



Article

Keyword Search over RDF: Is a Single Perspective Enough?

Christos Nikas ^{1,2}, Giorgos Kadilierakis ^{1,2}, Pavlos Fafalios ¹ and Yannis Tzitzikas ^{1,2,*}

¹ Information Systems Laboratory, FORTH-ICS, 70013 Heraklion, Greece; cnikas@ics.forth.gr (C.N.); kadilier@csd.uoc.gr (G.K.); fafalios@ics.forth.gr (P.F.)

² Computer Science Department, University of Crete, 70013 Heraklion, Greece

* Correspondence: tzitzik@ics.forth.gr

Received: 6 August 2020; Accepted: 24 August 2020; Published: 27 August 2020



Abstract: Since the task of accessing RDF datasets through structured query languages like SPARQL is rather demanding for ordinary users, there are various approaches that attempt to exploit the simpler and widely used *keyword-based search paradigm*. However this task is challenging since there is no clear unit of retrieval and presentation, the user information needs are in most cases not clearly formulated, the underlying RDF datasets are in most cases incomplete, and there is not a single presentation method appropriate for all kinds of information needs. As a means to alleviate these problems, in this paper we investigate an interaction approach that offers multiple presentation methods of the search results (*multiple-perspectives*), allowing the user to easily switch between these perspectives and thus exploit the added value that each such perspective offers. We focus on a set of *fundamental perspectives*, we discuss the benefits from each one, we compare this approach with related existing systems and report the results of a task-based evaluation with users. Their key finding of the task-based evaluation is that users not familiar with RDF (a) managed to complete the information-seeking tasks (with performance very close to that of the experienced users), and (b) they rated positively the approach.

Keywords: keyword search; RDF; interactive information retrieval

1. Introduction

The Web of Data contains thousands of RDF datasets available online (see [1] for a recent survey), including cross-domain knowledge bases (KBs) (e.g., DBpedia and Wikidata), domain specific repositories (e.g., DrugBank [2], ORKG [3], and recently COVID-19 related datasets [4]), as well as Markup data through `schema.org`. These datasets are queried mainly through structured query languages, i.e., SPARQL. Faceted Search is a user-friendlier paradigm for interactive query formulation and exploratory search, however, the systems that support it (see [5] for a survey) also need a keyword search engine as a flexible 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. Moreover, keyword search allows for multiple-word (even paragraph-long) queries that can address many topics, and such information needs could be difficult to formulate even in structured query languages. The results of such queries allow users to detect associations of entities that they were not aware of, thus favoring the discovery of new information.

In general, we could say that *structured queries* (e.g., using SPARQL) and *unstructured queries* (keyword search) are fundamental components of all access methods over RDF. Figure 1 shows the general picture of access services over RDF. Apart from *Structured Query Languages* and *Keyword Search* we can see the category *Interactive Information Access*. That refers to access methods that are beyond the simple “query-and-response” interaction, i.e., methods that offer more interaction

options to the user and exploit also the *interaction session*. In this category, we have methods for browsing, methods for faceted search [5], methods for formulating OLAP queries (e.g., [6]), and methods for assistive query building (e.g., [7]). Finally, in the category *natural language interfaces* we have methods for question answering, dialogue systems, and conversational interfaces. As the figure shows, both *interactive information access* and *natural language interfaces* pre-suppose effective and efficient support of structured and unstructured queries.

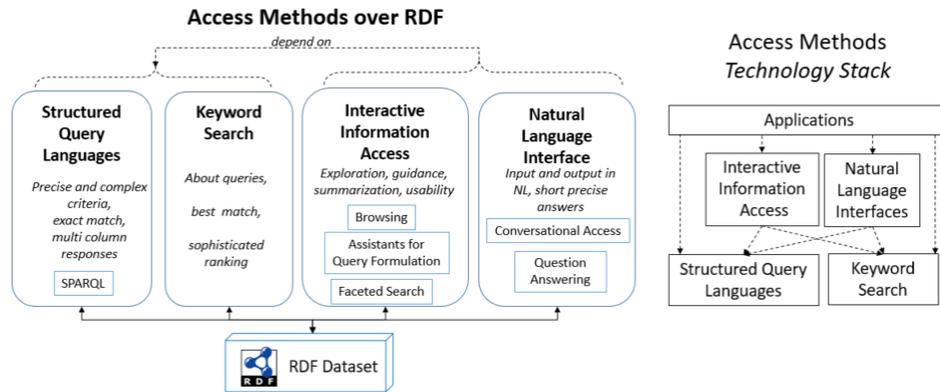


Figure 1. Access Methods over RDF.

However, keyword search over RDF datasets is a challenging task since (a) in RDF there is no clear unit of retrieval and presentation, (b) it is difficult to understand, from a usually small keyword query, the intent of the user, (c) the data are in most cases incomplete (making the provision of effective retrieval difficult), and (d) there is not a single presentation method appropriate for all kinds of information needs.

To tackle these challenges, in this paper we focus on the value stemming from offering *multiple-perspectives* of the search results, i.e., multiple presentation methods, each presented as a separate *tab*, and allowing the user to easily *switch* between these perspectives, and thus exploit the added value that each such perspective offers. To grasp the idea, Figure 2 shows the search results for the query “El Greco museum”, as presented in each of the five currently supported tabs.

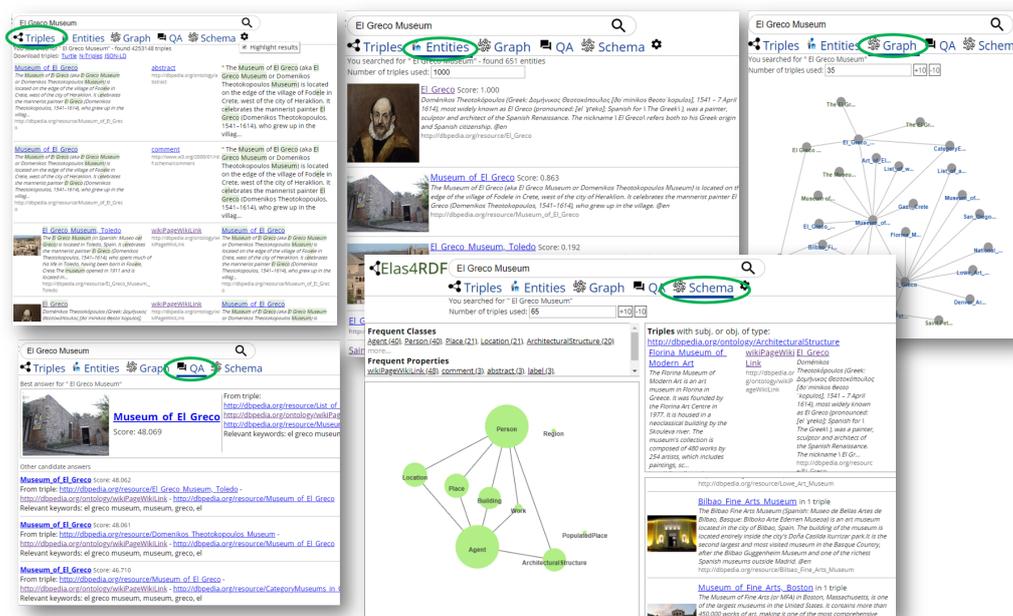


Figure 2. Search results for the query “El Greco museum”.

As basic keyword-search retrieval method, we assume the *triple-centered* approach proposed in [8] (which in turn relies on Elasticsearch) because it is schema-agnostic (and thus general-purpose), and it offers efficient and scalable retrieval services with effectiveness comparable (as evaluated using DBpedia-Entity test collection for entity search [9]) to the effectiveness of dedicated systems for keyword search over RDF (more in [8]). Over this basic service, in this paper we motivate the provision of certain *fundamental perspectives*, we showcase the benefits from each one, and we evaluate what users can achieve if they have all of them at their disposal.

In comparison to previous works (the demo paper [10]), in this paper: we motivate the multi-perspective approach, we discuss the added value of each perspective, we introduce additional perspectives, we compare the functionality of the implemented system with other systems over DBpedia, and mainly we report the results of a *task-based evaluation with users* that provides interesting insights related to the validation of the main research hypothesis of this paper, i.e., whether the provision of more than one tab is helpful for the users. The key finding is that the success rate of all users was very high even of those users not familiar with RDF.

