Visual query interfaces for semantic datasets: an evaluation study☆

Guillermo Vega-Gorgojo[a], Laura Slaughter[b], Martin Giese[a,*], Simen Heggestøyl[a], Ahmet Soylu[a], Arild Waaler[a]

[a] *Department of Informatics, University of Oslo, Norway*
[b] *Oslo University Hospital, Oslo, Norway*

Abstract

The rapid growth of the Linked Open Data cloud, as well as the increasing ability to lift relational enterprise datasets to a semantic, ontology-based level means that vast amounts of information are now available in a representation that closely matches the conceptualizations of the potential users of this information. This makes it interesting to create ontology based, user-oriented tools for searching and exploring this data. Although initial efforts were intended for tech users with knowledge of SPARQL/RDF, there are ongoing proposals designed for lay users. One of the most promising approaches is to use visual query interfaces, but more user studies are needed to assess their effectiveness. In this paper, we compare the effect on usability of two important paradigms for ontology-based query interfaces: form-based and graph-based interfaces. In order to reduce the number of variables affecting the comparison, we performed a user study with two state-of-the-art query tools developed by ourselves, sharing a large part of the code base: the graph-based tool OptiqueVQS*, and the form-based tool PepeSearch. We evaluated these tools in a formal comparison study with 15 participants searching a Linked Open Data version of the Norwegian Company Registry. Participants had to respond to 6 non-trivial search tasks using alternately OptiqueVQS* and PepeSearch. Even without previous training, retrieval performance and user confidence were very high, thus suggesting that both interface designs are effective for searching RDF datasets. Expert searchers had a clear preference for the graph-based interface, and mainstream searchers obtained better performance and confidence with the form-based interface. While a number of participants spontaneously praised the capability of the graph interface for composing complex queries, our results evidence that graph interfaces are difficult to grasp. In contrast, form interfaces are more learnable and relieve problems with disorientation for mainstream users. We have also observed positive results introducing faceted search and dynamic term suggestion in semantic search interfaces.

Keywords: Semantic search, Visual query interfaces, User studies, Usability

1. Introduction

The increasing availability of Linked Data is changing the ways that developers design for interaction with Web content. Heath [1] outlines this shift in metaphor, away from the current document-centric Web, to one in which users are interacting with things (data and objects) and the connections between them. This is especially true for interfaces for retrieving information from the Web. Existing Linked Data can be used in everyday tasks, such as making decisions about a car purchase or researching the potential success of opening a new organic-food shop. However, it is unclear how average Web users can find and digest Linked Data in

☆This research has been partially funded by the Norwegian Research Council through the Semicolon II project (http://www.semicolon.no/), and the European Commission through the Optique (FP7 GA 318338) and BYTE (FP7 GA 619551) projects. The authors also want to thank the participants in the user study.

*Corresponding author. Tel.: +47 22852737.

Email addresses: guiveg@ifi.uio.no (Guillermo Vega-Gorgojo), laura.slaughter@gmail.com (Laura Slaughter), martingi@ifi.uio.no (Martin Giese), simenheg@ifi.uio.no (Simen Heggestøyl), ahmets@ifi.uio.no (Ahmet Soylu), arild@ifi.uio.no (Arild Waaler)

order to fulfill their information needs, i.e. without requiring specific knowledge of RDF and SPARQL. The ubiquitous document retrieval style interfaces for organizing and locating Web pages (i.e. documents) are not meant for finding Linked Data. One of the challenges of designing for the Semantic Web includes finding new ways to allow people to use this content [2]. Unfortunately, most tools available are not easily used by searchers having few-to-no technical skills [3]. Most are SPARQL query interfaces that require a user to learn how to write SPARQL syntax to access triplestore endpoints.

The lack of intuitive, non-technical, and novice-friendly search interfaces essentially blocks many ordinary users from gaining access to published Linked Data. To overcome this problem, there have been attempts at developing Linked Data search tools that hide SPARQL syntax from the user. Of these tools, the present types of search statement input styles include: natural language, keyword-based, form or template-based, and other visual approaches such as graphs. Some user interfaces (UIs) employ only one style of interaction at input and others use multiple (mixed) styles. For example, K-search [4] presents an ontology tree view combined with a form-based query entry interface. Results of queries, the retrieved RDF content, are often presented as tables and lists, or as graphical data objects.

Although various types of search interfaces have been developed and reported in the literature, there is a lack of empirical evidence of the effectiveness of these approaches [5]. Innovative search interfaces for querying RDF triples are often described technically and with little discussion of design for human interaction. It is not often that they have conducted usability tests. Elbedweihy et al. [6] published one of the few papers on evaluating search query approaches for the Semantic Web. They state the importance of working towards a comprehensive evaluation framework that provides guidance to developers, including design criteria for task type (e.g. search for facts) and user type (e.g. domain experts lacking technical skills), stating "there are very few studies that have focused on assessing the usability of semantic search systems."

As we have established, user studies are scarce and this motivates our work, which pursues the proposal of a portable search tool for mainstream users, the non-savvy searcher having little to no knowledge of Semantic Web technical standards, looking for facts and having well-defined searches (as opposed to ill-defined or vaguely defined information needs). For example, users such as journalists who need to know how many people in Norway voted in the last election within a specific geographical region.

The test case for the project was based on a freely accessible government data site, a Linked Open Data version of the Norwegian Company Registry[1]. Based on the prior work discussed above, we identified form-based interfaces as best for the mainstream user [6], and the PepeSearch interface was therefore constructed as a multi-class form, taking inspiration from facet-based interfaces. This design is compared to the graph-based interface of OptiqueVQS [7] that displays the underlying classes and relationships in a visual query-building environment intended for domain experts.

In this paper we present the results of a comparison study between OptiqueVQS* and PepeSearch. We collect user feedback on both interfaces, asking participants to complete specific search tasks and to report their satisfaction with the tools. Our contribution is to determine whether graph-based and form-based interfaces are effective for non-tech users in terms of retrieval performance and usability. We hypothesise that for mainstream users, form interfaces will outperform graph interfaces, and that satisfaction scores will be higher. We look at interface strengths and weaknesses, user feelings of being in control, and disorientation. In addition, we investigate user confidence in the result sets. This study fits into a larger framework that will bring further knowledge to the community of developers working on Semantic Web challenges [2, 8], addressing some of the current shortcomings of understanding the average user's perspective and needs for Semantic Web tools.

We organize the paper by first presenting relevant background on the definition of mainstream users and semantic search, as well as previously developed interfaces and evaluation studies that are related to our design. We then describe the two interfaces in detail before moving on to report the methods. Results are presented in detail followed by a discussion section that includes both limitations and plans for future work.

[1] `http://www.brreg.no/`

2. Background

2.1. Semantic search for mainstream users

The Semantic Web is moving out of the stage where only programmers and those experts with the technical skills necessary for working with raw RDF are using it. Efforts to expose data openly to citizens and customers for analysis and reuse have been made. Examples of early adopters of Linked Data for the mainstream have included open government data, such as Data.gov in the USA [9]. News media have also worked toward opening up information sources, for instance the Guardian's Open Platform[2].

Although there are now many open data sites for mainstream consumption, there exists a significant gap in support for non-technical lay users. Dadzie [10] states that "in the Web of Data, people should not be required to learn SPARQL or to have extraordinary technical skills to access data." The first step in creating tools for the non-technical is to understand these potential users and what tasks they need to accomplish. There are surprisingly few papers that discuss the characteristics of the intended audience for Linked Data.

Battle [11] proposed a framework of user types and their intended tasks specific in the context of the Semantic Web. Users are grouped in three high-level categories: (1) end-users, (2) content curators, and (3) ontologists. It is the first category of end-users that we are interested in. These users are defined as "ordinary people who are either seeking information or trying to accomplish something in the course of their everyday life or work." They could be news seekers, patients, graduate students, medical researchers, or car buyers. Their range of domain knowledge of the subject matter could be anywhere along the continuum from low to high, and the main tasks are information seeking tasks, information-synthesis tasks, and action-oriented tasks. The usability studies described above, carried out by [12, 13], cite this definition as the basis for their *casual user* category.

Dadzie et al.[10] distinguishes three types of users in the following categories: (1) lay users or mainstream, (2) domain experts, and (3) tech users. Similar to Battle's end-user definition, lay users are computer literate, have some searching skills to find resources, and are looking to do everyday tasks such as making comparisons while shopping. However, Dadzie's work defines lay users as not having much domain knowledge, and so this grouping would not include, for example, medical researchers or graduate students. Instead, there are two other categories for the non-mainstream: tech users who understand Semantic Web technologies, and another for domain experts who make use of "sophisticated domain-specific analysis tools". For the purposes of this study we adhere to this classification, focusing our interest on mainstream users.

While different definitions of *semantic search* can be found in the literature [14, ch. 3], we refer here to finding results to information needs by using Semantic Web data. This is probably the most common interpretation and, hence, it does not prescribe a particular query specification mechanism — existing approaches are analysed in Section 2.2. Further, there are different types of information needs, although there is no consensus on a single classification [15, ch. 3][16]. Broder [17] proposed a very influential taxonomy for Web searches, identifying navigational, transactional, and informational searches. In the first case the immediate intent is to reach a particular site. For the transactional type, the purpose is to perform some web-mediated activity such as e-mail checking. The goal of informational searches is to acquire some information and their complexity can vary considerably. Semantic search is especially purposed for complex informational tasks, as envisioned in the Semantic Web literature [18].

2.2. Main approaches for semantic search

When analyzing the different query specification alternatives for semantic search, we are mainly interested in the usability and expressiveness criteria. Usability refers to learnable, efficient and intuitive user interfaces [19], while expressiveness indicates the variety of queries that can be posed. Since there is a trade-off between both dimensions [20], some query expressiveness is commonly sacrificed to achieve a usability gain. Therefore, we can find in the literature the following broad types of semantic search interfaces: query editors, information retrieval-based interfaces, natural language interfaces and visual approaches. RDF browsers like LodLive [21] are also available, but they do not support querying and are just intended to navigate the contents of RDF datasets.

Query editors can be employed to compose queries using a formal language such as SPARQL,

[2] `http://www.theguardian.com/open-platform`

e.g. see the query editor provided for the DBpedia dataset at `http://dbpedia.org/sparql`. Supported expressiveness is very high since the whole set of constructs in the query language can be employed. However, users have to know the syntax of the query language. This latter requirement excludes mainstream users and domain experts, making this option only viable to tech users. Moreover, users need some familiarity with the data content and the structure, since the query editor does not give any clue.

Information retrieval approaches mimic traditional keyword queries, allowing users to pose bags of words as queries — see Sindice [22] for an example. The search system then tries to match keywords to classes or instances, possibly exploiting relations in the dataset to create a structured query. Its simplicity and its similarities with the ubiquitous web search interfaces make this approach attractive to mainstream users. However, while supported expressiveness may be sufficient for navigational or simple informational searches, it is not enough for the type of complex searches meant for the Semantic Web (see Section 2.1).

Natural language interfaces (NLIs) allow users to enter full sentence questions, and have the system attempt to find responses by interpreting the query as a whole using natural language processing techniques. With respect to usability, natural language queries are more intuitive than keyword queries, especially for novice searchers [15, ch. 4]. While NLIs try to approximate the expressiveness of natural language, their effectiveness is limited due to ambiguities and linguistic variability (known as the habitability problem [13]). To overcome this limitation, some NLIs engage the user in feedback cycles for disambiguation, e.g. Querix [23], or limit the input language to a controlled subset, e.g. Ginseng [24].

