NL-Graphs: A Hybrid Approach toward Interactively Querying Semantic Data

Khadija Elbedweihy, Suvodeep Mazumdar, Stuart N. Wrigley, and Fabio Ciravegna

Department of Computer Science, University of Sheffield, UK
{k.elbedweihy, s.mazumdar, s.wrigley, f.ciravegna}@dcs.shef.ac.uk

Abstract. A variety of query approaches have been proposed by the semantic web community to explore and query semantic data. Each was developed for a specific task and employed its own interaction mechanism; each query mechanism has its own set of advantages and drawbacks. Most semantic web search systems employ only one approach, thus being unable to exploit the benefits of alternative approaches. Motivated by a usability and interactivity perspective, we propose to combine two query approaches (graph-based and natural language) as a hybrid query approach. In this paper, we present NL-Graphs which aims to exploit the strengths of both approaches, while ameliorating their weaknesses. NL-Graphs was conceptualised and developed from observations, and lessons learned, in several evaluations with expert and casual users. The results of evaluating our approach with expert and casual users on a large semantic dataset are very encouraging; both types of users were highly satisfied and could effortlessly use the hybrid approach to formulate and answer queries. Indeed, success rates showed they were able to successfully answer all the evaluation questions.

Keywords: #eswc2014Elbedweihy

1 Introduction and Related Work

The overall goal of a (semantic) search system is to assist users in fulfilling their information needs. Users' experience and satisfaction with this information seeking process is influenced by many aspects including the query format, the performance of the search system and the presentation of the results. Most of the work done to date has focused on the second aspect: improving the effectiveness and retrieval performance of semantic search systems. Several query approaches have also been proposed over the years, with varying levels of interactivity and user involvement. While the progress in these areas has been encouraging, the third aspect (usability) has been largely ignored by the Semantic Web community[1].

[1] A significant lack of attention to usability and user experience has been a concern to the community. Indeed, David R. Karger's keynote talk (`http://videolectures.net/eswc2013_karger_semantic/`) stressing the need for designing Semantic Web solutions with explicit attention to users bears testimony to the need for such solutions

In our previous work [7], we conducted a user-based study to evaluate the usability of different query approaches with expert and casual users. The results showed that, on one hand, both types of users liked the support given by view-based (graph- and form-based) approaches in constructing queries through visualising the search space. On the other hand, the main drawback of these approaches was the amount of effort and time required to formulate queries. Therefore, in [17], we presented the results of a learnability study which investigated whether the effects of the latter could be alleviated by practice and frequency of use of a graph-based tool (Affective Graphs). The results revealed an improvement in users' performance as well as satisfaction over time, while the effort and time involved in query formulation for frequent tasks needed much improvement, even after three practice sessions. Additionally, observations during the evaluations noted that users heavily used a (text-based) concept-search feature in addition to manual visual lookups while performing either task. This was highly interesting, as users were observed to be combining different mechanisms to perform their tasks. This finding motivated the next stage of our work and seeded ideas for a more integrated hybrid approach that can harmoniously combine differing query approaches.

In broad terms, a *hybrid query approach* uses a combination of approaches (natural language (NL)-based, keyword-based, graph-based or form-based) as the query format. However, the term *hybrid approach* has been used interchangeably in literature with *hybrid search* and *hybrid web search* to refer to different concepts. [8] and [16] use one or more of these terms to describe their application of semantic web techniques (such as using ontologies to find concepts related to the input query terms) to improve the precision of traditional keyword-based search. In a different way, [3] used two query formats: keywords and forms to perform both keyword-based traditional search and semantic search, respectively, and combine the results of both. The two query approaches are separated and linked to two different underlying data indexes. The keyword-based approach is used to search traditional documents while the form-based approach is used to visualise semantic data and ontologies. [10] combined keyword search and view-based search to support users in formulating their search queries and offer them flexibility in expressing their information needs. Their methodology is based on mapping the underlying domain ontologies into views, which facilitates view-based search.