The rest of this paper is organized as follows: Section 2 discusses the related work, Section 3 provides the motivation for the multi-perspective approach and describes its architecture, while Section 4 describes each individual perspective and the tab-switching interaction approach. Section 5 compares the proposed approach with related (and comparable) systems and presents the results of a task-based evaluation with users. Finally, Section 6 concludes the paper and identifies issues for further research.

2. Related Work

At first we provide some background about RDF (in Section 2.1), then we discuss the existing approaches for keyword search over RDF (in Section 2.2), and finally, we discuss the visualization of RDF search results (in Section 2.3).

2.1. Background: RDF

RDF stands for *Resource Description Framework* and it is a framework for describing resources on the web. Essentially it is a structurally object-oriented model. RDF uses Uniform Resource Identifiers (URIs), or anonymous nodes, to denote resources, and literals to denote constants. Every statement in RDF can be represented as a *triple*. A triple is a statement of the form subject-predicate-object $\langle s, p, o \rangle$, and it is any element of $T = (U \cup B) \times (U) \times (U \cup B \cup L)$, where U , B and L are the sets of URIs, blank nodes and literals, respectively. Any finite subset of T corresponds to an RDF graph (or dataset). We can divide the URIs in three disjoint sets: entities (e.g., <http://dbpedia.org/resource/Aristotle>), properties (e.g., <http://dbpedia.org/property/dateOfBirth>) and RDF classes (e.g., <http://dbpedia.org/ontology/Philosopher>).

2.2. Keyword Search over RDF Datasets

Keyword search over RDF data can be supported either by translating keyword queries to structured (SPARQL) queries (like in [11,12]), or by building or adapting a dedicated information retrieval system using classical IR methods for indexing and retrieval. This paper builds upon approaches that follow the second direction. In general, systems of that kind construct the required indexing structures either from scratch or by employing existing IR engines (like Lucene and Solr), adapt the notion of a virtual document for the RDF data, and rank the results (entities, triples or subgraphs) according to commonly used IR ranking functions. There are various systems that fall in this category, like [13–15]. Most such systems rely on adaptations of the TD-IDF weighting, as in [16] where the keyword query is translated to a logical expression that returns the ids of the matching entities. Another direction is to return ranked subgraphs instead of relevant entity URIs, like in [17], while in [18] the returned subgraphs are computed using statistical language models.

Ranking is usually based on extensions of the BM25 model, e.g., in [19,20]. The work in [21] introduced the TSA+VDP keyword search system, which first builds offline an index of documents over a set of subgraphs via a breadth-first search method, and at query-time, it returns a ranked list of these documents based on a BM25 model. Regarding the retrieval unit, most works return either URIs or subgraphs, except [8,22] that follow a triple-centered approach.

With respect to works that rely on document-centric information retrieval systems, LOTUS [22] makes use of Elasticsearch and provides a keyword-search entry point to the Linked Data cloud, focusing on issues of scalability. Elasticsearch has been also used for indexing and querying Linked Bibliographic Data in JSON-LD format [23]. Finally, Kadilierakis et al. [8] adapts Elasticsearch for supporting keyword search over arbitrary RDF datasets. Through an extensive evaluation, the authors studied questions related to the selection of the triple data to index, the weighting of the indexed fields, and the structuring and ranking of the retrieved results. In our work, we make use of the approach proposed in [8] because it is schema-agnostic and returns ranked lists of triples, which offers us the flexibility to provide different visualizations of the search results.

2.3. Visualization of RDF Search Results

There are several approaches for browsing, exploring and visualizing RDF datasets in general, e.g., see the surveys [24,25]. Regarding the visualization of SPARQL results, there are a few works, however, since the form of the results of such queries is essentially that of a relational table, these approaches provide amenities for the visualization of tabular data, i.e., various plots and charts for analytics [26–28].

As regards the visualization of keyword search results over RDF, which is the main focus of our work, DBpedia Precision Search & Find (<http://dbpedia.org/fct/>) returns entities and for each one it shows its URI, its title, the URI of the named graph it belongs to, as well as a description with highlighted the query terms. Also, the user can browse on the Linked Data by clicking on the shown resources. The keyword search systems LOTUS [8,22,29] do not focus on presentation and visualization. LOTUS returns triples by providing the full URIs of the resources, while [8] returns triples and/or entities using an API. In general, most works (including [30,31]) do not pay attention to the presentation of results; they focus on the ranking of entities/subgraphs that they compute.

Finally, Stab et al. [32] and Kontiza et al. [33] the exploitation of semantics in the visualization of search results. The work in [32] uses visualization techniques for offering visual feedback about the reasons a set of search results was retrieved and ranked as relevant. In [33] the authors performed an analytical inspection and a user study of the interface offered by two semantic search engines: *Kngine* and *Sig.ma* (both are not active anymore). In particular, the authors investigated if the exploitation of semantics enables a better visualization of search results and thus a better user experience.

To our knowledge, our work is the first that investigates and evaluates (with real users) a multi-perspective interactive approach to present the search results of a keyword search system over RDF.

3. Multi-Perspective Presentation of Search Results: Rationale and Architecture

3.1. Rationale

The rationale for the multi-perspective (and tabs-switching interaction) approach that we propose can be summarized as:

- *No Clear Unit of Retrieval and Presentation.* In RDF data, there is not the notion of document or web page as is the case in web searching. Therefore, the retrieval, presentation and visualization of RDF data is challenging due to the complex, interlinked, and multi-dimensional nature of this data type [25].

- *No Clear Information Need.* The user query is just an attempt to formulate his/her information need. Some user needs require a single fact, others a list of entities or a set of facts, other how a set of entities are connected, other have an exploratory nature, and so on.
- *Incomplete Data.* The underlying dataset is in most cases incomplete [34] (also demonstrated by the number of papers that aim at completing the missing data [35]), therefore the retrieved triples cannot be considered neither complete, nor appropriately ranked. However, the provision of more than one method, each consuming different proportions of the list of top hits (and of their context), increases the probability that one method achieves to return something that is useful for the user's information need.
- *There is not a single presentation method appropriate for all kinds of information needs.* An established method on how to present RDF results for arbitrary query types does not exist yet, and it seems that a single approach cannot suit all possible requirements. Different kinds of information needs need different ways to present the results.

For the above reason we propose a *multi-perspective* approach, where each perspective is presented in a different *tab*, stressing a different aspect (and proportion) of the hits. The user can inspect all tabs and get a better overview and understanding of the search results. The *tabs-switching interaction* that we propose is easy to understand and perform by the user, just like plain Web search engines offer various such tabs (for images, videos, news, etc.). Below, in Section 4, we shall discuss the rationale (added value) of each particular tab and how it is defined. An orthogonal but important challenge is how to provide several such presentation methods at real time, for enabling the user to switch fast between the different perspectives, i.e., the multi-perspective and tab-switching approach should not add a noticeable latency to the responses.

3.2. Architecture

As keyword search service we adopt the approach proposed in [8] because it is schema-agnostic, directly applicable, has good evaluation results, and its triple-centered approach facilitates the multi-perspective approach. Specifically, we exploit the REST API that is offered by that service which accepts keyword queries and returns results in JSON format (code available at <https://github.com/SemanticAccessAndRetrieval/Elas4RDF-search>). On top of this search service we build the multi-perspective approach.

The full DBpedia 2015-10 dataset has been indexed using 2 approaches (i.e., *baseline* and *extended*, described in [8]). We have used that version of DBpedia because it is the version used in the DBpedia-Entity test collection for entity search [9], which allowed us to get comparable results related the effectiveness of the approach (as detailed in [8]). The number of virtual documents (triples) in both cases is 395,569,688. In our setup and experiments, the average query execution time is around 0.7 s for the baseline method and 1.6 s for the extended, and depends on the query type.

4. The Fundamental Perspectives of Keywords Search Results

Below we describe each individual perspective (for short *tab*) and then (in Section 4.6) we discuss the role of each in tab in the general search process. In the description of each perspective we consider the DBpedia 2015-10 dataset and the query $q_{run} = \text{"El Greco museum"}$ as our running example.

4.1. Triples Tab

Rationale: 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 the user to easily understand which triples involve the same entities.

Description: A ranked list of triples is displayed to the user (as fetched from the search service described in Section 3.2), where each triple is shown in a separate row. For visualizing 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 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 this resource (if any). For the query $q_{run} = \text{"El Greco museum"}$, more than 4.2 million triples are retrieved. The first two triples are about the Museum of El Greco in Crete, the third about the El Greco Museum in Toledo, the fourth about the entity El Greco, the fifth is a triple about a list of works by El Greco, and so on.