Visual approaches comprise a wide range of user interfaces that allow the construction of queries through the interaction with visual and manipulable elements. The query language syntax is hidden from the user, while available actions are limited so as to only produce valid queries. The majority of visual approaches for semantic search can be classified either as graph-based or form-based. Graph-based query editors allow to visually construct a query by adding nodes (representing concepts or individuals) and arcs (representing properties). Since Linked Data can be represented as graphs, this type of representation is a natural choice. The SPARQL syntax is quite apparent in graph-based editors like iSPARQL[3] or LuposDate [25], while approaches like NITELIGHT [26] or QueryVOWL [27] do a better job hiding the intricacies of SPARQL. However, there is evidence that mainstream users are not particularly comfortable with graph visualizations [28], making this approach questionable for this user group from a usability point of view.

Finally, form-based interfaces allow the specification of queries through the use of text boxes, drop-down menus, radio buttons and other form elements. This type of interface is commonly employed for specific search tasks such as flight booking. A prominent subcategory is facet-based interfaces [15, ch. 8], employed to obtain the instances of a class that comply with a set of restrictions. This approach has been successfully applied in e-commerce sites, e.g. to find mobile phones with a certain screen size, storage capacity, price range and from a particular brand.

Generally, form-based interfaces are adequate for mainstream users, although they tend to be specifically designed for a limited search task. Therefore, the challenge for Linked Data is to generalize form-based interfaces to be applied in an open and unrestricted context. Preliminary attempts have reproduced faceted search for semantic data by exploiting the vocabulary structure, e.g. Virtuoso Facets[4]. Unfortunately, expressiveness is typically low, since it is not possible to include more than one concept in a query. More recent developments have tried to overcome this limitation by providing pivoting operations, i.e. changing the focus of a query from one class to another, as in Rhizomer [29]. tFacet [30] supports so-called hierarchical facets, providing additional menu forms to include facets from other classes in the user query. Facet Graphs [31] provides a mixed approach that merges graphs with facets, offering a single viewpane that displays both result sets and facets as nodes in a graph visualization. In contrast, SemFacet [32] just nests all the facets from connected classes in a list. Overall, the effectiveness of these developments is uncertain for mainstream users, given their increased complexity and the scarcity of user studies to assess their benefits.

[3] `http://wikis.openlinksw.com/dataspace/owiki/wiki/OATWikiWeb/InteractiveSparqlQueryBuilder`

[4] `http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtuosoFacetsWebService`

2.3. Semantic search user studies

With the renewed interest in semantic search, two initiatives have become noticeable for evaluating semantic search systems. The Semantic Search Challenge [33] deals with entity search, i.e. obtaining a ranked list of objects from a collection of RDF data in response to keyword queries obtained from actual Web search logs. The other relevant initiative is Question Answering over Linked Data (QALD) [34] and employs research questions formulated in natural language. In both cases the focus relies on evaluating retrieval performance of search systems in a repeatable way, but neglecting the user interaction part, which is not considered at all.

Research works such as [8, 10] stress the importance of search interfaces for the success of the Semantic Web vision. Although various types of search interfaces have been developed and reported in the literature (see Section 2.2), there is a lack of empirical evidence of the effectiveness of these approaches [5]. Innovative search interfaces for querying RDF data are often described technically, with little discussion of design for human interaction, and usability studies are even more scarce. In the following paragraphs we summarize the studies found that discussed human-computer interaction, user satisfaction measures, or usability testing with respect to semantic search interfaces.

In our previous work [35], we tested a graph-based search interface with 18 university lecturers using a set of research questions aimed to find educational tools. The main goal was to assess whether keyword search would be preferable to visual-based semantic search, so we employed a regular web search interface as a baseline. We found that retrieval performance with the graph-based search interface was significantly better and indeed was the preferred option among the participants. However, some of them had problems manipulating graphs for building queries and understanding the meaning of the concept nodes.

Heim, Ertl, and Ziegler [31] compared Facet Graphs, a mixed graph-based faceted interface (see Section 2.2), with Parallax, a faceted interface with limited search capabilities. The usability test was done to evaluate the Facet Graphs interface introduced by the authors. They report that the graph-based facet approach is powerful, but participants found it complicated to use, especially for simple exploratory tasks.

In a very influential study, Kaufmann and Bernstein [13] compared four user interfaces, three NLIs with different degrees of prescription and a graph-based query builder. The authors investigated how useful the natural language query interfaces were in comparison with the visual interface that served as a baseline. Participants found this latter interface innovative, but too complicated, and this was their least preferred option. Among the available NLIs, the most user-friendly interface was situated in the middle ground of formality, allowing full-text sentences as input but also including an additional query tool to allows users to further specify intended meaning by selecting from a list of ontology classes and properties.

Finally, Elbedweihy et al. [6] compared experts' and casual users' searching using five different search interfaces (three of these were developed and tested previously in [13]): two NLIs, two graph-based and one form-based. Their study was the first with the primary goal of conducting an experiment that could inform design criteria for different users as well as task types. They report on a number of useful lessons learned when designing for Semantic Web search interaction: the group of expert users preferred graph- and form-based approaches, but got frustrated with NLIs; casual users preferred the form-based approach and found graph-based tools more complex if the entire ontology is not shown. Both types of users found the unrestricted NLI as the most natural interface, but were not happy with the results due to the mismatch between their query terms and the ones expected by the tool (the habitability problem, see Section 2.2).

In summary, empirical evidence on semantic search user studies is somewhat limited, especially for visual query approaches. The main insights from these studies are that NLIs are intuitive but not very effective; graph-based interfaces are powerful but complex; and form-based interfaces strike a balance between usability and query expressiveness. However, user studies such as [13] and [6] reported low usability ratings and performance scores were not particularly high. This suggests that the visual query interfaces under test were not particularly effective. Therefore, there is a need of finding new ways to solve those usability problems and then carry out new studies with users to validate their adequacy.

2.4. Our experimental search tools

In our work on search interfaces for the Semantic Web, we are especially interested in supporting non-tech users. Since we were not convinced about the usability of existing approaches (see Section 2.3), we have proposed two search tools with innovative visual interfaces. Our first experimental search tool is OptiqueVQS*, a visual query builder that can be used to query an arbitrary SPARQL endpoint. This tool is a branch of OptiqueVQS [36], adopting its general user interface design. OptiqueVQS* integrates graphs with form elements in the search interface, as shown in Fig. 1, and has been proposed to fulfill the demands of industrial domain experts within the Optique project[5].

The query is shown as a graph in the upper panel, representing concepts as nodes and properties as arcs. The graph is manipulable, so the user can change the position of the nodes, zoom in/out, or delete a node. New concepts can be added to the query by using the list in the bottom-left panel. If a query node is selected, the faceted widget in the bottom-right panel shows controls for refining the corresponding concept, e.g. setting a value for a property or switching to a more specific class. Once a restriction is set, it is reflected in the label of the corresponding node in the query graph. Moreover, we use a text search engine to make dynamic term suggestions during query specification – for example, after typing "os" in the municipality textbox, the following suggestions will be shown: OS, OSLO, ØSTRE TOTEN, OSTERØY, and OSEN. This allows the inclusion of existing values in the dataset without involving the user in a trial-and-error cycle.

After pushing the 'Get results!' button, a SPARQL query is built and sent to the endpoint. Obtained results are then presented in a tabular representation, as shown in Fig. 2. There are controls for sorting and paging the results, and, beyond that, the user can set a filter by clicking on a cell or remove an existing restriction, if desired. The query is actually modified, reflected in the upper panel, and might be sent to the SPARQL endpoint for new results if the query scope is broadened. All the information about a found resource can be obtained by clicking the 'i' button of the corresponding cell.

OptiqueVQS* is targeted towards domain experts, so our main design goal was to support a good expressiveness without compromising too much the usability of the tool. Although there are of course limits to the supported expressiveness, e.g. `AND` operators are used by default, it is possible to construct rather complex queries with a number of classes and restrictions. In comparison with the graph-based tools tested in [13] and [6], we have tried to hide all the RDF and SPARQL jargon, and users are freed from knowing the URI of a concept or which properties are available for a specific class. Moreover, the faceted widget provides a cohesive overview of a class and keeps the query graph simple, since there is no need to paint new nodes and arcs for each datatype property restriction. Other notable differences include dynamic term suggestion and the tidy and flexible presentation of results in OptiqueVQS* – the graph-based tools in [6] present answers as URIs alone and participants complained about their lack of readability and clarity.

Our second search tool aims to simplify the specification of queries even more. It is called PepeSearch [37], and provides a multi-class search form for querying SPARQL endpoints. This form is automatically constructed from the underlying RDF model of the dataset, making this process generally applicable. An analyzer module is employed to generate beforehand a schema out of the dataset structure by querying the target SPARQL endpoint. This schema contains all the classes, object properties, datatype properties and value types. For an arbitrary RDF class, PepeSearch creates a form block in which datatype properties are mapped to widget elements, e.g. a text-based datatype property is mapped to a text input box. We include a new form block for each RDF class that is connected with an object property to the former RDF class, e.g. see Municipality or Address in the Organization search form shown in Figure 3. Note that this form block is essentially the same component as the faceted widget employed in OptiqueVQS*.

Initially, the user selects from a list the concept she is interested in. Then, a form is shown beginning with the selected class and a series of collapsibles for each related class — see Fig. 3. Each class form block includes form controls for refining its attributes. The selection of the appropriate controls is based on the possible values of a datatype property for an RDF class as domain, as obtained by the analyzer module. For instance, if values are numerical, the minimum and maximum values in the dataset are stored in the schema and a slider is set in the form – for dates the procedure is similar. For text strings we use checkboxes if the possible values are less than six or a text input box with dynamic term suggestions otherwise.