Based on the findings from our evaluations, we propose a hybrid approach that benefits from the strength of the graph-based approach in visualising the search space, while attempting to balance the time and effort required during query formulation using a NL input feature. We hypothesise that this would provide a high level of support and satisfaction for users during query formulation. To evaluate this hypothesis, we developed our approach – NL-Graphs – as a proof of concept and conducted a third user-based study with expert and casual users to assess its usability and user satisfaction. However, it is important to note that the NL- and the graph-based components were evaluated separately in recent work [6, 17] and thus, the evaluation described here focuses on the usability of the hybrid approach as a new query mechanism. Our contribution in this paper is three-fold:

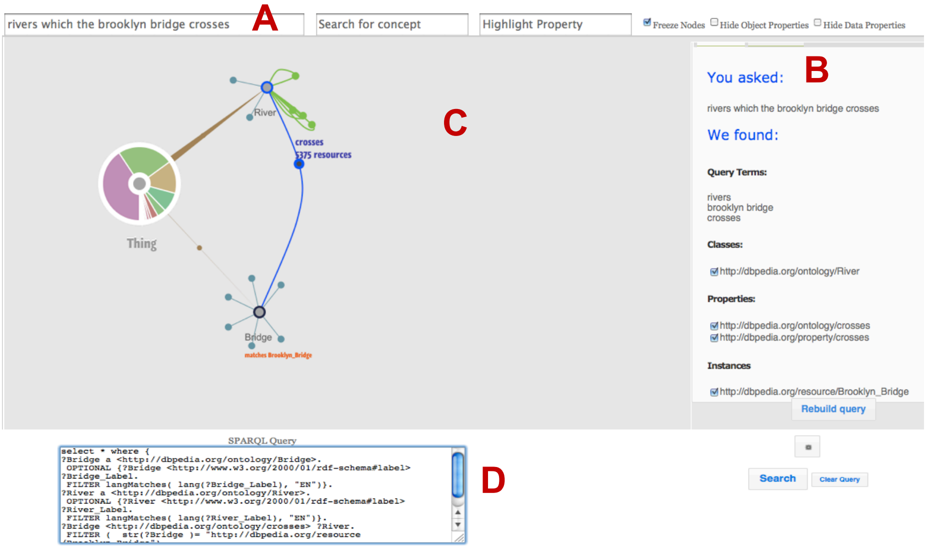
- Propose a hybrid query approach combining the benefits of NL and view-based approaches.
- Validate our approach with a user-based evaluation involving 24 participants.
- Compare how casual and expert users interact with the query approach.

The remainder of the paper is structured as follows: the architecture of NL-Graphs (implementing the hybrid approach) is described in Section 2, together with illustrative scenarios showing the querying experience. Then, in Section 3, the usability study conducted to assess the usability of the approach and its usefulness in supporting users during query formulation is presented. Finally, conclusions and future work are discussed in Section 4.

2 NL-Graphs: A Hybrid Approach toward Querying Semantic Data

As discussed previously, the hybrid approach presented here combines two different query approaches (NL- and graph-based) to support users during query formulation. *NL-Graphs* is implemented as a proof-of-concept for realising this hybrid approach. It combines the NL-approach, presented in details in [6], and Affective Graphs, similarly presented in [17]. Therefore, we limit our description to the core functionalities of each constituent approach and the interaction between them in NL-Graphs as shown in Figure 1.

- Text Entry for NL Query (A): This allows the user to enter a NL query. Since the main drawback of the graph-based approach – when used separately – was the amount of effort and time required to formulate queries, this component provides the means for an easy-and-fast starting point for query construction. Users are free to enter keywords, phrases or full questions.
- Input Interpretation and Query Validation (B): One of the main difficulties for NL-based query approaches is mapping users' query terms onto the correct ontological concepts and properties and Linked Data entities. This is necessary to understand the correct query intent and, in turn, provide accurate answers. The employed NL-approach – similar to state-of-the-art NL-approaches – does not yet experience very high performance in this aspect and hence some query terms can be incorrectly mapped to concepts, properties or instances. As such, this component is intended to provide users with the ability to verify the interpretation of the system for their input query and perform corrections if needed.
- Visual Approach (C): As stated above, the output of the NL-component might contain incorrect interpretations of the user's query, or could be incomplete when no suitable mappings are found for one or more query term. Therefore, the visual approach provides the means for users to 1) verify the interpretation of the system for their input query; 2) correct or complete the



rivers which the brooklyn bridge crosses

You asked:

rivers which the brooklyn bridge crosses

We found:

Query Terms:

rivers
brooklyn bridge
crosses

Classes:

☑ http://dbpedia.org/ontology/River

Properties:

☑ http://dbpedia.org/ontology/crosses
☑ http://dbpedia.org/property/crosses

Instances

☑ http://dbpedia.org/resource/Brooklyn_Bridge

SPARQL Query

```
select * where {
?Bridge a <http://dbpedia.org/ontology/Bridge>.
 OPTIONAL {?Bridge <http://www.w3.org/2000/01/rdf-schema#label>
?Bridge_Label.
 FILTER langMatches( lang(?Bridge_Label), "EN")}.
?River a <http://dbpedia.org/ontology/River>.
 OPTIONAL {?River <http://www.w3.org/2000/01/rdf-schema#label>
?River_Label.
 FILTER langMatches( lang(?River_Label), "EN")}.
?Bridge <http://dbpedia.org/ontology/crosses> ?River.
 FILTER (  str(?Bridge )= "http://dbpedia.org/resource
```

Fig. 1. NL-Graphs interface for the query "*rivers which the brooklyn bridge crosses*".

visual query which is automatically built using the NL-component's output – as will be explained later; 3) understand the structure of the underlying data; and finally 4) explore the context surrounding their query (related concepts and properties).

- Formal Query (D): Having the formal query presented for users in the interface is motivated by the results of the usability study discussed in [7]. These showed that the formal representation of the constructed queries provided experts with the means to verify the queries and, therefore, increased their confidence in what they were doing. Additionally, this component provides expert users with an alternative to the above methods to perform direct changes to their queries (which was shown to increase the expressiveness of the query language as reported in the same study). Note that this component can be hidden for casual users since the same study has shown that the presentation of the formal query is not suitable for them and does not provide any added advantage to casual users.

2.1 NL-Graphs Architecture

As shown in the workflow presented in Figure 2, a user's query is first processed by the NL-component. The steps: 1) recognition and disambiguation of named entities; 2) parsing the NL query; 3) matching query terms with ontology concepts and properties; and finally 4) generation of candidate triples, explained in details in [6], are applied in order. In these steps, firstly, named entities are recognised using AlchemyAPI and disambiguated using a word sense disambiguation