4.2. Entities Tab

Rationale: If the user is interested in entities, and not in particular facts, this view provides the main entities.

Description: Here the retrieved triples are *grouped* based on entities (subject and object URIs), and the entities are *ranked* following the approach described in [8] 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 visualizing 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. For q_{run} the returned entities include "El Greco", the two museums of El Greco (in Crete and Toledo), particular paintings, like "Saint Peter and Saint Paul", the music album "El Greco" by Vangelis, the film "El Greco (2007)", and so on.

4.3. Graph Tab

Rationale: This tab allows the user to inspect a larger number of triples without having to scroll down. Most importantly, this view reveals the grouping of triples, how they are connected, and whether there is one or more poles and interesting connections.

Description: The retrieved triples are visualized as a graph for stressing how the triples are connected. By default, the graph shows the top-15 triples; however, the user can increase or decrease this number, while the nodes are clickable, pointing to the corresponding resource in DBpedia. In our implementation we use JavaScript InfoVis Toolkit (<https://philogb.github.io/jit/>). For q_{run} the user can see how the top ranked triples are connected and can easily spot the nodes that have high connectivity.

4.4. Schema Tab

Rationale: The objective is to show which are the more frequent schema elements of the retrieved triples. This is useful for (a) understanding the conceptual context of the hits, (b) for exploring (restricting) interactively the triples or entities of the answer (by filtering with respect to class or property), and (c) for helping an experienced user to inspect which classes and properties occur in the answer, if after the keyword search, the user would like to formulate a SPARQL query (directly or through a faceted search system, or through a query builder in general like [7,36]).

Description: The schema tab is divided in four frames as shown in Figure 3.

Upper Left Frame: It shows the more frequent classes and properties, accompanied by their frequency. Let A be the top- K triples retrieved for the current query, P the properties in A , i.e., $P = \{p \mid (s, p, u) \in A\}$, and C the classes of the URIs in the triples of A , i.e., $C = \{c \mid (s, rdf : type, c), s \in SP\}$. For each $c \in C$, its frequency is defined as $freq(c) = |\{o \in SP \mid (o, rdf : type, c) \in KB\}|$, while for each $p \in P$, $freq(p) = |\{(s, p, o) \in A\}|$. Through a parameter F we control the number of visible elements, i.e., initially the user can see only the F in number elements of C with the highest frequency, and the F in number elements of P with the highest frequency (however, the user can expand the visible elements to see all of them). By clicking a class or a property the user can see the corresponding triples and entities in the frames at the right side that will be described later.

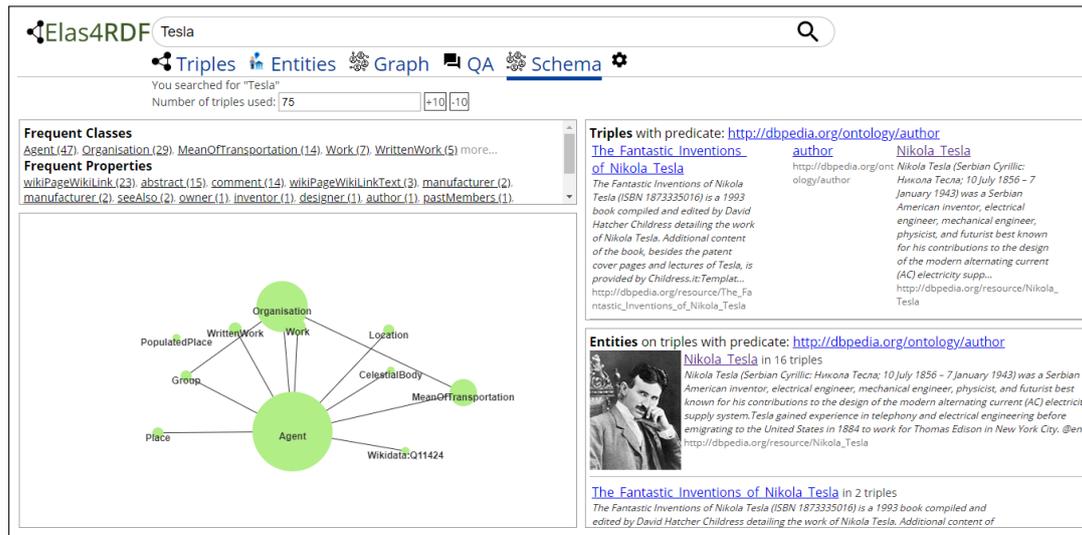


Figure 3. The Schema Tab (Tesla).

Bottom Left Frame: It shows graphically the more frequent classes and properties. A parameter K (just like in the graph tab) controls the number of triples that feed the schema tab (the user can increase decrease it as she wishes to). In particular, the graph $\Gamma = (Nodes, Edges)$ that is visualized is defined as $Nodes = C$, and $Edges = \{(c, c') \in C \times C' \mid (s, p, o), (s, rdf : type, c), (o, rdf : type, c') \in A\}$, i.e., an edge connects two classes c and c' if there is at least one triple in A that connects an instance of c with one instance of c' . Ideally the graph visualization should make evident the frequencies, i.e., the more frequent classes and properties should be visualized with bigger boxes and arrows. It is not hard to see that the number of edges, i.e., $|E|$, can be higher than the number of distinct properties that occur in A , e.g., if $(s, p, o) \in A$ and s is classified to two classes $c1$ and $c2$, and o to two classes $c3$ and $c4$, then the graph will contain the four edges $\{(n(c1), n(c3)), (n(c1), n(c4)), (n(c2), n(c3)), (n(c2), n(c4))\}$. The reverse is also possible, i.e., $|E|$ can be less than the number of distinct properties, e.g., if $(s, p1, o)$ and $(s, p2, o)$ belong to A , and each of s and o is classified to one class, then only one edge will be visible between these two classes. Please note that several variations and extensions are possible from the area of semantic model visualization and summarization.

Right Upper and Right Bottom Frames: These frames show the *triples* and *entities*, related with the user's click. Suppose the user has clicked on a frequent class "c1(18)". The triples frame will show all triples $\{(s, p, o) \in A \mid (s, rdf : type, c1) \wedge (o, rdf : type, c1)\}$, and let call this set T . The entities frame will show the more frequent entities that occur in T . If the user clicks on a frequent property "p2(10)", the triple frame will show the 10 triples A that have $p2$ as property, let call this set T , and the entity frame will show the more frequent entities of those occurring in T . The above behavior is supported also by the graph, i.e., clicking on a node is interpreted as if the user had clicked on the corresponding frequent class.

Returning to q_{run} , we can see the classes Person, Agent, Location, Work, etc. and various properties. The right frames show the triples and entities after having clicked on "Architectural Structure", i.e. triples and entities that are related to the query *and* classified under the class "Architectural Structure" (we can see information about a museum in Florina, another in Bilbao, etc.).

As another example, for the query "Tesla", the user is getting what is shown in Figure 3, enabling him to focus on the desired triples or entities, i.e., to those related to: Tesla Motors (Organization), Nicola Tesla (Agent), Tesla Model X (Mean of Transportation), Tesla West Virginia (Place). By increasing the number of triples he can also find Tesla Band (Group). By clicking on the property "author" the

user can directly see the triple related to works authored by Nicola Tesla. In general, in this tab the user can increase a lot the number of consumed triples: although more classes and properties will appear their number is not high, hence in most cases they will not clutter the diagram (in the example of Figure 3 the schema tab consumes 75 triples).

4.5. Question Answering (QA) Tab

Rationale: Here we attempt to interpret the user's query as a *question* and try to provide a *single compact answer*. The challenge is to retrieve the most relevant triple(s) and then extract natural language answers from them.

Description: QA over structured data is a challenging problem in general (e.g., see [37] for a recent survey), and any QA over KB approach could be applied in this tab. In our current implementation, we only support questions that can be answered by a single triple. We extract a set of terms from the question by applying lemmatization and expansion to multi-word expressions. Then we attempt to retrieve triples where two components (subject, predicate, or object) are similar to terms extracted from the question. To do that, we use Elasticsearch's query Domain Specific Language to search for combinations of terms in the positions of subject, predicate, or object. For example, for the question "Who developed Skype?" we find the answer "Microsoft" from the triple: <http://dbpedia.org/resource/Skype>–<http://dbpedia.org/ontology/developer>–<http://dbpedia.org/resource/Microsoft>. The system returns the more probable answer accompanied by a score, plus a list of other possible answers. In our running example, this tab returns the Museum of El Greco (in Crete).