After clicking the 'Get results!' button, Pepe-

[5] `http://optique-project.eu/`

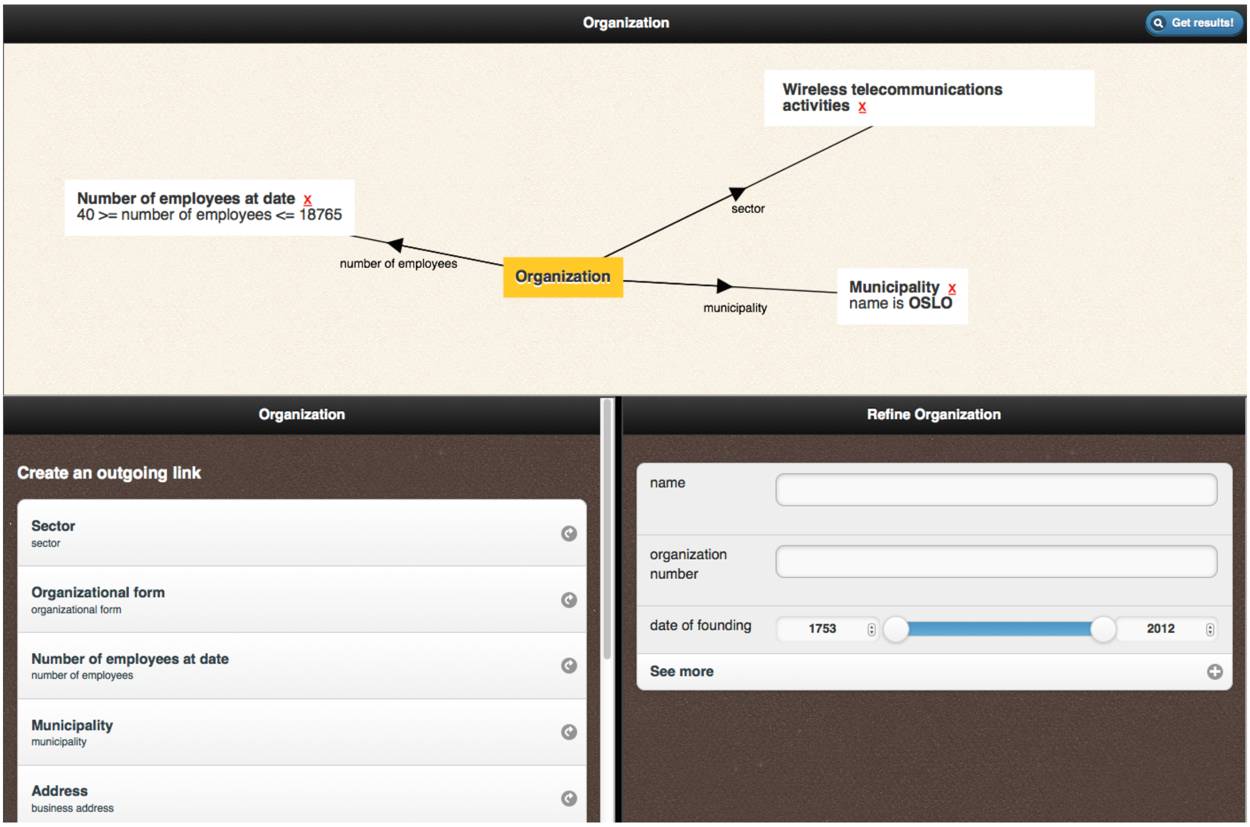


Figure 1: Snapshot of OptiqueVQS* querying for wireless telecommunication companies settled in Oslo with 40 or more employees (Norwegian Company Registry dataset).

Search transparently builds a SPARQL query with the form constraints and sends it to the endpoint. Note that classes in a collapsed state are not added to the query, since we assume that the user is not interested in them. Results are shown in a table with sorting and paging controls. The user can browse through the results by clicking the links — see Fig. 4 for an example.

While form-based interfaces can become bloated with many form elements, we have taken a number of measures in PepeSearch to limit the impact of this problem in the case of complicated RDF models. First, the use of collapsibles is an effective way to provide class summaries that the user may be interested in to fulfill her information need; expanding a collapsible class shows the attributes of the corresponding class, allowing the user to restrict them. Second, the number of collapsibles will typically be low, since a collapsible is built for each interconnection of the seed RDF class and other classes through object properties. Third, we provide controls for selecting a class type in the case of concept taxonomies. For instance, Fig. 3 shows the `Accommodation` form block, one of the 380 subtypes of `Sector` in the test dataset. And fourth, we include a "See more" button in the faceted widget (see Fig. 3) to hide the less relevant datatype properties of a class.

Our assessment as interface designers is that PepeSearch is comparatively less demanding of users than OptiqueVQS*. Composing a query with PepeSearch is similar to filling a form in which users can choose which classes to constrain. The connections among classes are automatically arranged by PepeSearch, so users do not need to deal with properties (or graphs!) as in the OptiqueVQS* case. Note, however, that PepeSearch further limits the expressiveness of the queries for the sake of simplicity. Specifically, it is not possible to build queries with long chains of concepts, unless they are directly connected through an object property (as explained above). Indeed, PepeSearch is mainly intended for mainstream users without special knowledge on Semantic Web or database topics. K-

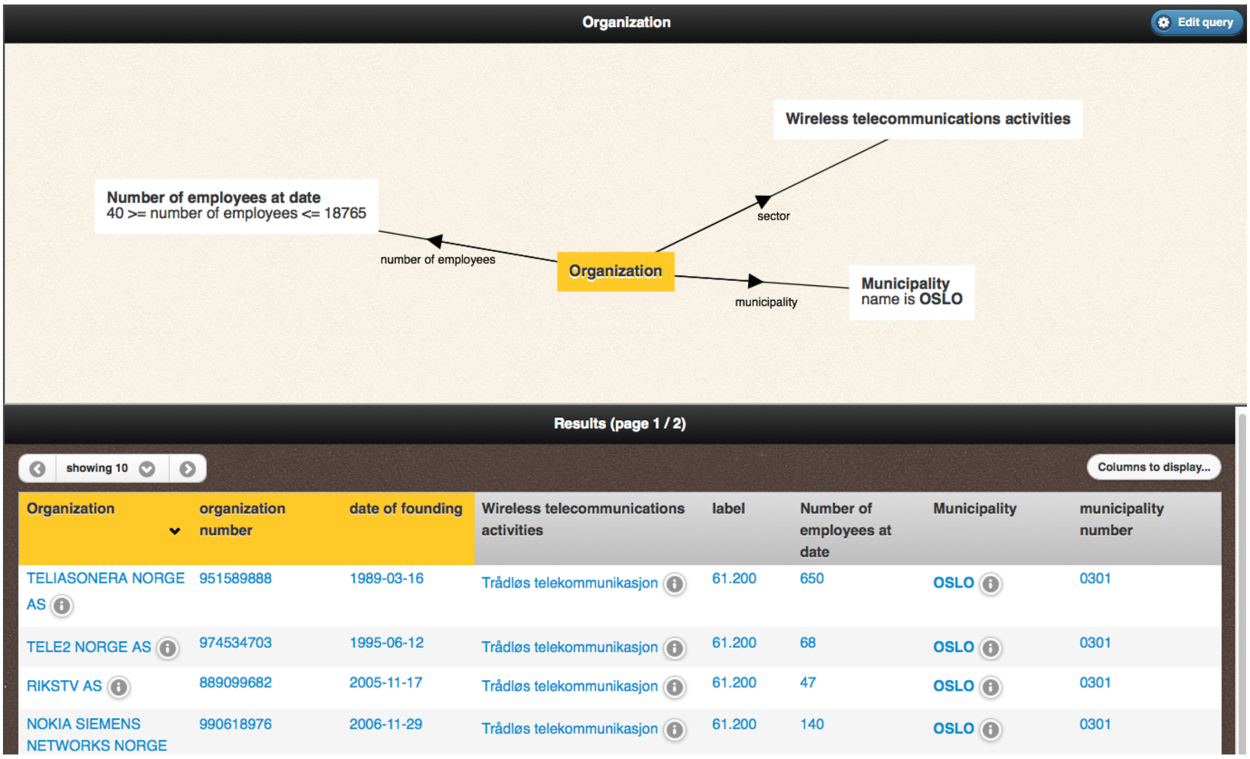


Figure 2: Snapshot of OptiqueVQS* showing a subset of the results found for the query in Fig. 1 (Norwegian Company Registry dataset).

Search, the form-based interface tested in [6], seems more complicated due to the inclusion of an ontology browser and to the use of Boolean `AND` and `OR` constructs. Furthermore, K-Search uses the same text input box for every datatype property (even in the case of fillers with numerical values) and dynamic text suggestion is not supported.

3. Experimental setup

With our experimental tools in place, we aimed to gain insight into the following hypotheses:

1. Visual query interfaces, like PepeSearch and OptiqueVQS*, are both effective for non-tech users without specific knowledge of SPARQL or RDF.
2. Form-based interfaces such as PepeSearch are adequate for mainstream searchers needing simple and learnable user interfaces.
3. Graph-based interfaces such as OptiqueVQS* are adequate for expert searchers requiring expressive queries.

We designed our study as a formal comparison of OptiqueVQS* and PepeSearch, taking inspiration from the interactive track of the Text REtrieval Conferences (TREC) [38]. This track has formalized an evaluation method of systems designed to support interactive information retrieval that has become a *de facto* standard [39]. A typical evaluation experiment involves a group of searchers that use one or more experimental systems to find information about a small set of prescribed search tasks. Common outcome measures are performance and usability; performance measures are based on the number of relevant documents searchers find, while usability measures are based on searchers' responses to questionnaires or interactions with the system.

We configured PepeSearch and OptiqueVQS* to query a version of the Norwegian Company Registry[6]. Specifically, we have employed the SPARQL endpoint exposed by the Semicolon II partner Computas[7] at `http://data.computas.com:3030/`

[6]It should be noted that OptiqueVQS* and PepeSearch can be easily set up to query other SPARQL endpoints, although this is not the focus of this evaluation study.

[7]`http://www.computas.com/`

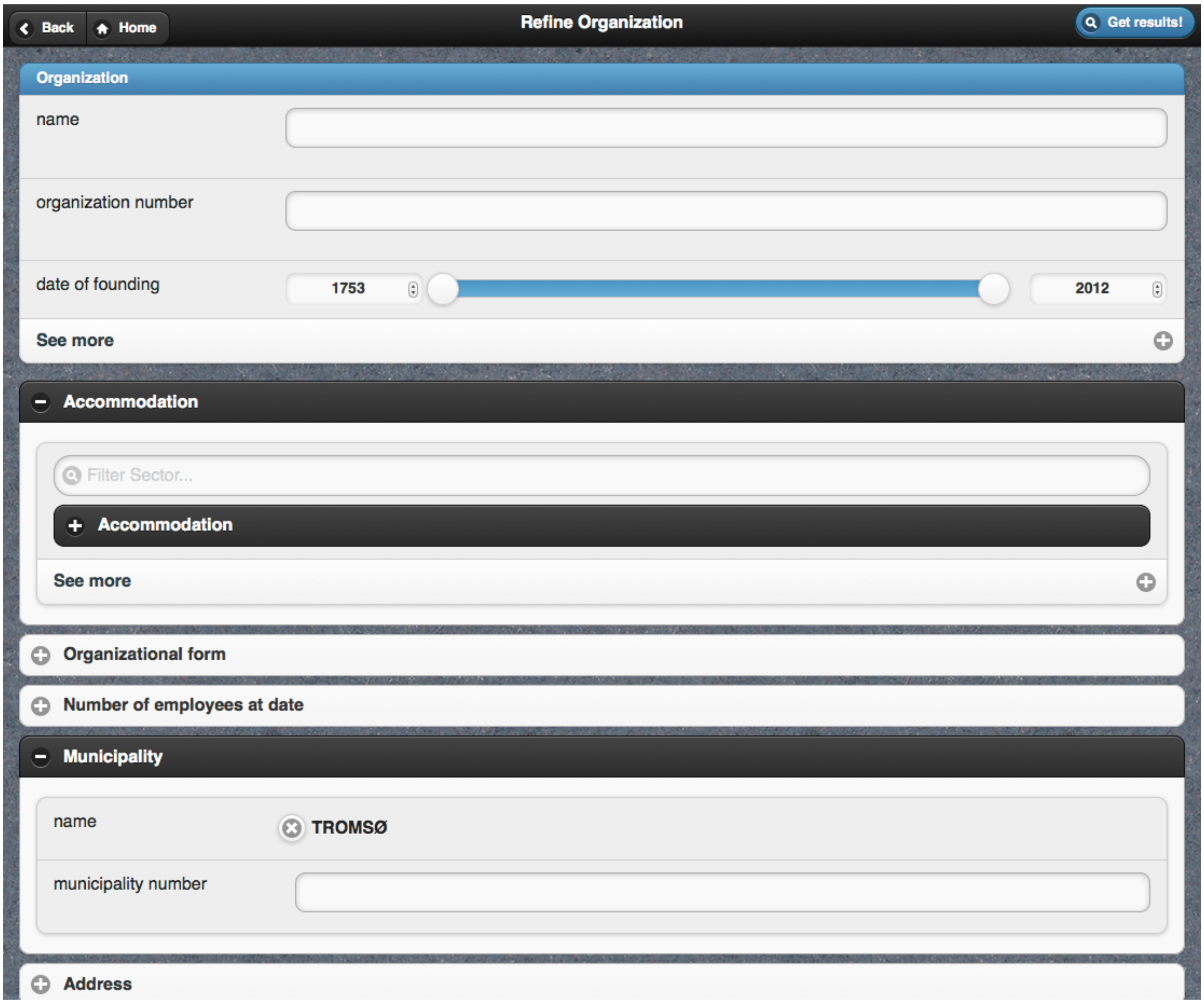


Figure 3: Snapshot of PepeSearch with a query of accommodation establishments in Tromsø (Norwegian Company Registry dataset).