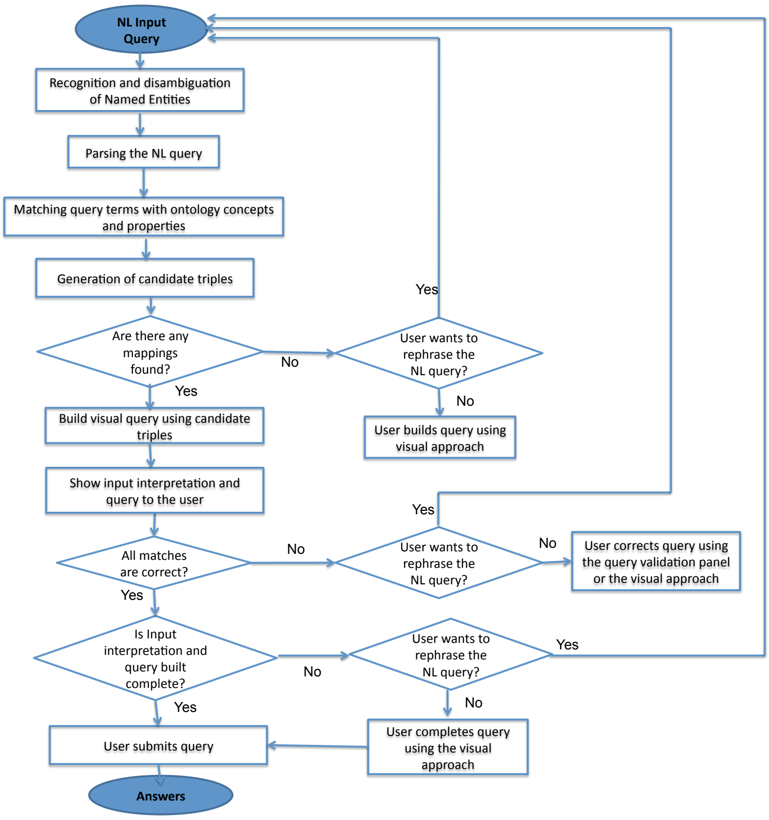

NL Input Query
Recognition and disambiguation of Named Entities
Parsing the NL query
Matching query terms with ontology concepts and properties
Generation of candidate triples
Are there any mappings found?
No
Yes
User wants to rephrase the NL query?
Yes
No
User builds query using visual approach
Build visual query using candidate triples
Show input interpretation and query to the user
All matches are correct?
No
Yes
User wants to rephrase the NL query?
Yes
No
User corrects query using the query validation panel or the visual approach
Is Input interpretation and query built complete?
No
Yes
User wants to rephrase the NL query?
Yes
No
User completes query using the visual approach
User submits query
Answers

Fig. 2. NL-Graphs workflow

(WSD) approach as described in [6]. Then, the Stanford parser is used to generate the lemma and part of speech (POS) tag of each query term and store them with the term's position with respect to the rest of the query. These terms are then matched to concepts and properties in the underlying ontologies. Noun phrases, nouns and adjectives are matched with both concepts and properties, while verbs are only matched with properties. The structure of the ontology (taxonomy of classes and domain and range information of properties) is then used to link the identified matches (concepts, properties and instances) to generate candidate triples. An example of these triples is shown below:[2]:

```
<res:Brooklyn_Bridge> <dbo:crosses> ?river.
?river a <dbo:River>.
```

These triples are then passed to the graph-based component. Even if no complete triples are generated (for instance, if only one query term was matched

[2] The prefix `res` refers to: `<http://dbpedia.org/resource/>`, `dbp` refers to: `<http://dbpedia.org/property/>` and `dbo` refers to: `<http://dbpedia.org/ontology/>`.

with an ontology concept or with an instance) these mappings are similarly passed to the graph-based component to be visualised in the graphical panel (Figure 1: C). This is performed within the next step in the workflow: "*Build visual query using candidate triples*".

In the graph-based component, any concepts found in the list of terms received from the NL-component are analysed first. Each concept is loaded, along with all its respective data and object properties[3]. Then, the instances are analysed where each instance type is added into the existing query, and a restriction (constraint) value of the instance is applied on the concept. For example, the concept *River* is loaded first and then, the constraint `res:Brooklyn_Bridge` is then applied on the concept as a text filter. The properties are finally analysed, the concepts which are domains or ranges for a property are loaded (if not previously loaded). When the analysis of all terms is complete, the final step includes loading the visual query, inspecting the query variables and completing the formal query.

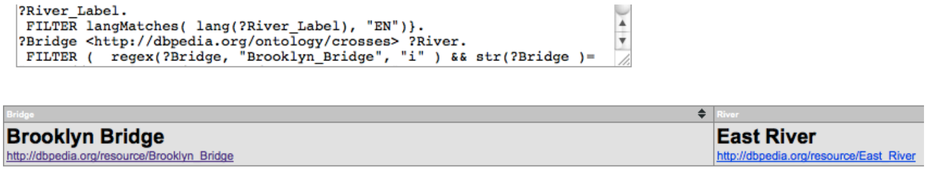
Next in the workflow, the interpretation of the NL query – all matches for concepts, properties or instances – is shown in the query validation panel on the middle right side of the user interface (Figure 1: B). Additionally, the output of the graph-based component – either mappings or visual query – is displayed in the graphical panel on the middle left side of the user interface (Figure 1: C). If the system's interpretation for the user's query contains incorrect mappings, then the user can correct them using either of these panels according to their preference. Otherwise, the user can continue to submit the query if the system's interpretation and the query built were complete – entities, concepts and relations connecting them were identified. If any of the latter was missing, then the user can complete the query using the visual approach as will be explained in the next section.