4.6. Tabs' Roles and Extra Tabs

There are several other tabs that could be supported and could be useful in certain kinds of information needs, e.g., *image* tab, *geo* tab, *time* tab, etc. Each can be construed as a tool that could aid the user to focus on a particular aspect, based on the task/information need at hand, each enacted by a simple click (therefore the required effort is minimal). One rising question is how to provide an *overview* of these in an effortless manner, and/or how to rank them if that is desired. For reasons of transparency and exploration, it is beneficial to make the user aware of the existence of these, instead of promoting and showing only one, as some Web Search Engines (WSE) do. However, we should mention that it is the task of QA to identify the *question type* and the *expected answer type*, therefore, based on the analysis of the QA perspective, a short answer (presented in the appropriate way), could be promoted (just like WSE do), therefore, one direction for further research is to investigate the applicability of approaches like [38,39] for complex questions.

In this current paper we confine ourselves on the previous five tabs since we believe that they are both *KB-independent* and *task-independent*, hence they can be considered to be fundamental. The added value from each of these basic perspectives is summarized in Figure 4. The diagram also shows some main paths that indicate why a user may decide, in a tab-switching interaction, to move from a tab to another (of course, the user is free to follow any order). Below we provide a few additional examples showcasing the benefits from using more than one tab.

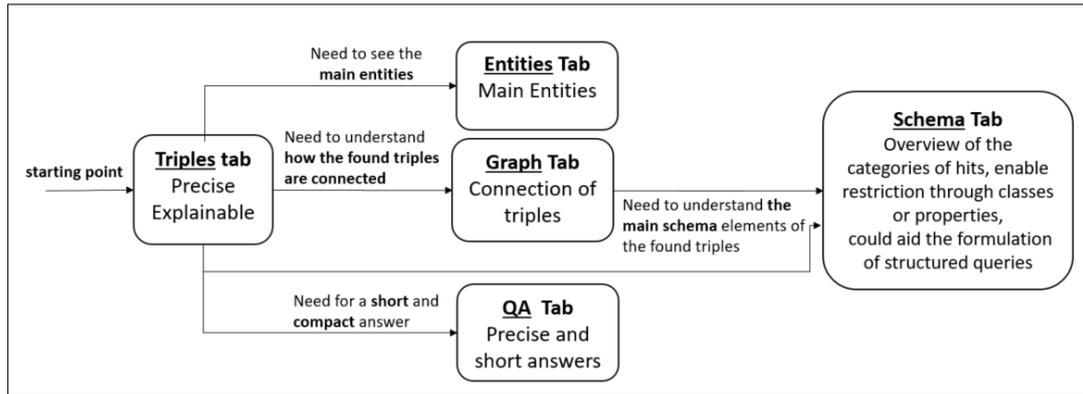


Figure 4. The Added Value of each Perspective.

For the query q = “El Greco and Kazantzakis” in the Entities Tab, as shown in Figure 5, the user can find in the first two positions the two main entities of the query, i.e., “El Greco” (the painter), and “Nikos Kazantzakis” (the writer and philosopher), while in the Triples Tab the user can find a triple that connects these two entities. From the Graph Tab the user can see the triples grouped in two poles (one for each entity) and the user can realize that there is only one triple that connect these two poles (in the top-35 triples). Finally, with the Schema Tab the user can refine to Location and find entities whose name is related to the main entities, like “El Greco Apartments” and “Nikos Kazantzakis (municipality)”.

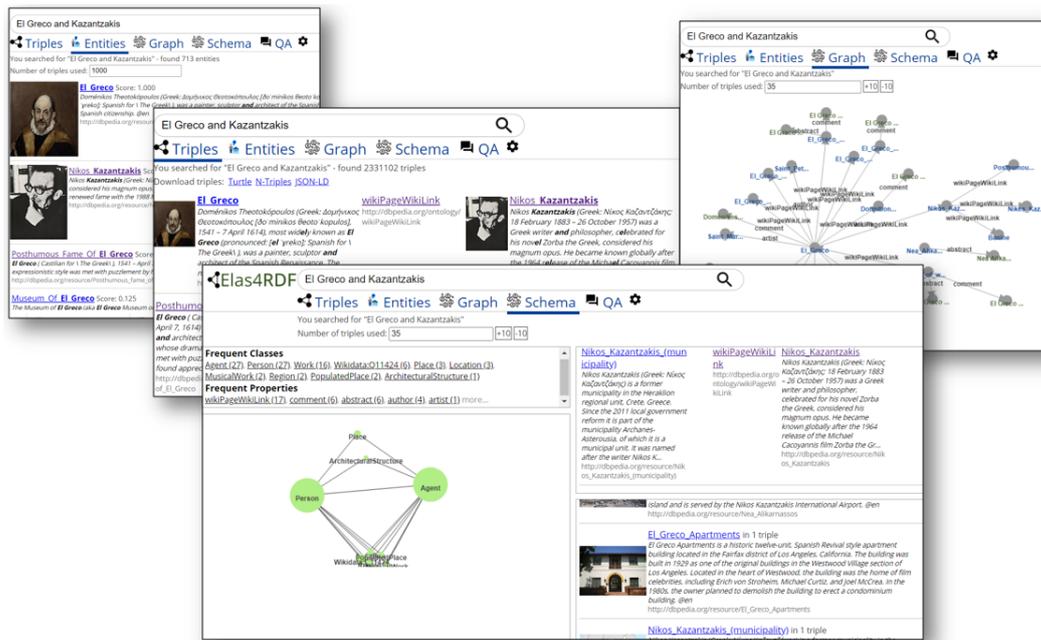


Figure 5. Search results for the query “El Greco and Kazantzakis”.

As another example, for the query “Paintings with dogs” in the Triples Tab, as shown in Figure 6, the user can find relevant specific information including information about “Painted Dog Conservation” (a non-profit organization for the protection of the painted dog, or African wild dog), information about particular paintings, information about “Greg Rasmussen” the founder of the “Painted Dog Conservation”, etc. In the Entities tab the user can find the main entities, including the “Painted Dog Conservation”, the species “African Wild Dog”, one painting of Goya (The Dog), the “Dogs Playing Poker” (the series of 16 oil paintings by C. M. Coolidge), etc. The Schema Tab shows the classes and properties of the found triples, through which the user can understand that there are related: species, (art) works, locations, etc. Moreover, the user can refine/explore the

information space as she wishes to. In Figure 6 the user has refined using the class “Work” and in the right bottom frame he can find various paintings with dogs including: “The Dog (Goya)”, “The Sentry (painting)”, “The Hunt In The Forest”, “Interior With A Young Couple And A Dog” “Portrait Of Charles V With A Dog” etc. Finally, the QA Tab returns two entities “Francisco Goya” (the painter of the painting “The Dog”), and “Coenraad Jacob Temminck” (a Dutch aristocrat, zoologist, and museum director who first described scientifically in 1820 the species African Wild Dog).

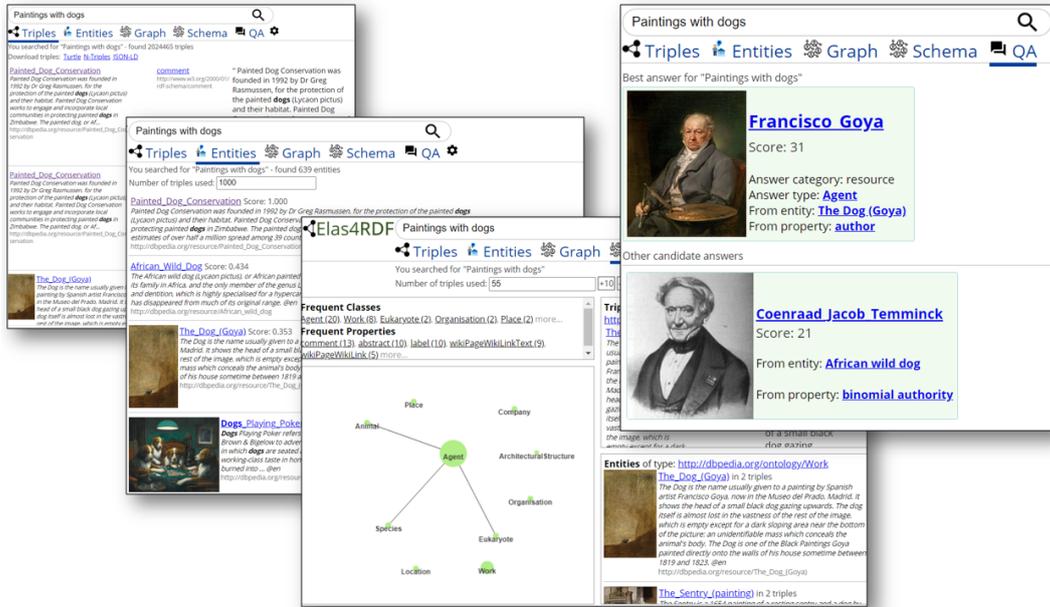


Figure 6. Search results for the query “Paintings with dogs”.