`sparql`. This dataset includes administrative and financial information of Norwegian companies until 2010. In total, this dataset comprises about 50M triples of more than 354K companies. We then devised a set of six search tasks that were representative of the dataset content. We tried to choose search tasks that are meaningful and interesting for Norwegian citizens. Two of the authors proposed a preliminary list of search tasks corresponding to expected information needs for the domain of Norwegian companies. We then chose the ones we considered more relevant and tested them in a pilot session with two Norwegian students in informatics. No concerns were raised about the selected search tasks – these are shown in Table 1.

Concerning the procedure of the study, we employed a Latin Square design to balance the order of exposure of the search tasks and the tools – this design is recommended for usability studies [15, ch. 2] and is the approach taken in the interactive track of TREC. Following the Latin Square design, we defined a unique sequence of search tasks (Dairy-Wireless-Rennemo-Income-Accommodation-Brønnøy), while the starting task for each searcher was randomly determined. This was controlled with a script that sequentially presents each task with the assigned search tool. After completing a task, participants had to fill a search questionnaire with their answer and their confidence in their responses. Par-

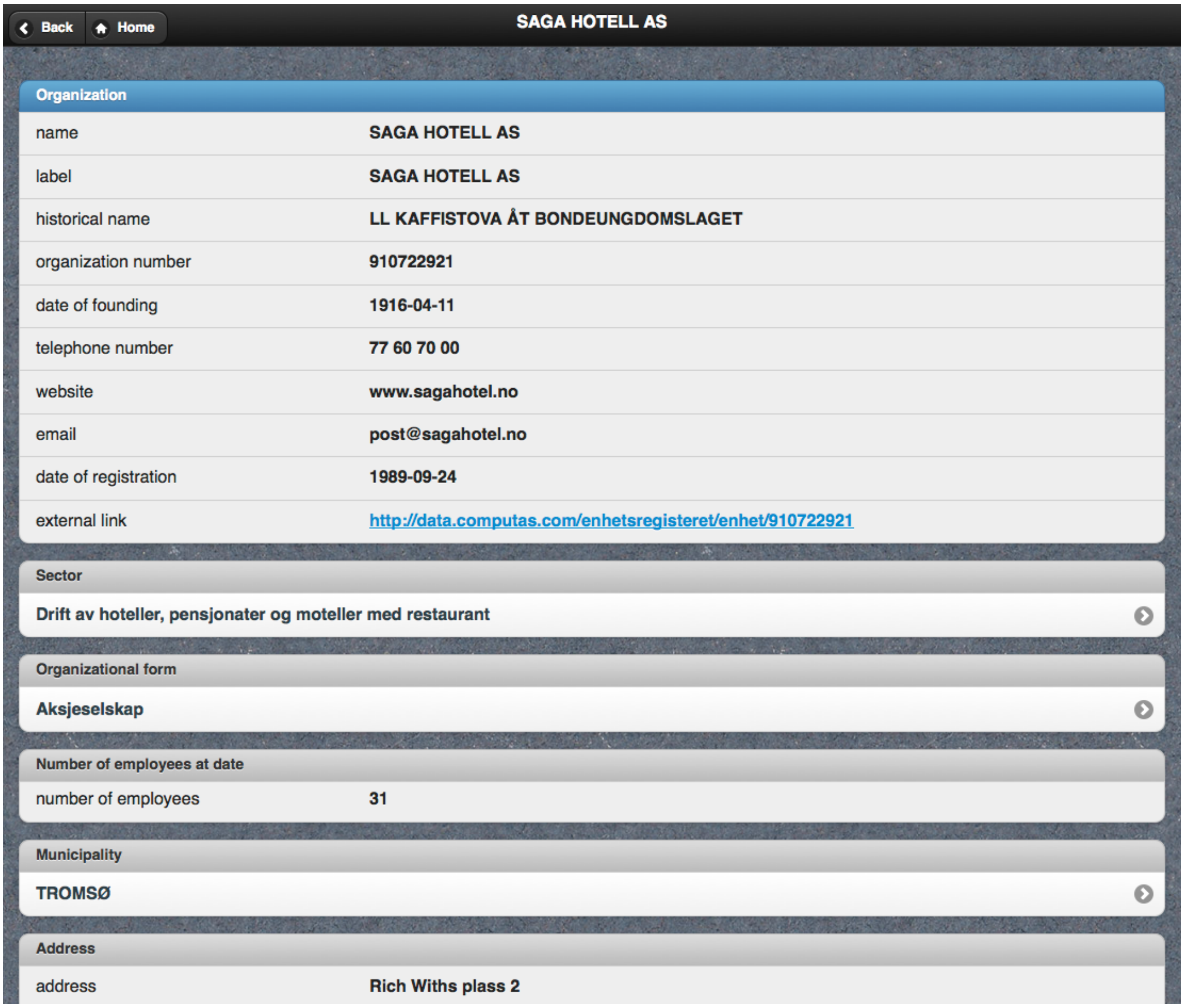


Figure 4: Snapshot of PepeSearch showing information of a hotel in Tromsø (Norwegian Company Registry dataset).

Table 1: Search tasks employed in the evaluation study.

Search task	Description
Income	Which Norwegian companies had a net income of more than 1,000,000,000 (one billion) Norwegian kroner in 2008, but no more than 4 employees?
Accommodation	Which are the eight oldest accommodation establishments (hotels or other) in Tromsø?
Brønnøy	Which companies are settled in Brønnøy, street Industriveien 30?
Dairy	Which were the seven manufacturing companies of dairy and ice-cream products with the highest net income in 2010?
Wireless	Which wireless telecommunications companies settled in Oslo have above 40 employees?
Rennemo	Which companies does "Svein Rennemo" lead?

ticipants also had to complete a usability form for each tool and an exit questionnaire for gathering feedback about the whole experience. Fig. 5 graphically depicts the procedure and identifies the data sources.

We follow a mixed methods approach [40] to evaluate the results of the study. On the quantitative side, we aim to investigate the effect of the different factors considered in the design on retrieval performance and participants' confidence. The specific factors correspond to the search tool, the search task, the order of realization, the time taken in a task, and the participants' expertise. With respect to retrieval performance, we pursued to employ a single-valued metric for the sake of simplicity, choosing the F-measure due to its suitability for interactive evaluations in which the searcher's goal is to select relevant results rather than ranking [41]. The F-measure is calculated as the harmonic mean of precision and recall – see equation 1. Precision is the percentage of retrieved documents that are relevant to an information need, while recall is the percentage of all relevant documents that are retrieved [15, ch. 2][42, ch. 3]. Given that we have multiple categorical factors that affect the response, we employ an n-way analysis of variance, or n-way ANOVA [43, pp. 250-253], to test their effect.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{1}$$

On the qualitative side, we aim to gain more insight about the study by looking through a qualitative lens to the remaining data sources. Specifically, the tool form is a usability questionnaire (see Table 6), while in the exit form we asked about the strong and weak points of each search tool, as well as their preferred tool.

In order to recruit the participants, we arranged a competition[8] with an award of a 1000 kroner book voucher for the best performer. This competition was advertised at the University of Oslo and at the Oslo Akershus University College of Applied Sciences. Eventually, 15 individuals participated in the competition with a background either in informatics or in library and information science – see Table 2. All but one were students, and 8 already held a university degree. They were generally aware of the Semantic Web and had some knowledge of databases, based on their own ratings. Interestingly, all of them were heavy users of computers and were very used to web searches. They had to register in order to participate in the study, and we sent them a link to the tutorials of OptiqueVQS* and PepeSearch a few days before the experiment — note that these tools were not available for testing.

The competition was carried out in a single session of one hour and a half. At the beginning, we welcomed the participants and gave some basic instructions for signing up and submitting the results of a search task. All the participants were able to finish before the deadline; in fact, eleven used less than an hour. It is important to note that participants did not receive any training of the two experimental search tools.

4. Evaluation results

We present here the results of the study, beginning with a quantitative analysis of search performance and participants' confidence. Then, we discuss the usability results, as well as the main strengths and weaknesses of the experimental tools.

4.1. Analysis of search performance and confidence

For the analysis of retrieval performance, we first obtained the correct responses for each search task and then computed the F-measure for each observation by analyzing the results reported in the search questionnaires. Then, we fitted an ANOVA model of the F-measure with the factors presented in Section 3. Since ANOVA models employ categorical predictors, the time employed for each task was characterized as a categorical variable with five levels using the sample quintiles to fix the threshold of each level. With respect to the participant profiles, we employed a binary variable to distinguish between mainstream and expert users. The scale defined in Table 3 was used to assign an expert score to users. A higher level of studies serves to obtain more points in the scale, while the field of expertise gives a surplus to users with a technical background. Computer skills and knowledge of expert topics are included in the scale to compensate a lack of formal education. We arbitrarily set a threshold of 12 points for classifying an individual as expert searcher – this corresponds to an individual with a bachelor in library studies, with good computer and web search skills, and with some knowledge on database and semantic web topics. This scale is flexible enough to allow an individual to be classified as an expert searcher even if she falls behind in

[8] See the *Hvor flink er du til å søke?* website at `http://heim.ifi.uio.no/guiveg/utfordring/Apen_data_utfordring/Velkommen.html`

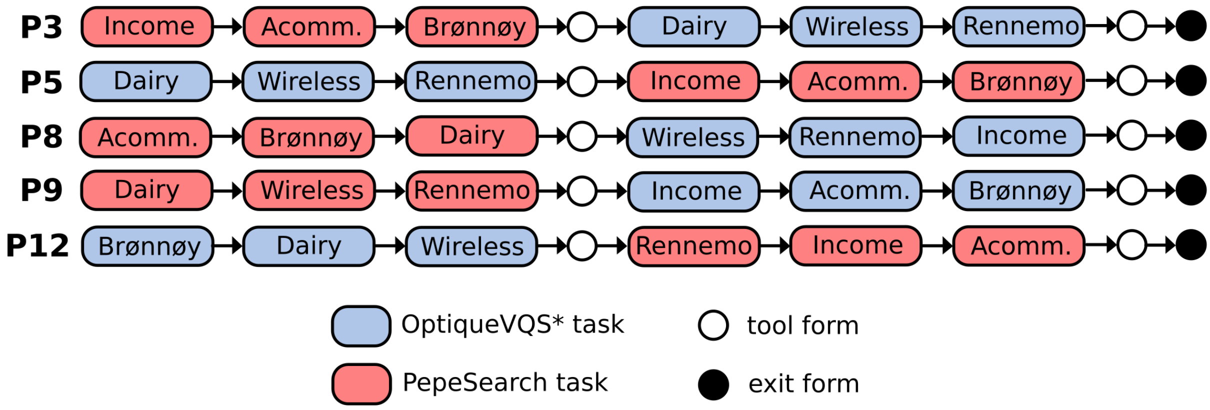


Figure 5: Illustrative procedure of the study and collected data sources for a random set of participants.