2.2 Querying in NL-Graphs: – The User Experience

In order to begin the querying process with NL-Graphs, the user enters a NL query into the search box as shown in Figure 1:(A). In this example, the user enters the phrase "*rivers which the brooklyn bridge crosses*". Similar output would be generated for the complete question "*Give me all rivers which the brooklyn bridge crosses.*" or the keywords "*river brooklyn bridge crosses*". When the query is submitted, three pieces of information are shown to the user: input interpretation (B), visualised query (C) and formal query (D). The user understands from the input interpretation that the system identifies the three query terms *rivers*, *brooklyn bridge* and *crosses* and matches them to the class `dbo:River`, the instance `res:Brooklyn_Bridge` and the properties `dbo:crosses`

[3] Loading or exploring a concept refers to the way *Affective Graphs* creates and renders a node (as a circular visual object, with a pie chart illustrating the distribution of instances across its sub-classes). In addition, the relevant data and object properties are also visualised as bezier curves and lines. The visual representation of various semantic elements are discussed in [17]

and `dbp:crosses`, respectively. The visualised query presents the same information where the *River* and *Bridge* concepts are shown to the user and linked together with the property *crosses* to formulate the required query. Moreover, as shown in the figure, the instance *Brooklyn_Bridge* causes a filter (shown in orange) to be added on the concept *Bridge*. Finally, the expert user – with knowledge of formal queries – can validate or directly perform changes on the query shown at the bottom of the interface (D). In this example, the user would find the correct interpretation and complete query built and therefore, continues to submit the query to retrieve answers as shown in Figure 3.



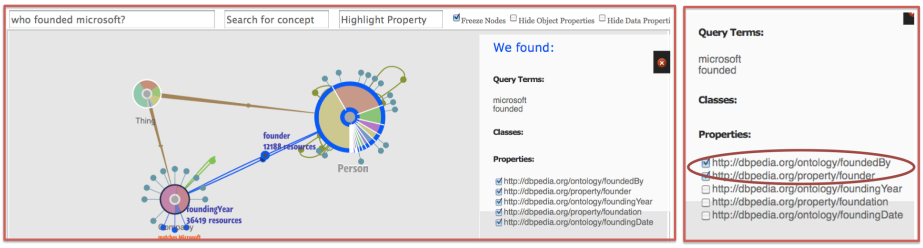
```
?River_Label.
 FILTER langMatches( lang(?River_Label), "EN")}.
?Bridge <http://dbpedia.org/ontology/crosses> ?River.
 FILTER (  regex(?Bridge, "Brooklyn_Bridge", "i" ) && str(?Bridge )=
```

Bridge	River
Brooklyn Bridge http://dbpedia.org/resource/Brooklyn_Bridge	**East River** http://dbpedia.org/resource/East_River

Fig. 3. NL-Graphs results for the query "*rivers which the brooklyn bridge crosses*".

Presentation of results is a challenging research problem which can have different solutions and styles. Indeed, both the content (what) and the presentation style (how) of the results affect the usability of a search system and users' satisfaction. However, since this is not the focus of this work, we decided to present results in a simple format (as a list of NL answers associated with URIs) for both casual and expert users to understand and be able to evaluate the system.

Validating and Correcting Input Interpretation As discussed earlier, for some queries, the system's interpretation and resulting mappings might not be satisfying for a user. For instance, consider the query "*who founded microsoft?*". Since no exact match is identified for the query term *founded*, then the algorithm returns all matches whose similarity exceeds a predefined threshold. Therefore, the properties `dbo:foundedBy`, `dbp:founder`, `dbo:foundingYear`, `dbp:foundation` and `dbo:foundingDate` are generated as candidate mappings



who founded microsoft?

☑ Freeze Nodes ☐ Hide Object Properties ☐ Hide Data Properti

We found:

Query Terms:

microsoft
founded

Classes:

Properties:

☑ http://dbpedia.org/ontology/foundedBy
☑ http://dbpedia.org/property/founder
☑ http://dbpedia.org/ontology/foundingYear
☑ http://dbpedia.org/property/foundation
☑ http://dbpedia.org/ontology/foundingDate

Query Terms:

microsoft
founded

Classes:

Properties:

☑ http://dbpedia.org/ontology/foundedBy
☑ http://dbpedia.org/property/founder
☐ http://dbpedia.org/ontology/foundingYear
☐ http://dbpedia.org/property/foundation
☐ http://dbpedia.org/ontology/foundingDate