For list questions, i.e., questions with a set of elements as the correct response, like “Which cities does the Weser flow through?” the user may decide to inspect only the QA Tab and the Entities Tab as shown in Figure 7.

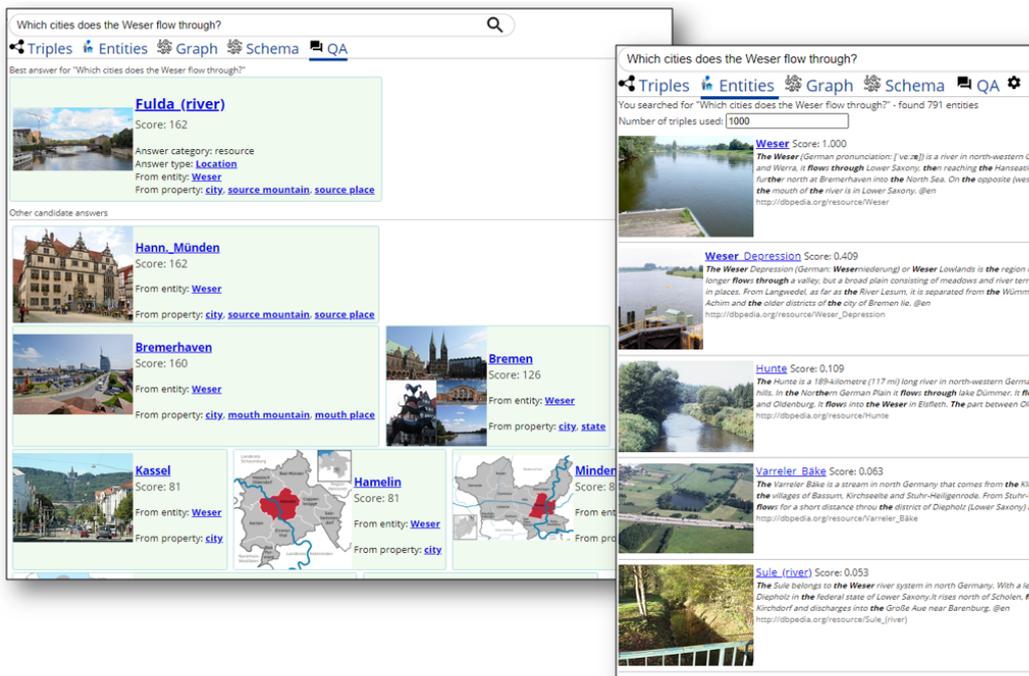


Figure 7. Search results for the query “Which cities does the Weser flow through?”.

Longer queries are also possible, for instance for the query “Greek philosopher from Athens who is credited as one of the founders of Western philosophy”, from the Entity Tab (as shown in Figure 8) the user we can see that Socrates received the higher score, while from the QA tab the user can see various other philosophers as candidate answers.



Figure 8. Search results for the query “Greek philosopher from Athens who is credited as one of the founders of Western philosophy”.

5. Evaluation

Below we evaluate the proposed approach by (a) comparing its *functionality* with those of related systems, (b) proving its feasibility by discussing *efficiency*, (c) discussing the retrieval *effectiveness* of the system, and (d) reporting the results of a *task-based evaluation with users* that examines the usefulness of the proposed multi-perspective approach, as well as some results by *log analysis*.

5.1. Comparing the Functionality with Related Systems

Since DBpedia is a core dataset of the Linked Open Data cloud [40], we decided to compare with *interactive systems* (not just APIs) that offer a kind of access/search facility over DBpedia. For this reason, we considered the following systems: LOTUS [22], GraFa [41] (<http://grafa.dcc.uchile.cl/>), RelFinder [42] (<http://www.visualdataweb.org/relfinder.php>), DBpedia Search & Find (<http://dbpedia.org/fct/>), SPARKLIS [43] (<http://www.irisa.fr/LIS/ferre/sparklis/>), and our system Elas4RDF (<https://demos.isl.ics.forth.gr/elas4rdf/>).

The results are summarized in Table 1. The table has a column for each of the following features: *triple search*, *entity search*, *graph-view*, *faceted search*, *QA*, *relation finder*, *SPARQL query support*. The last column sums up the number of features each system supports: we count each supported feature with 1, and each partially supported feature with 0.5, as an indicator of the spectrum of the provided access services. We can see that most systems focus on only one or two access methods, while our system offers four, hence it provides a wider spectrum of access services.

Table 1. Search Systems over DBpedia.

| System | Triple Retrieval | Entity Search | Graph View | Faceted Search | QA | Relation Finder | SPARQL Support | SUM |
|-----------------------------|------------------|-----------------------------------|------------------------------------|-----------------------|-----|-----------------|-------------------------|-----|
| LOTUS [22] (no online demo) | Yes | No | No | No | No | No | No | 1/7 |
| GraFa [41] | No | No | No | Yes | No | No | No | 1/7 |
| RelFinder [42] | No | Partial (through auto completion) | Partial (only of related entities) | No | No | Yes | No | 1/7 |
| DBpedia Search & Find | Yes (no images) | No | No | Partial (simple) | No | No | Partial (query display) | 2/7 |
| SPARKLIS [43] | No | No | No | Yes (Very Expressive) | No | No | Yes | 2/7 |
| Elas4RDF | Yes | Yes | Yes | No | Yes | No | No | 4/7 |

5.2. Efficiency

The efficiency of the back-end search service (i.e., of the ranking service) was evaluated in [8]. Here we focus on the cost for providing the multiple perspectives of the search results. The key point is that the implementation of the perspectives on top of the search service, described in Section 3.2, does not add significant overhead, preserving the real-time interaction. Furthermore, the triples and entities retrieved from the search service are *cached*, further improving load times when the same query is issued on different perspectives.

In Table 2, the average load time of each perspective is displayed (with and without caching), considering 10 queries of varying length from 1 to 8 words and using an instance of the system that runs on a machine with 6 physical cores and maximum memory allocation size set to 8GB. We can see that even without caching all responses are returned in less than 3 seconds, while with caching enabled, the average time is around 150 ms.

Table 2. Average load times for each perspective.

| Perspective | Triples | Entities | Graph | Schema | QA |
|-----------------|---------|----------|---------|--------|---------|
| Without caching | 980 ms | 2582 ms | 1018 ms | 924 ms | 2869 ms |
| With caching | 145 ms | 124 ms | 91 ms | 175 ms | 118 ms |

5.3. Evaluation of Effectiveness

Another evaluation aspect is the effectiveness of the system, i.e., its capability to fulfill the information needs of the user. Note here that since one can use his own retrieval, ranking or visualization method in any of the fundamental perspectives, evaluating the performance of the method used in each different tab is out of the scope of this paper. As regards the implementation of the tabs in our prototype (described in Section 4), the ranking of the entities in the *entities* tab has been extensively evaluated in [8], demonstrating a high performance. This provides a very positive evidence about the quality of the triples that feed all tabs, in the sense that if triple-ranking were not effective, then it would be hard for the *entities* tab to be effective. More importantly, the results of the user study (that we shall see in Section 5.4) validate the good quality of the results shown in each tab. Specifically, the large majority of users managed to find correct answers for most of the requested tasks. That would be impossible if most of the results in the tabs were irrelevant (more about the user study below in Section 5.4).