Table 2: Profiles of the participants in the user study.

PID	Highest qualification	Field	Computer skills	Web search skills	Databases	Semantic Web
P1	Master	Library	Average	Good	Know sth	Know sth
P2	High school	Library	Good	Good	Know sth	Know sth
P3	High school	Library	Good	Good	Know sth	Have heard
P4	High school	Library	Average	Good	Know well	Know sth
P5	High school	Library	Good	Very good	Know well	Know well
P6	High school	Informatics	Very good	Very good	Know sth	Know sth
P7	Master	Informatics	Good	Good	Know sth	No clue
P8	Master	Informatics	Very good	Good	Know sth	Know well
P9	PhD	Informatics	Good	Good	Know sth	Know sth
P10	Bachelor	Informatics	Very good	Good	Know well	Know well
P11	High school	Informatics	Very good	Very good	Know sth	No clue
P12	High school	Informatics	Very good	Good	Know sth	Have heard
P13	Bachelor	Informatics	Very good	Good	Know sth	No clue
P14	Bachelor	Informatics	Very good	Good	Know sth	No clue
P15	Bachelor	Informatics	Very good	Good	Know well	Know well

Table 3: Scale for search expertise.

Highest qualification		Field		Computer skills		Web search skills		Databases		Semantic Web	
High school	0	None	0	Poor	0	Poor	0	No clue	0	No clue	0
Bachelor	2	Other	1	Average	1	Average	1	Have heard	1	Have heard	1
Master	4	Library	2	Good	2	Good	2	Know sth	2	Know sth	2
PhD	6	Informatics	3	Very good	3	Very good	3	Know well	3	Know well	3

one or two categories. Participants obtained scores from 9 to 17 points, and 10 out of 15 were classified as expert searchers.

When fitting the ANOVA model, we found that the order of realization of a task alone was not relevant and there was not a significant effect among the tool and the search task. Indeed, two-factor interactions were only included in the model if they explained a significant part of the model variation. Table 4 summarizes the data of the fitted ANOVA

Table 4: Summary of analysis of variance model for the F-measure.

Source	Degrees of freedom	Sum of squares	Mean square	F value	$Prob > F$
Tool	1	0.2887	0.2887	3.9842	0.0497
Task	5	0.9777	0.1955	2.6987	0.0272
Time	4	0.7488	0.1872	2.5836	0.0441
Tool*Order	5	0.8387	0.1677	2.3151	0.0523
Tool*Expert	1	0.3612	0.3612	4.9848	0.0287
Error	72	5.2170	0.0725		
Total	88	8.7213			

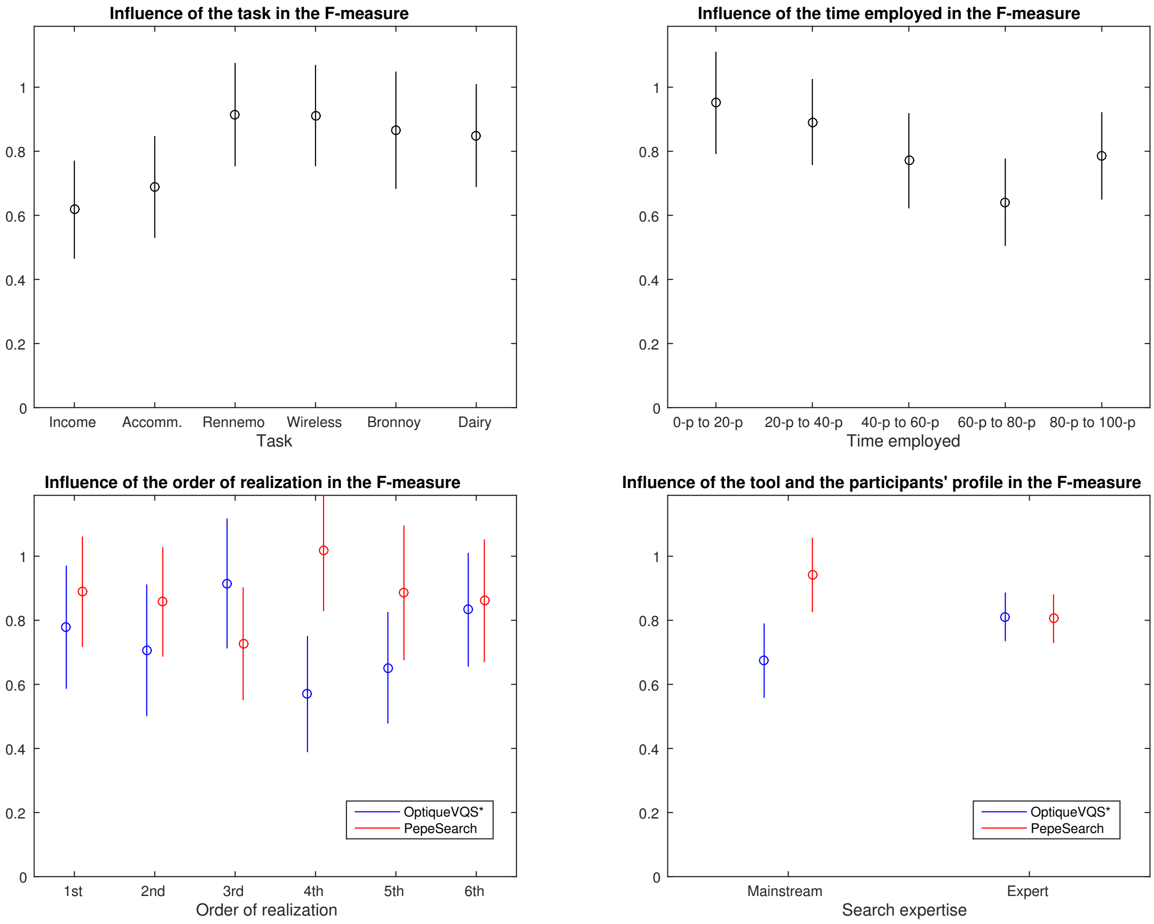


Figure 6: Influence of (a) the task, (b) the time employed, (c) the order of realization, and (d) the tool and the participants' profile in the F-measure.

model, showing the sources of variability, the variation of the response due to each source and their p-values (last column). For a standard 95% confidence level, all the sources of variation were statistically significant except the tool–order of realization interaction that was very close to significance. We checked the ANOVA assumptions [43, pp. 334] in order to assess the validity of the model. Specifically, the residuals passed the Kolmogorov-Smirnov [44] normality test and visual diagnostic tests [43, pp. 334-335] did not show trends in the mean or spread of the errors.

With the produced ANOVA model, we compared the mean F-measure of OptiqueVQS* (**0.7424**) and PepeSearch (**0.8733**) and we found that the difference is statistically significant – so the use of PepeSearch entails an improvement in retrieval performance. Remarkably, the F-measure obtained with the two experimental tools was really high, given that the maximum possible value of the F-measure is 1. Therefore, this result strongly indicates that OptiqueVQS* and PepeSearch can be effective for searching RDF datasets.

Digging deeper in the ANOVA model, we analyzed the influence of the task in the response. Though there are no significant differences, participants had more difficulties in tasks 'Income' and 'Accommodation', as shown in Fig. 6(a). Analyzing the logs, we observed that some participants had problems setting restrictions both to the financial data and the number of employees in the first case. With respect to the 'Accommodation' task, several participants just considered hotels, but not other lodging types. In the remaining tasks retrieval performance was higher and we did not find other relevant facts in the query logs.

The influence of the time employed in a search is shown in Fig. 6(b). Results indicate that better retrieval performance is positively correlated with shorter search times, obtaining a correlation of 0.77; this difference is statistically significant between the fastest and the second slowest groups – note that the confidence intervals of these two groups do not overlap. A possible interpretation is that longer time periods are associated with participants having difficulties with a search task, and thus retrieval performance is lower. We are not completely sure since we are using the time employed as a shortcut, and other factors may have an influence here, e.g. a participant might have taken additional time to browse the dataset during a search task.

While the order of realization alone is unimportant, the interaction with the tool is interesting to analyze. Although no relevant differences were found in most cases, retrieval performance is significantly higher with PepeSearch than with OptiqueVQS* in the fourth task — see Fig. 6(c). It is important to notice that participants had to switch from a tool to the other in the fourth task (see Fig. 5). This result thus shows that participants had some problems in the transition from PepeSearch to OptiqueVQS*, but not in the opposite case. Our interpretation is that OptiqueVQS* requires more effort to master than PepeSearch, making this difference evident when switching from the simplest tool to the most complex one.

We also analyzed the interaction of the tool with the participants' profile, graphically depicted in Fig. 6(d). It can be seen that participants classified as mainstream users were significantly better with PepeSearch than with OptiqueVQS*. Given that PepeSearch is intended for more casual users, this result is expected. In contrast, the expert group did not suffer appreciable differences in their performance using OptiqueVQS* or PepeSearch.

In addition to search performance, we carried out a quantitative analysis of participants' confidence that was reported after a search task[9]. Again, we were able to set up a valid ANOVA model that complied with the normality assumptions. Since the order of realization was irrelevant for explaining participants' confidence, we removed this variable from the model. In addition, we considered two-factor interactions if explaining a significant part of the variation. Table 5 summarizes the data of the fitted ANOVA model. Note that all the factors are statistically significant except the task and the tool–expertise interaction.

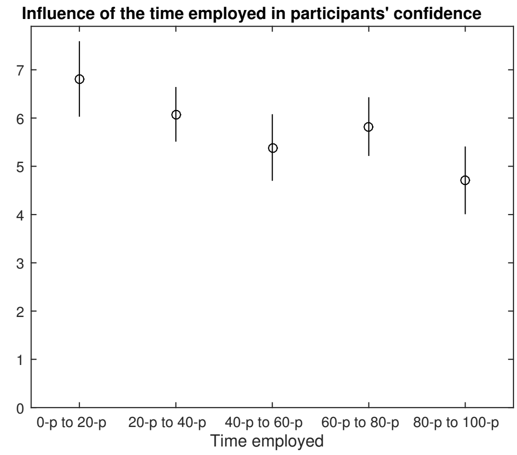
Noteworthy, participants' confidence was really high, **5.8** in a Likert scale from 1 to 7, but no differences were found using OptiqueVQS* or PepeSearch. The time employed in a task did have an impact, Fig. 7(a) shows that confidence is inversely proportional to the time spent. Indeed, the difference in confidence between the fastest and the slowest group is statistically significant. Therefore, participants spending more time on a search task were less confident in their responses. Interestingly, this same behavior was observed when analyzing search performance — see Fig. 6(b).

