L., Mangiacrapa, F., Pagano, P., Perciante, C., Castelli, D., Taconet, M., et al.: Methods and tools for supporting the integration of stocks and fisheries. In: Chapter in Information and Communication Technologies in Modern Agricultural Development, Springer, 2019. Springer (2019)