5.4. Evaluation with Users

Since there is no dataset that could be used for evaluating the particular multi-perspective interaction we decided to carry out a task-based evaluation with users. Specifically, we wanted to understand how users would use such a system, whether they find useful and/or like the multi-perspective approach, and for collecting general and specific feedback.

5.4.1. Information-Seeking Tasks

Since we are in keyword-search setting (and not in a structured query building process), we selected several tasks that have IR nature, and at the same time are not trivial (some of them are hard to answer, and/or DBpedia has related but not exactly the requested information). We also tried to capture various kinds of information needs, while keeping the list of tasks short for attracting more participants. The selected 11 tasks are shown in Table 3. They include queries of various kinds (entity property queries, entity relation queries, fact checking queries, entity list queries). In total, answering these questions requires at least 30 min.

Table 3. Evaluation Tasks.

| ID | Task |
|-----|---|
| T1 | Is there any person that is fisherman, writer and poet? Provide at least 3 related names (or URIs). |
| T2 | Is there any writer and astronaut from Russia? Provide related names or URIs. |
| T3 | Find information that relates Albert Einstein with Stephen Hawking. |
| T4 | Find if El Greco was influenced by Michelangelo. |
| T5 | Is there any reference of Freud to the ancient Greece? |
| T6 | How is Mars related to Crete? |
| T7 | Find mathematicians related to Pisa. |
| T8 | Find painters of the Ancient Greece. |
| T9 | Are there drugs that contain aloe? |
| T10 | Which cities does the Weser flow through? |
| T11 | Find at least 5 rivers of Greece. |

5.4.2. Participants, Questionnaire and Results

We invited by email various persons to participate in the evaluation voluntarily. The users were asked to carry out the tasks and to fill (anonymously) the prepared questionnaire. No training material was given to them, and the participation to this evaluation was optional (invitation by email). Eventually, 25 persons participated (from 5 May 2020 to 18 May 2020). The number was sufficient for our purposes since, according to [44], 20 evaluators are enough for getting more than 95% of the usability problems of a user interface. In numbers, the participants were 32% female and 68% male, with ages ranging from 20 to 54 years; the distribution is almost uniform, only the age of 23 is the more frequent 20%, as shown in Figure 9.

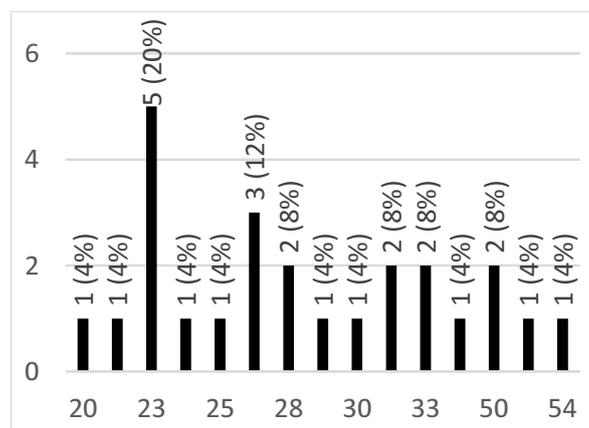


Figure 9. Age distribution of participants.

As regards occupation and skills, all have studied Computer Science, except one Physicist. In detail, 20% were undergraduate students, 15% of them postgraduate computer science students, and the rest computer engineers, professionals and researchers. Students came from at least 3 different

universities, while 40% of all the participants have never used DBpedia before. The questionnaire is shown below, enriched with the results of the survey in the form of percentages written in bold:

- E1 How would you rate the Triples tab?: Very Useful (**40%**), Useful (**44%**), Little Useful (**16%**), Not Useful (**0%**)
- E2 How would you rate the Entities tab?: Very Useful (**44%**), Useful (**28%**), Little Useful (**24%**), Not Useful (**4%**)
- E3 How would you rate the Graph tab?: Very Useful (**32%**), Useful (**52%**), Little Useful (**12%**), Not Useful (**4%**)
- E4 How would you rate the Schema tab?: Very Useful (**16%**), Useful (**40%**), Little Useful (**36%**), Not Useful (**8%**)
- E5 How would you rate the QA tab?: Very Useful (**16%**), Useful (**36%**), Little Useful (**40%**), Not Useful (**8%**)
- E6 Did you find it useful that the system offers multiple perspective of the search results?: Very much (**48%**), Fair (**48%**), Not that Useful (**4%**), Not Useful (**0%**)
- E7 Mark the perspective(s) that you think are redundant: Triples Tab (**0%**), Entities Tab (**8%**), Graph Tab (**8%**), Schema Tab (**40%**) QA Tab (**16%**) All tabs are useful, none is redundant (**44%**)
- E8 Have you used DBpedia before: Never (**40%**), Only a few times (without using SPARQL) (**16%**), Quite a lot (I have used SPARQL to query it) (**44%**).
- E9 How would you rate the entire system? Very Useful (**32%**), Useful (**60%**), Little Useful (**8%**), Not Useful (**0%**)
- E10 You can report here errors, problems, or recommendations. (free text of unlimited length)

5.4.3. Results Analysis and Discussion

User Ratings. As regards *ratings*, most users appreciated the multi-perspective approach (the positive options of E6, Very Much and Fair, sum to 96%). Moreover, all tabs received positive results by some users. By adding the percentages of Very Useful and Useful, the ranked list of *more preferred* tabs is:

{GraphTab (84%), TriplesTab (84%)}, EntitiesTab (72%), SchemaTab (56%), QATab(52%) }.

The *less preferred* tabs, according to the sum of Little Useful and Not Useful percentages, is:

{ QATab (48%), SchemaTab (44%), EntitiesTab (28%), {GraphTab (16%), TriplesTab (16%)} }.

Please note that these numbers correspond to the percentages of users that *would not be satisfied* if only the corresponding perspective were provided to them.

It is also clear that different users have different preferences for perspectives: there are persons that rated the Schema Tab as Very Useful, while others marked it as Redundant. Probably this depends on the background of the participants: a person with no knowledge of RDF would not be able to understand (and exploit) the notion of schema, and we have seen that 20% of the participants were undergraduate and 40% have never used DBpedia. This is also evident from Figure 10 that depicts the sum of Very Useful and Useful percentages per tab; the black bars correspond to the users that had never used DBpedia, while the white bars correspond to the users that had used DBpedia before.

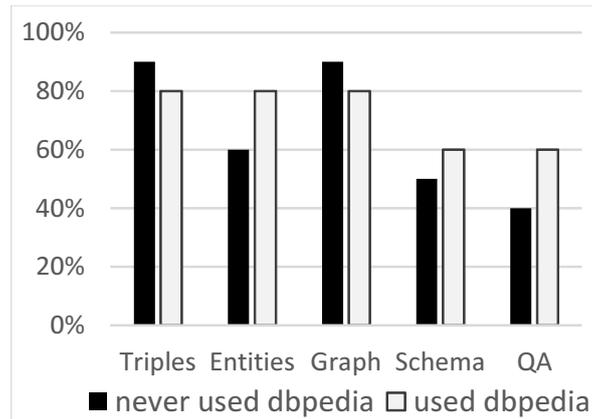


Figure 10. ‘Very Useful’ and ‘Useful’ preference percentages per tab and category of users.

By looking at the responses of the questionnaire, we can see that the group of users that had never used DBpedia, preferred the Triples Tab and the Graph Tab (40% found them Very Useful, 50% Useful, and 10% Little Useful, for both tabs), and the least useful tab for them was the Schema Tab (10% Very Useful, 40% Useful, and 50% Little Useful), because a basic understanding of the RDF data model is required to use it. Regarding this user group’s opinion of the multi-perspective approach, 30% found it to be Very Useful, and 60% found it Fair. Only one user did not find the approach useful. Also, 50% of these users responded that None of the perspectives are redundant.

Statistical Significance. As regards *statistical significance*, by assuming as *positive* the options Very Useful and Useful, and as *negative* the options Little Useful and Not Useful, the lower bound of *Wilson score confidence interval* shows that with 95% confidence, the percentage of users (of the entire community) that would upvote each perspective would be:

⟨ TriplesTab (65%), GraphTab (65%), EntitiesTab (52%), SchemaTab (37%), QATab (33%) ⟩

Now by considering *all* 4 options quantified as: Very Useful (4), Useful (3), Little Useful (2), Not Useful (1), we can use *Bayesian Approximation* to compute the *expected average rating* for each perspective, in the scale 1 (Worst)–4 (Best), in the entire community of users. These expected ratings are:

⟨ TriplesTab (2.84), GraphTab (2.73), EntitiesTab (2.69), SchemaTab (2.30), QATab (2.27) ⟩

where a perspective with score X means that it will have an average rating greater than X , with 95% confidence.