We also analyzed the interaction of the tool with the participants' profile, graphically depicted in Fig. 7(b). We observe that the mainstream group felt more confident with PepeSearch, while the group of experts was more confident with OptiqueVQS*. Indeed, confidence is significantly different between these two groups when using OptiqueVQS*. This result is expected given that OptiqueVQS* is purposed for more advanced users (see Section 2.4).

[9]The specific wording for this question was: "Please indicate to what extent you are confident with your response: (Very unconfident 1 2 3 4 5 6 7 Very confident)"

Table 5: Summary of analysis of variance model for the confidence.

Source	**Degrees of freedom**	**Sum of squares**	**Mean square**	F **value**	$Prob > F$
Task	5	5.4926	1.0985	0.7933	0.5580
Expert	1	19.6416	19.6416	14.1840	0.0003
Time	4	23.5137	5.8784	4.2451	0.0038
Tool*Task	5	34.5541	6.9108	4.9906	0.0006
Tool*Expert	1	3.0604	3.0604	2.2101	0.1415
Error	72	99.7034	1.3848		
Total	88	184.0449			



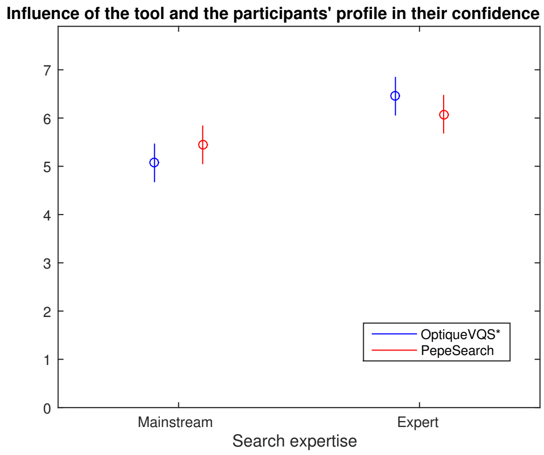


Figure 7: Influence of (a) the time employed, and (b) the tool and the participants' profile in their confidence, measured with a 7-Likert scale.

4.2. Usability of the experimental tools

For analyzing the usability of the experimental search tools we employed the questionnaire proposed in [45] for assessing subjective reactions to interactive interfaces, as recommended in [15, ch. 2]. This questionnaire was slightly adapted for the domain of search user interfaces and corresponds to the tool form in the study procedure (see Fig. 5). Most of the questions are answered with a 7-point Likert scale ranking from "strongly agree" to "strongly disagree"; the questions on aesthetics and the summary have the same scale, but the wording is different (see Table 6). The obtained results, along with the specific questions, are presented in Table 6. We also conducted a t test to assess whether there was a significant difference between the responses to OptiqueVQS* and PepeSearch.

The first set of questions refers to the *perceived ease of use* — the extent to which an individual believes that using a computer system will be free of effort. The scores of PepeSearch are 1.0 points higher on average and the difference is significantly better for questions E1 and E3, thus reflecting that participants found this tool simpler to use than OptiqueVQS*. This is expected, since PepeSearch is designed for more casual users. Although participants in the study perceived that OptiqueVQS* was more difficult, they considered that it was easy to find companies both with PepeSearch and OptiqueVQS* (see question E5). This evidences that they eventually learned to use OptiqueVQS*, as reflected by the high values on search performance analyzed in Section 4.1.

The *perceived usefulness* is defined as the extent to which users perceive that using a computer system will increase their performance. In this case the differences are less evident between the two tools

Table 6: Participants' perception of OptiqueVQS* and PepeSearch (averages, value range from 1 to 7) and p-value of the difference (* if statistically significant). Since questions with a † have a negative tone, their values are flipped to match the poles of the other questions.

ID	Question	OptiqueVQS*	PepeSearch	p-value
	Perceived ease of use			
E1	Learning to use this tool was easy	3.7	5.2	0.0193*
E2	Becoming skillful at using the tool was easy	4.3	5.3	0.0689
E3	The tool was easy to navigate	4.0	5.3	0.0200*
E4	Formulating queries with this tool was easy	4.3	5.3	0.1222
E5	Finding information about companies with this tool was easy	5.2	5.5	0.6332
	Perceived usefulness			
U1	I was able to formulate the queries that I wanted	5.0	5.2	0.7359
U2	The responses to my queries were accurate	5.5	5.8	0.5090
U3	The responses to my queries were fast	4.3	4.8	0.3870
U4	I was able to find the information that I wanted	5.7	5.6	0.8781
U5	I find this tool efficient for finding information about companies	5.3	5.7	0.2811
U6	I find this tool useful for finding information about companies	5.3	6.1	0.0969
	Disorientation			
D1	I felt lost †	3.6	5.1	0.0461*
D2	I felt like I was going around in circles †	4.6	5.9	0.0568
D3	After browsing for a while I had no idea where to go next †	5.1	6.1	0.0766
	Intensity of flow – involvement			
I1	I thought about other things †	5.5	5.9	0.4509
I2	I had to make an effort to keep my mind on the activity †	5.9	6.5	0.1257
	Intensity of flow – control			
C1	I knew the right things to do	3.9	5.1	0.0275*
C2	I felt like I received a lot of direct feedback	4.3	4.8	0.3551
C3	I felt in control of myself	4.7	5.5	0.0719
	Aesthetic quality			
	I judge the tool to be (from 1 to 7):			
A1	very complex/very simple	4.1	5.6	0.0020*
A2	very disordered/very ordered	4.3	5.4	0.0320*
A3	very ugly/very beautiful	4.8	5.0	0.6919
A4	very meaningless/very meaningful	5.9	5.9	1
A5	very incomprehensible/very comprehensible	4.6	5.6	0.0327*
	Summary			
	I judge the tool to be (from 1 to 7):			
S1	very bad/very good	4.8	5.7	0.0504

and PepeSearch obtained 0.4 points more on average. Scores are really high in this dimension, thus indicating that participants considered these tools useful for searching information about companies. However, response time for queries obtained lower ratings, especially with OptiqueVQS*. In this latter case it is easy to compose an expensive query by including lots of concepts and restrictions. To mitigate this effect we can impose some limit in the number of concepts in a query or detect problems such as query cycles. In the server side it is also possible to set up a faster machine so as to improve query response times.

Disorientation is the feeling experienced by users who do not know where they are within a system or how to move to desired locations. From the results, OptiqueVQS* was found 1.3 points worse than PepeSearch on average. Indeed, the sense of feeling lost (question D1) was perceived by many participants when using OptiqueVQS* and this difference is statistically significant. This tool can be somewhat difficult to grasp at the beginning, especially in comparison with PepeSearch. Nevertheless, scores to question D3 are much better, and our interpretation is that participants' disorientation with OptiqueVQS* was eventually diminished, given that question D3 puts the focus not at the start, but at a latter stage. In contrast, the positive results of PepeSearch suggest that this tool is easy to navigate.

Flow is a is a psychological condition in which a person feels cognitively efficient, motivated and happy. With respect to the involvement dimension, scores were very high in both cases — PepeSearch just 0.5 points better — meaning that participants were really concentrated on the proposed search tasks. Concerning the control dimension of flow, the score of OptiqueVQS* is 0.8 points less on average, especially due to responses to question C1 in which the difference was statistically significant. This seems related with the sense of disorientation discussed above.

Aesthetics are the properties of an object associated with its visual appeal. Again, participants judged the aesthetic quality of PepeSearch better than OptiqueVQS* (0.7 points on average). The biggest discrepancy is found on tool simplicity (question A1), although differences on order (A2) and comprehensibility (A5) are also significant and appreciable. Overall, the rating of OptiqueVQS* was 4.8 versus 5.7 of PepeSearch, which is consistent with the better perception of PepeSearch for each set of usability questions.

In the exit form (see Fig. 5) the participants in the study were asked to elaborate the strengths and weaknesses of each experimental search tool. After analyzing their responses we identified the strong and weak points and summarized the results for OptiqueVQS* and PepeSearch in Tables 7 and 8, respectively. We indicate the number of participants that support each case as well as a sample comment for illustration purposes. Note that we only include a point if supported by at least two participants, so as to avoid irrelevant or spurious judgements.

The obtained findings are quite revealing to understand the previous results, as well as to take actions in the near future to improve the experimental search tools. Beginning with OptiqueVQS*, the main strength is its powerfulness for creating complex queries. Participants also appreciated the visualization of queries using graphs. In contrast, the weaker points are its learnability and navigability, as discussed in the usability results.

In addition, the distinction between outgoing and incoming links for composing queries was not clear — this is employed in OptiqueVQS* to clarify the source and the target of object properties, since an RDF model is a directed graph. Some participants also indicated problems with the domain data model (see sample comment in Table 7). It is important to note that neither of these two problems were reported for PepeSearch, so this approach seems effective for hiding these details of RDF and the data model. Besides, reported crashes were due to timeouts in the SPARQL endpoint for some of the most expensive queries. Finally, two participants did not particularly like the look of the user interface.

Concerning PepeSearch, most of the participants appreciated its learnability. Other comments praised its navigability, comprehensibility and intuitiveness, in line with the usability results above. Some participants also considered that query responses were fast; since PepeSearch constrains the type of queries that can be posed, the SPARQL endpoint can respond faster than in the OptiqueVQS* case. The other side of the coin is that some complex queries cannot be formulated, and some participants stressed this point (see Table 8).

Others suggested removing the initial form with the available concepts and directly begin with organization. Although `Organization` is the most prominent class in the data model of the Norwegian Company Registry, we consider that fixing the preliminary concept makes the search interface quite inflexible and does not work well in more generic datasets with many potential relevant classes. Moreover, the initial form does not only serve to begin a query, but also provides a useful starting point for getting to know the underlying dataset. Another suggestion for improvement refers to the manipulation of sliders using the mouse, particularly with large range values; this limitation

Table 7: Strengths and weaknesses of OptiqueVQS*.

	Point	**Support**	**Sample comment**
+	Powerful	6/15	*It was easy to search on several things at the same time, you could get more specific results* [P3]
+	Good query visualization	4/15	*Good visualization of the query. It makes it easier to chain several concepts after each other.* [P10]
−	Difficult to learn	5/15	*It takes more time to understand how it works, it might require knowledge about database and sql language* [P1]
−	Difficult to navigate	5/15	*It was difficult to navigate, the design wasn't very pretty* [P3]
−	Not clear the distinction between outgoing and incoming links	4/15	*I have no idea what the directions of the arrows means. I used only outgoing because incoming didn't have all the different criteria. It seemed to work* [P6]
−	Problems with the data model	3/15	*The information under each category could be better. In "Rolle" it was unclear what the difference between "Agent" and "Verv" was. Almost every category could get better information. Maybe some tooltips?* [P15]
−	Some crashes	3/15	*It crashed some times while querying with the message "Error!"* [P15]
−	Not pretty	2/15	*The colors did not appeal to me* [P2]

Table 8: Strengths and weaknesses of PepeSearch.