Task Performance. As regards *task performance*, i.e., the responses to the 11 tasks, from the $11 \times 25 = 275$ responses, 46 (16.7%) reported failure to find the requested information. The failure rate was 20.9% in the (10) users that had never used DBpedia, and 13.9% in the rest (15) users. As shown in Figure 11, the participants faced problems, mainly in T2, T4, T5: T2 is tricky (there is such a space-engineer not astronaut), while T4 and T5 are hard to answer, due to dataset issues (non-existing information, `wikiPageWikiLink` with no explanation) therefore, these cannot be considered to be failures of the system. Another interesting observation is that for most tasks inexperienced users were almost as successful as experienced ones.

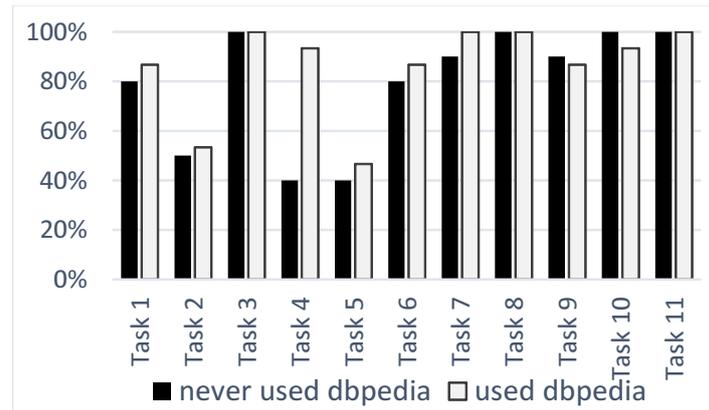


Figure 11. Success rates for experienced and inexperienced users.

Free form Feedback. With respect to the *free form feedback*, 18 of the 25 users provided very interesting and lengthy comments. For reasons of space, here we only summarize the main ones. In general, they (a) spotted problems related to the DBpedia dataset (missing relationships, unexplained `wikiPageWikiLink` relationships, duplicates), and (b) they made suggestions for improving the tabs: Triples Tab (not score with 1.0 a triple if not all query terms are included in that triple, addition of property filters), Schema Tab (add the more frequent labels in the edges of the schema graph, highlight the query words in the hits), Graph Tab (set the size so that all related entities are shown).

General Remarks. Overall, the rating and the feedback that users provided was very positive. Of course, it is not hard to understand that the results depend on the quality of each individual perspective (which in turn depends also on the effectiveness of the underlying search service). Moreover, the *order of tabs* affects the results that concern *user preferences*: in information needs that the first tab(s) provide a satisfying answer, the user will not visit the subsequent tabs (or just a few for verification purposes). That means the harder an information need is, the higher the probability the user visits all tabs. However, our main research hypothesis is not related to the comparison of the individual tabs, but on the usefulness of the multi-perspective approach, and the results of the evaluation provide positive evidence about the value of the multi-perspective approach. Overall, the key finding is that users not familiar with RDF (a) managed to complete the information-seeking tasks (with performance very close to that of the experienced users), and (b) they rated positively the approach.

5.4.4. Log Analysis

Since the system became public and was disseminated in social media on 27 April 2020, below we report some points related to the total traffic of the system; not only from the task-based evaluation with users. More than half of the users (102, in total) have interacted with at least 3 different tabs. The most visited tab is the Triples Tab (35.7% of requests for a tab) which is expected since it is the first tab presented to the user, followed by the Entities Tab (19.1%), the Schema Tab (18.7%), the Graph Tab (16.8%), and the QA tab (9.7%). On average, a user issued 4.6 requests per query (where a request involves: clicking a tab, changing page, adjusting the number of shown triples, or clicking a class or property in the schema tab). Also, a user in average performed 6.7 interactions per query in the schema tab. This is expected since the Schema Tab allows for interactive exploration of the data by clicking on classes and predicates, and adjusting the number of retrieved triples.

5.4.5. Discussion: Related Systems

To our knowledge, the only system that is currently available and offers unrestricted free-text search (which is the focus of our work) is DBpedia Search & Find (<http://dbpedia.org/fct/>). This system offers a single visualization of the results, in particular it returns *entities*, so it is like using

only the Entities Tab provided by our system. The objective of our evaluation is to investigate if a single visualization method is enough, what is answered by the user study; if the Entities Tab were enough, this would be evident in the evaluation results, e.g., in the answers of the questions E1–E7.

6. Concluding Remarks

Keyword search over RDF datasets is a challenging task. To help the user find and explore the requested information, we have investigated a *multi-perspective* approach for keyword search in which multiple perspectives (tabs) are used for the presentation of the search results, each tab stressing a different aspect of the hits. The user can easily inspect all tabs and get a better overview and understanding of the search results. We have focused on five fundamental (i.e., KB and task agnostic) perspectives (triples, entities, graph, schema and QA) and we have implemented this approach over a general keyword search engine over DBpedia.

With respect to related systems that provide keyword access over DBpedia, we could say that the proposed approach is probably the more complete with respect to the access methods that it offers. The task-based evaluation with users has shown that (a) 96% of the users liked the multi-perspective approach (48% Very much, 48% Fair), (b) the success rate of all users was very high (even of those not familiar with RDF), (c) users seem to have quite different preferences on perspectives.

There are several issues that are worth further work and research. We plan to advance the QA tab, to improve the Graph Tab, and to add additional tabs. Moreover, we would like to investigate how to exploit the equivalence (`owl:sameAs`) relationships. The system is available to all at <https://demos.isl.ics.forth.gr/elas4rdf/>.

Author Contributions: Conceptualization, Y.T.; methodology, Y.T., C.N. and P.F.; software, C.N. and G.K.; validation, Y.T., C.N., P.F.; writing—original draft preparation, Y.T., C.N. and P.F.; writing—review and editing, Y.T., C.N. and P.F.; supervision, Y.T.; project administration, Y.T.; funding acquisition, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We thank all participants of the user study for dedicating time on the evaluation and providing valuable feedback.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|--------|---|
| API | Application Programming Interface |
| TF-IDF | Term Frequency - Inverse Document Frequency |
| KB | Knowledge Base |
| OLAP | Online Analytical Processing |
| QA | Question Answering |
| RDF | Resource Description Framework |
| REST | Representational State Transfer |
| SPARQL | SPARQL Protocol and RDF Query Language |

References

1. Mountantonakis, M.; Tzitzikas, Y. Large-scale Semantic Integration of Linked Data: A Survey. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 103. [[CrossRef](#)]
2. Wishart, D.S.; Feunang, Y.D.; Guo, A.C.; Lo, E.J.; Marcu, A.; Grant, J.R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Res.* **2018**, *46*, D1074–D1082. [[CrossRef](#)] [[PubMed](#)]