	Point	**Support**	**Sample comment**
+	Easy to learn	11/15	*Easier to learn and use. Very simple* [P7]
+	Easy to navigate	5/15	*It was easy to navigate and construct queries* [P10]
+	Intuitive	3/15	*It was intuitive; it didn't feel like I had to "Figure it out", I just started using it* [P5]
+	Fast	3/15	*Gave fast responses* [P15]
+	Comprehensible	2/15	*It was the fastest and most comprehensible one* [P11]
−	Initial form not useful	3/15	*In the beginning there were not a clear way to go. If one chose organization it was much easier to get the information needed instead of choosing one of the other ways to go* [P15]
−	Multiple selection not possible	3/15	*I did not understand how to search on several "sektor" at once — something I think could be very useful* [P3]
−	Hard to connect different pieces of information	2/15	*Hard to connect different information, e.g. "regnskap" with results from other queries.* [P8]
−	Sliders can be improved	2/15	*The scale was horrible to use if you only used the mouse (net income jumping from minus some millions to plus a million)* [P4]
−	Some crashes	2/15	*Some of the searches with wide criteria took very long time (>1min), and most of those returned error* [P6]

can be easily circumvented by setting a value in the textbox to the left of the slider, but some participants were not aware of it. Others suggested to allow multiple selection of items of class tax-

Table 9: Preferred search tool.

Tool	# of votes	Sample comment
OptiqueVQS*	9 overall: 2 mainstream 7 experts	*I think OptiqueVQS* was more precise and gave me the answers I needed, but the design on PepeSearch was easier to understand and navigate. I think overall something inbetween would be perfect.* [P3]
		Even though OptiqueVQS was harder to get a grip on and uglier than PepeSearch I feel like if I got a chance to learn it properly it would be a good tool for searching.* [P2]
		Once you get the hang of it, you feel like you have more power. The speed of the results can get better though. [P10]
		I like the visual graph. It helped to make me more aware of what I was doing and where I was, like breadcrumbs on websites. [P9]
		It was faster. If I understood the graph fully, it would probably be powerful. [P6]
PepeSearch	6 overall: 3 mainstream 3 experts	*PepeSearch was easier to use which meant that I got to execute the queries faster and got faster response. The hierarchy solution in PepeSearch was easier to understand. OptiqueVQS* lacked a lot of information for a first time user and it also crashed several times.* [P15]
		Easier to use, had no problems with it as I did with OptiqueVQS.* [P7]
		It's the fastest and easiest tool. [P11]
		It was easier to understand what I was doing and where I was searching. It was intuitive that I had to go at it another way rather than get an empty result. [P4]

onomies, e.g. selecting various organization sectors at the same time.

In this exit questionnaire, we also asked the participants to select their preferred tool. Table 9 shows the votes for each tool, along with a sample of the justifications given. Supporters of PepeSearch stressed its simplicity and effectiveness, while those opting for OptiqueVQS* emphasized the potential of this tool and its powerfulness for creating complex queries. Noteworthy, these justifications coincide with our design goals for OptiqueVQS* and PepeSearch. Furthermore, supporters of OptiqueVQS* mainly correspond to the group of expert searchers (7 out of 10) which are closer to the target users of OptiqueVQS*.

5. Discussion

As a wrap-up, we interlink the insight obtained from the qualitative and quantitative methods, and generalize our results to a comparison between graph-based and form-based search interfaces. Table 10 presents the main findings of this analysis with the supporting data.

One of the main goals of this study was to assess whether the two experimental search tools are effective for searching RDF datasets. Obtained results show that both OptiqueVQS* and PepeSearch seem adequate for this purpose, given the high values of retrieval performance and user confidence in results of the searches. Moreover, perceived usefulness was highly rated by the participants in the study. In both cases, the interfaces allowed participants to formulate the queries they wanted, were efficient in obtaining the required information, and were seen to be useful. The effectiveness of both OptiqueVQS* and PepeSearch indicate that they are sensible exemplars of graph and form search interfaces, respectively.

In our study, participants were highly motivated to obtain accurate results because of the prize awarded for the highest score. They were not overly constrained by time, and they knew that accurate results existed in the queried dataset. Participants were also recruited in an environment where general computer skills are higher than average – note that 10 out 15 were classified as expert searchers. All of these factors can contribute to

Table 10: Main findings of the study and supporting data.

Finding	Supporting evidence	Source
OptiqueVQS* is effective for searching RDF datasets	Retrieval performance was very high: F-measure = 0.7646	Search
	Participants' confidence was very high: 5.9	Search
	Participants' perceived usefulness was high: 5.2	Tool form
	Participants judged OptiqueVQS* to be somewhat good: 4.8	Tool form
PepeSearch is effective for searching RDF datasets	Retrieval performance was very high: F-measure = 0.8522	Search
	Participants' confidence was very high: 5.9	Search
	Participants' perceived usefulness was high: 5.5	Tool form
	Participants judged PepeSearch to be really good: 5.7	Tool form
Expert searchers prefer graph interfaces	OptiqueVQS* was the tool of choice within the group of expert searchers	Exit form
	Slightly better performance and confidence within the group of expert searchers when using OptiqueVQS*	Search
Mainstream searchers prefer form interfaces	PepeSearch was the tool of choice within the group of mainstream searchers	Exit form
	Better performance and confidence within the group of mainstream searchers when using PepeSearch	Search
Graph interfaces are powerful but difficult to grasp	6/15 participants praised the powerfulness of OptiqueVQS*	Exit form
	Perceived ease of use with OptiqueVQS* was so-so: 4.3	Tool form
	Disorientation with OptiqueVQS* was appreciable: 4.4	Tool form
	Flow of control with OptiqueVQS* was so-so: 4.3	Tool form
	5/15 participants found OptiqueVQS* difficult to learn	Exit form
	5/15 participants found OptiqueVQS* difficult to navigate	Exit form
	Weaknesses of OptiqueVQS* and specific suggestions for improvement	Exit form
Form interfaces are learnable but not as powerful as graphs	11/15 participants praised the learnability of PepeSearch	Exit form
	Perceived ease of use with PepeSearch was high: 5.3	Tool form
	Participants stressed the navigability, intuitiveness and comprehensibility of PepeSearch	Exit form
	Weaknesses of PepeSearch and specific suggestions for improvement	Exit form
Benefits of faceted widgets and dynamic term suggestion in semantic search interfaces	Minor complaints about facets, basically some suggestions for improvement	Exit form
	The well-known issue of cluttered graphs was not raised	Exit form
	Dynamic term suggestion widely employed, especially in the 'Brønnøy' and 'Rennemo' tasks	Search
	Overall good results suggest that the use of faceted widgets and dynamic term suggestion worked well	Search

high search performance for both search interfaces. In general, the idea behind semantic search UIs such as OptiqueVQS* and PepeSearch are based on the premise of the Semantic Web, that machine-readable data and information across sites can be compared and assembled automatically, rather than relying on manual work by a human [18]. The search tasks we proposed in our study are rather similar to the tasks given by the evaluations of semantic web search UI cited in Section 2.3. These tasks are well-defined, specific, factual tasks that involve simple to more complex look-up [16]. We can generalise our results to these types of search tasks. Through the variation of tasks, we were able to uncover interface weaknesses and strengths.

As in the [6] study, our results show that expert searchers had a clear preference for the graph-based interface. OptiqueVQS* was the preferred tool, specially within the group of expert searchers for which this tool is purposed (see Table 9). This preference for graph-based interfaces is probably due to their powerfulness, since a number of participants spontaneously praised the capability of OptiqueVQS* for composing complex queries. Indeed, this is the main advantage of graph over form interfaces, corroborated in the Kaufmann study [13] with participants pointing out the enhanced query precision and expressiveness.

However, graph-based interfaces are difficult to grasp by non-tech users, especially mainstream ones (see supporting evidence in Table 10) – this is the counterbalance to the increased powerfulness and expressiveness discussed above. The main challenge for users is learning to express their information needs as query graphs. Interface designers should find ways to support this process, hiding the intricacies of RDF and SPARQL as much as possible. In this regard, the handling of properties in OptiqueVQS* can be improved, since participants were confused about the distinction between outgoing and incoming links (see Table 7). The difficulty here is showing the right amount of detail to the user. Other graph interfaces simply employ a readable label of a property, e.g. QueryVOWL. While this approach can be sufficient in some cases, the class range may also be needed if the property label is not sufficiently informative.

Concerning form interfaces, they are generally perceived as much simpler than graph-based interfaces. This was quite evident in our study, since main reactions to PepeSearch stressed the ease of learnability, navigability, intuitiveness, and comprehensibility of this tool (see Table 8). In addition, obtained retrieval performance and answers to questions E1 and E2 in Table 6 also suggest that the form-based interface was more learnable – all these metrics are commonly applied to measure learnability in the literature [46]. This may be due to the widespread use of form-based interfaces in everyday tasks. In contrast, graph-based search interfaces are not so common, thus requiring more effort to learn to use them. It is important to note that the majority of semantic search user evaluations, including this one, are formal studies in which participants test the experimental interfaces for the first time under different conditions, e.g. search tasks. Participants in our study did not receive previous training in order to better reflect this first usage situation. However, an open question is whether graph or form-based search interfaces would be preferable in the long run for the different types of users. This should be evaluated in longitudinal studies [47] that are purposed to understand how usage changes as participants learn about the system and how usage varies over a wide range of information needs.

Despite the perceived simplicity of form-based query editors, it is not easy to translate SPARQL query patterns into a generic form interface. The solution in PepeSearch is to take a proximity rule for constructing a form using the concepts that are directly connected to a seed class, previously chosen by the searcher. The study results indicate that this approach works, given the reported retrieval performance and usability scores. However, complicated queries involving long chains of concepts cannot be formulated in this way – as remarked by some participants in the study (see Table 8). Therefore, an interesting challenge is the design of form-based interfaces that are still effective and usable when further expressiveness is required.

We were also able to observe the use of a faceted widget and dynamic term suggestion in our study, given that those components are the same in OptiqueVQS* and PepeSearch. Dynamic term suggestion [15, ch. 4] and faceted search [15, ch. 8] are very popular techniques in search interfaces, although they are not so common in semantic search systems. Participants in our study profusely used suggested terms for setting literal values, especially for constructing successful queries in the 'Brønnøy' and 'Rennemo' tasks. As a result, dynamic term suggestion avoided spelling errors and contributed to reduce the number of zero results.

One problem of graph-based interfaces is that they get easily cluttered after setting a number of restrictions [6]; this is because a new node and a new arc are created with every additional constraint. In contrast, the query graph can be kept simple by using a faceted widget for the edition of the datatype properties supported by a class, thus reducing the cognitive load of the searcher. Since the issue of cluttered graphs was not raised in the study, our interpretation is that the faceted widget alleviated this problem. Moreover, facets provide a coherent view of a class by grouping its related properties. This latter aspect is especially important to organize the content of form-based interfaces in a sensible way.

6. Conclusions and Future Work

This paper presents the results of a user study and contributes to further guidelines when designing semantic search interfaces. It was our goal to demonstrate that the PepeSearch and the OptiqueVQS* interfaces can be used to successfully search an RDF datastore. To briefly summarise, we have evidence that expert searchers prefer graph-based to form-based interfaces, mainly due to the increased expressiveness of graph-based query editors. However, form-based interfaces are more easily learned and relieve problems with disorientation for mainstream searchers. We have also observed positive results introducing faceted search and dynamic term suggestion in semantic search interfaces. Overall, the use of graph-based and form-based editors empower non-tech users to explore Linked Data and find relevant information for their needs.

Since our future work involves designing for semantic search of RDF by mainstream users, our implementation plans are centred on preparing PepeSearch for use generically by any SPARQL endpoint, including federated search across sites. Our future work will go beyond look-up searches within one specific RDF data source. We also plan to consider more complex tasks, that are representative of real-life situations. We have already begun to focus on users with work-related, more complex problem-solving goals, including domain experts without a technology background (e.g. physicians) as well as more general "everyday searchers". Specifically, we have tested OptiqueVQS with targeted end-users in Siemens AG and Statoil ASA – see [48]. They carried out a series of search tasks in turbine diagnostics and oil & gas, respectively. The tasks chosen ranged from simple, e.g. "list all fields", to complex, e.g. "list all producing fields operated by Statoil Petroleum AS that has a wellbore containing gas and a wellbore containing oil". Those more complicated search tasks were comparable to the ones proposed in the study reported here. Participants from Siemens AG and Statoil ASA had no problems using the search interface and they all were able to author queries joining up to 8 relations. Note that such queries cannot be formulated with PepeSearch and for this reason it was not tested in those studies.

References

[1] T. Heath, A Taskonomy for the Semantic Web, Semantic Web 1 (1,2) (2010) 75–81.

[2] T. Heath, J. Domingue, P. Shabajee, User interaction and uptake challenges to successfully deploying Semantic Web technologies, in: Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI2006), co-located with the 5th International Semantic Web Conference (ISWC2006), Athens, GA, USA, 2006.

[3] C. Bizer, T. Heath, T. Berners-Lee, Linked Data – The story so far, International Journal on Semantic Web and Information Systems, Special Issue on Linked Data 5 (3) (2009) 1–22.

[4] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, D. Petrelli, Hybrid search: Effectively combining keywords and semantic searches, in: Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications (ESWC'08), Tenerife, Spain, 554–568, 2008.

[5] G. Hachey, Semantic Web User Interfaces: A Systematic Mapping Study, Master's thesis, Athabasca University, `http://dtpr.lib.athabascau.ca`, last visited September 2015, 2011.

[6] K. Elbedweihy, S. N. Wrigley, F. Ciravegna, Evaluating semantic search query approaches with expert and casual users, in: Proceedings of the 11th International Conference on The Semantic Web (ISWC 2012), Springer-Verlag, Boston, MA, USA, 274–286, 2012.

[7] A. Soylu, M. Giese, E. Jimenez-Ruiz, G. Vega-Gorgojo, I. Horrocks, Experiencing OptiqueVQS: a multi-paradigm and ontology-based visual query system for end users, Universal Access in the Information Society (2015) 1–24.

[8] T. Heath, M. Dzbor, E. Motta, Supporting User Tasks and Context: Challenges for Semantic Web Research, in: Proceedings of the Workshop on End-User Aspects of the Semantic Web (UserSWeb 2005), Heraklion, Crete, 2005.

[9] L. Ding, D. Difranzo, A. Graves, J. R. Michaelis, X. Li, D. L. McGuinness, J. Hendler, Data-gov Wiki: Towards Linking Government Data, in: 2010 AAAI Spring Symposium Series, 2010.

[10] A. Dadzie, M. Rowe, Approaches to visualising linked data: A survey, Semantic Web 2 (2) (2011) 89–124.

[11] L. Battle, Preliminary Analysis of Users and Tasks for the Semantic Web, in: Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI 2006), Athens, GA, USA, 2006.

[12] E. Kaufmann, A. Bernstein, How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?, in: Proceedings of the 6th International Semantic Web Conference (ISWC'07), no. 4825 in LNCS, Springer-Verlag, Busan, Korea, 281–294, 2007.

[13] E. Kaufmann, A. Bernstein, Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases, Web Semantics: Science, Services and Agents on the World Wide Web 8 (4) (2010) 377–393.

[14] K. Elbedweihy, Effective, Usable and Learnable Semantic Search, Ph.D. thesis, University of Sheffield, UK, 2014.

[15] M. Hearst, Search user interfaces, Cambridge University Press, 2009.

[16] B. M. Wildemuth, L. Freund, Search tasks and their role in studies of search behaviors, in: Third Annual Workshop on Human Computer Interaction and Information Retrieval, Citeseer, Washington, WA, USA, 2009.
[17] A. Broder, A taxonomy of web search, SIGIR forum 36 (2) (2002) 3–10.
[18] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American 284 (5) (2001) 34–43.
[19] J. Nielsen, Usability 101: Introduction to Usability, URL: http://www.nngroup.com/articles/usability-101-introduction-to-usability/, last visited September 2015, 2012.
[20] A. Freitas, E. Curry, J. G. Oliveira, S. O. Riain, Querying heterogeneous datasets on the linked data web: Challenges, approaches, and trends, Internet Computing, IEEE 16 (1) (2012) 24–33.
[21] D. V. Camarda, S. Mazzini, A. Antonuccio, LodLive, exploring the Web of Data, in: Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS 2012), Graz, Austria, 197–200, 2012.
[22] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, G. Tummarello, Sindice. com: a document-oriented lookup index for open linked data, International Journal of Metadata, Semantics and Ontologies 3 (1) (2008) 37–52.
[23] E. Kaufmann, A. Bernstein, R. Zumstein, Querix: A natural language interface to query ontologies based on clarification dialogs, in: Proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, GA, USA, 980–981, 2006.
[24] A. Bernstein, E. Kaufmann, C. Kaiser, Querying the semantic web with Ginseng: A guided input natural language search engine, in: Proceedings of the15th Workshop on Information Technologies and Systems (WITS 2005), Las Vegas, NV, USA, 112–126, 2005.
[25] J. Groppe, S. Groppe, A. Schleifer, V. Linnemann, LuposDate: A Semantic Web Database System, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09), Hong Kong, China, 2083–2084, 2009.
[26] A. Russell, P. R. Smart, D. Braines, N. R. Shadbolt, Nitelight: A graphical tool for semantic query construction, in: Semantic Web User Interaction Workshop (SWUI 2008), Florence, Italy, 2008.
[27] F. Haag, S. Lohmann, S. Siek, T. Ertl, QueryVOWL – Visual Composition of SPARQL Queries, in: Proceedings of the 12th European Semantic Web Conference (ESWC2015), Portoroz, Slovenia, 2015.
[28] F. B. Viégas, J. Donath, Social network visualization: Can we go beyond the graph?, in: Proceedings of the 2006 20th anniversary conference on Computer Supported Cooperative Work (CSCW'06), Workshop on Social Networks, vol. 4, Banff, Canada, 6–10, 2004.
[29] J. M. Brunetti, R. García, S. Auer, From Overview to Facets and Pivoting for Interactive Exploration of Semantic Web Data, International Journal on Semantic Web and Information Systems (IJSWIS) 9 (1) (2013) 1–20.
[30] S. Brunk, P. Heim, tFacet: Hierarchical faceted exploration of semantic data using well-known interaction concepts, in: Proceedings of the International Workshop on Data-Centric Interactions on the Web (DCI 2011), Lisbon, Portugal, 31–36, 2011.
[31] P. Heim, T. Ertl, J. Ziegler, Facet Graphs: Complex Semantic Querying Made Easy, in: L. Aroyo, G. Antoniou, E. Hyvönen, A. t. Teije, H. Stuckenschmidt, L. Cabral, T. Tudorache (Eds.), The Semantic Web: Research and Applications, no. 6088 in LNCS, Springer-Verlag, 288–302, 2010.
[32] M. Arenas, B. Cuenca Grau, E. Kharlamov, S. Marciuska, D. Zheleznyakov, E. Jimenez-Ruiz, SemFacet: semantic faceted search over yago, in: Proceedings of the companion publication of the 23rd International Conference on World Wide Web (WWW'14), Seoul, Korea, 123–126, 2014.
[33] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, T. Tran, Repeatable and reliable semantic search evaluation, Web Semantics: Science, Services and Agents on the World Wide Web 21 (2013) 14–29.
[34] V. Lopez, C. Unger, P. Cimiano, E. Motta, Evaluating question answering over linked data, Web Semantics: Science, Services and Agents on the World Wide Web 21 (2013) 3–13.
[35] G. Vega-Gorgojo, M. L. Bote-Lorenzo, J. I. Asensio-Pérez, E. Gómez-Sánchez, Y. A. Dimitriadis, I. M. Jorrín-Abellán, Semantic search of tools for collaborative learning with the Ontoolsearch system, Computers & Education 54 (4) (2010) 835–848.
[36] A. Soylu, M. Giese, E. Jimenez-Ruiz, E. Kharlamov, D. Zheleznyakov, I. Horrocks, OptiqueVQS: towards an ontology-based visual query system for big data, in: Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems (MEDES'13), Neumünster Abbey, Luxembourg, 119–126, 2013.
[37] S. Heggestøyl, G. Vega-Gorgojo, M. Giese, Visual Query Formulation for Linked Open Data: The Norwegian Entity Registry Case, in: Proceedings of the 27th Norsk Informatikkonferanse (NIK 2014), Østfold, Norway, 2014.
[38] E. Lagergren, P. Over, Comparing interactive information retrieval systems across sites: the TREC-6 interactive track matrix experiment, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '98), ACM Press, Melbourne, Australia, 164–172, 1998.
[39] D. Kelly, S. Dumais, J. O. Pedersen, Evaluation Challenges and Directions for Information-Seeking Support Systems, Computer 42 (3) (2009) 60–66.
[40] A. Tashakkori, C. Teddlie (Eds.), Handbook of Mixed Methods in Social & Behavioral Research, Sage Publications, Inc., Thousand Oaks, CA, USA, 2003.
[41] D. W. Oard, Evaluating interactive cross-language information retrieval: Document selection, in: Proceedings of the First Cross-Language Evaluation Forum (CLEF 2000), LNCS 2069, Springer, Lisbon, Portugal, 57–71, 2001.
[42] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, Harlow, UK, first edn., 1999.
[43] R. Jain, The art of computer systems performance analysis. Techniques for experimental design, measurement, simulation and modeling, John Wiley, New York, NJ, USA, 1991.
[44] H. W. Lilliefors, On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown, Journal of the American Statistical Association 62 (318) (1967)

399–402.

[45] P. van Schaik, J. Ling, Five psychometric scales for online measurement of the quality of human-computer interaction in web sites, International Journal of Human-Computer Interaction 18 (3) (2005) 309–322.

[46] T. Grossman, G. Fitzmaurice, R. Attar, A survey of software learnability: metrics, methodologies and guidelines, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2009), Boston, MA, USA, 649–658, 2009.

[47] B. Shneiderman, C. Plaisant, Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies, in: Proceedings of the 2006 conference Advanced Visual Interfaces (AVI'06), Workshop on Beyond time and errors: novel evaluation methods for information visualization, Venezia, Italy, 1–7, 2006.

[48] A. Soylu, E. Kharlamov, D. Zheleznyakov, E. Jimenez-Ruiz, M. Giese, I. Horrocks, Ontology-based Visual Query Formulation: An Industry Experience, in: Proceedings of the 11th International Symposium on Visual Computing (ISVC 2015), Las Vegas, NV, USA, 2015.