3. Jaradeh, M.Y.; Oelen, A.; Farfar, K.E.; Prinz, M.; D'Souza, J.; Kismihók, G.; Stocker, M.; Auer, S. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In Proceedings of the 10th International Conference on Knowledge Capture, Marina del Rey, CA, USA, 19–22 November 2019; pp. 243–246.
4. Dimitrov, D.; Baran, E.; Fafalios, P.; Yu, R.; Zhu, X.; Zloch, M.; Dietze, S. TweetsCOVID19—A Knowledge Base of Semantically Annotated Tweets about the COVID-19 Pandemic. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Virtual Event, Ireland, 19–23 October 2020.
5. Tzitzikas, Y.; Manolis, N.; Papadakis, P. Faceted exploration of RDF/S datasets: A survey. *J. Intell. Inform. Syst.* **2017**, *48*, 329–364. [[CrossRef](#)]
6. Papadaki, M.E.; Tzitzikas, Y.; Spyrtatos, N. Analytics over RDF Graphs. In Proceedings of the International Workshop on Information Search, Integration, and Personalization, Heraklion, Greece, 9–10 May 2019; pp. 37–52.
7. Kritsotakis, V.; Roussakis, Y.; Patkos, T.; Theodoridou, M. Assistive Query Building for Semantic Data. In Proceedings of the SEMANTICS Posters&Demos, Vienna, Austria, 10–13 September 2018.
8. Kadilierakis, G.; Fafalios, P.; Papadakis, P.; Tzitzikas, Y. Keyword Search over RDF using Document-centric Information Retrieval Systems. In Proceedings of the Extended Semantic Web Conference (ESWC'2020), Heraklion, Crete, Greece, 31 May–4 June 2020.
9. Hasibi, F.; Nikolaev, F.; Xiong, C.; Balog, K.; Bratsberg, S.E.; Kotov, A.; Callan, J. DBpedia-Entity V2: A Test Collection for Entity Search. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 1265–1268.
10. Kadilierakis, G.; Nikas, C.; Fafalios, P.; Papadakis, P.; Tzitzikas, Y. Elas4RDF: Multi-perspective Triple-centered Keyword Search over RDF using Elasticsearch. In Proceedings of the Extended Semantic Web Conference (ESWC'2020), Heraklion, Crete, Greece, 31 May–4 June 2020.
11. Elbassuoni, S.; Ramanath, M.; Schenkel, R.; Weikum, G. Searching RDF graphs with SPARQL and keywords. *IEEE Data Eng. Bull.* **2010**, *33*, 16–24.
12. Lin, X.; Zhang, F.; Wang, D. RDF Keyword Search Using Multiple Indexes. *Filomat* **2018**, *32*, 1861–1873. [[CrossRef](#)]
13. Cheng, G.; Qu, Y. Searching linked objects with falcons: Approach, implementation and evaluation. *Int. J. Semant. Web Inform. Syst. (IJSWIS)* **2009**, *5*, 49–70. [[CrossRef](#)]
14. Delbru, R.; Rakhmawati, N.A.; Tummarello, G. Sindice at semsearch 2010. In Proceedings of the 19th International World Wide Web Conference, Raleigh, NC, USA, 26–30 April 2010.
15. Liu, X.; Fang, H. A study of entity search in semantic search workshop. In Proceedings of the 3rd International Semantic Search Workshop, Raleigh, NC, USA, 26–30 April 2010.
16. Delbru, R.; Campinas, S.; Tummarello, G. Searching web data: An entity retrieval and high-performance indexing model. *J. Web Semant.* **2012**, *10*, 33–58. [[CrossRef](#)]
17. Ouksili, H.; Kedad, Z.; Lopes, S.; Nugier, S. Using Patterns for Keyword Search in RDF Graphs. In Proceedings of the EDBT/ICDT Workshops, Venice, Italy, 21–24 March 2017.
18. Elbassuoni, S.; Blanco, R. Keyword search over RDF graphs. In Proceedings of the 20th ACM international Conference on Information and Knowledge Management ACM, Glasgow, UK, 19–23 October 2011; pp. 237–242.
19. Blanco, R.; Mika, P.; Vigna, S. Effective and efficient entity search in RDF data. In Proceedings of the International Semantic Web Conference, Bonn, Germany, 23–27 October 2011; pp. 83–97.
20. Pérez-Agüera, J.R.; Arroyo, J.; Greenberg, J.; Iglesias, J.P.; Fresno, V. Using BM25F for semantic search. In Proceedings of the 3rd International Semantic Search Workshop ACM, Raleigh, NC, USA, April 2010; p. 2. Available online: <https://dl.acm.org/doi/10.1145/1863879.1863881> (accessed on 27 August 2020).
21. Dosso, D.; Silvello, G. A Scalable Virtual Document-Based Keyword Search System for RDF Datasets. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 965–968.
22. Iliovski, F.; Beek, W.; van Erp, M.; Rietveld, L.; Schlobach, S. LOTUS: Adaptive text search for big linked data. In Proceedings of the European Semantic Web Conference, Crete, Greece, 29 May–2 June 2016; pp. 470–485.
23. Johnson, T. Indexing linked bibliographic data with JSON-LD, BibJSON and Elasticsearch. *Code4lib J.* **2013**, *19*, 1–11.

24. Bikakis, N.; Sellis, T. Exploration and visualization in the web of big linked data: A survey of the state of the art. *arXiv* **2016**, arXiv:1601.08059.
25. Dadzie, A.S.; Pietriga, E. Visualisation of linked data—reprise. *Semant. Web* **2017**, *8*, 1–21. [[CrossRef](#)]
26. Skjæveland, M.G. Sgvizler: A javascript wrapper for easy visualization of sparql result sets. In Proceedings of the Extended Semantic Web Conference, Crete, Greece, 27–31 May 2012; pp. 361–365.
27. Leskinen, P.; Miyakita, G.; Koho, M.; Hyvönen, E. Combining Faceted Search with Data-analytic Visualizations on Top of a SPARQL Endpoint. In Proceedings of the CEUR Workshop, Bolzano, Italy, 20–22 September 2018; pp. 53–63.
28. Vargas, H.; Buil-Aranda, C.; Hogan, A.; López, C. RDF Explorer: A Visual SPARQL Query Builder. In Proceedings of the International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019; pp. 647–663.
29. Ilievski, F.; Beek, W.; Van Erp, M.; Rietveld, L.; Schlobach, S. LOTUS: Linked Open Text UnleaShed. In Proceedings of the 6th International Workshop on Consuming Linked Data, Bethlehem, PA, USA, 12 October 2015; p. 6.
30. Rihany, M.; Kedad, Z.; Lopes, S. Keyword Search Over RDF Graphs Using WordNet. In Proceedings of the 1st International Conference on Big Data and Cyber-Security Intelligence BDCSIntell 2018, Hadath, Lebanon, 13–15 December 2018; pp. 75–82.
31. Dosso, D.; Silvello, G. Search Text to Retrieve Graphs: A Scalable RDF Keyword-Based Search System. *IEEE Access* **2020**, *8*, 14089–14111. [[CrossRef](#)]
32. Stab, C.; Nazemi, K.; Breyer, M.; Burkhardt, D.; Kohlhammer, J. Semantics visualization for fostering search result comprehension. In Proceedings of the Extended Semantic Web Conference, Crete, Greece, 27–31 May 2012; pp. 633–646.
33. Kontiza, K.; Bikakis, A. Web Search Results Visualization: Evaluation of Two Semantic Search Engines. In Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'14), Thessaloniki, Greece, 2–4 June 2014; pp. 1–12.
34. Mountantonakis, M.; Tzitzikas, Y. LODsyndesis: Global scale knowledge services. *Heritage* **2018**, *1*, 23. [[CrossRef](#)]
35. Belth, C.; Zheng, X.; Vreeken, J.; Koutra, D. What is Normal, What is Strange, and What is Missing in a Knowledge Graph: Unified Characterization via Inductive Summarization. In Proceedings of the Web Conference, Ljubljana, Slovenia, 20–24 April 2020; pp. 1115–1126.
36. Oldman, D.; Tanase, D. Reshaping the Knowledge Graph by connecting researchers, data and practices in ResearchSpace. In Proceedings of the International Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018; pp. 325–340.
37. Dimitrakis, E.; Sgontzos, K.; Tzitzikas, Y. A survey on question answering systems over linked data and documents. *J. Intell. Inform. Syst.* **2019**, *55*, 1–27. [[CrossRef](#)]
38. Cui, W.; Xiao, Y.; Wang, H.; Song, Y.; Hwang, S.W.; Wang, W. KBQA: Learning Question Answering over QA Corpora and Knowledge Bases. *Proc. VLDB Endow.* **2017**, *10*, 565–576. [[CrossRef](#)]
39. Lu, X.; Pramanik, S.; Saha Roy, R.; Abujabal, A.; Wang, Y.; Weikum, G. Answering Complex Questions by Joining Multi-Document Evidence with Quasi Knowledge Graphs. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 105–114.
40. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Berlin, Germany, 2007; pp. 722–735.
41. Moreno-Vega, J.; Hogan, A. GraFa: Scalable faceted browsing for RDF graphs. In *International Semantic Web Conference*; Springer: Berlin, Germany, 2018; pp. 301–317.
42. Heim, P.; Hellmann, S.; Lehmann, J.; Lohmann, S.; Stegemann, T. RelFinder: Revealing Relationships in RDF Knowledge Bases. In *Semantic Multimedia*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5887; pp. 182–187.
43. Ferré, S. Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. *Semant. Web* **2017**, *8*, 405–418. [[CrossRef](#)]
44. Faulkner, L. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behav. Res. Methods Instrum. Comput.* **2003**, *35*, 379–383. [[CrossRef](#)] [[PubMed](#)]