Fig. 4. Left–NL-Graphs input interpretation for the query "*who founded microsoft?*". Right – A user validates and corrects the input interpretation of NL-Graphs for the query

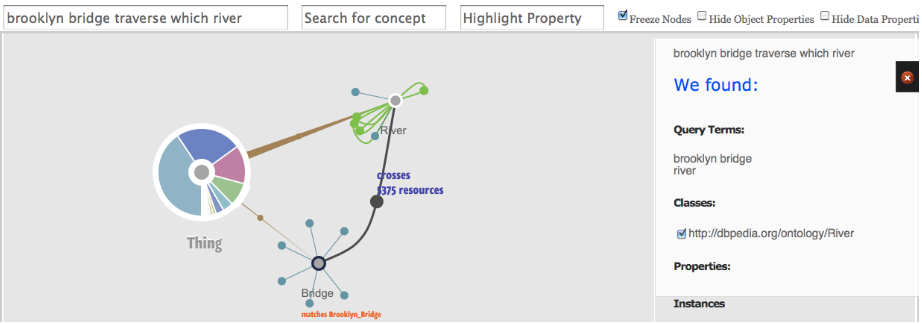
and presented in the validation panel, as shown in Figure 4 (Left). Additionally, the data properties `dbp:foundation, dbo:foundingYear` and `dbo:foundingDate` associated with the concept *Company* are highlighted in the graphical panel, while the object properties `dbo:foundedBy` and `dbp:founder` linking the concepts *Company* and *Person* cause the latter to be added to the panel.

Since the user is only interested in knowing the founding person, then they will deselect the other properties and choose to *Rebuild Query*. Both panels are then updated to reflect these changes, as shown in Figure 4 (Right). As noted previously, the user can similarly perform these changes from the graphical panel.

Completing a Query In some scenarios, the NL-component might not be able to successfully interpret and understand all key terms found in users' queries. This could be due to difficulties in either matching concepts, properties or instances to their ontological terms or in adding complex filters, for instance, featuring numerical or date ranges. To illustrate, consider the query "*brooklyn bridge traverse which river*" in which the algorithm failed to find matches for the term *traverse* in the ontology. However, to still support the user in constructing their query, Figure 5 shows the output of the system which contains mappings found for the other terms: *River* and *Brooklyn Bridge* and their datatype properties as well as object properties connecting them. The user can then directly construct the query by choosing the property *crosses* linking both concepts.

3 Evaluation

It is important to note that the NL-component and Affective Graphs were evaluated separately in terms of their performance; and usability and learnability, respectively [6, 17]. Therefore, the rest of this section is focused on the evaluation of the hybrid approach as a new query mechanism. To accomplish this, a user-based study was conducted with both expert and casual users to assess the usability of the hybrid approach and the level of support it provides for users and their resulting experience and satisfaction.



brooklyn bridge traverse which river

Search for concept

Highlight Property

☑ Freeze Nodes ☐ Hide Object Properties ☐ Hide Data Properti

brooklyn bridge traverse which river

We found:

Query Terms:

brooklyn bridge
river

Classes:

☑ http://dbpedia.org/ontology/River

Properties:

Instances

Fig. 5. NL-Graphs interpretation for the query "*brooklyn bridge traverse which river*".

Furthermore, as a first evaluation with real users, we note that our key interest is in understanding the benefit and usefulness of the hybrid approach in itself and identify interaction issues that can arise out of a novel query approach. A larger evaluation, in a comparative setting, would indeed be beneficial in a later stage. Our work stresses the need to engage users from an early stage of development, and this evaluation is a part of this effort which is currently receiving little attention in the SW community.

3.1 Dataset and Questions

To allow assessing the usefulness of the hybrid approach, we attempted to find a set of queries which would be problematic for NL-based approaches. These problems would, for instance, result from the difficulty of mapping user query terms to ontological ones or understanding complex questions such as those containing superlatives or advanced constraints. Therefore, five queries were selected from the DBpedia training and test data provided by the 2nd Open Challenge on Question Answering over Linked Data [13]. These queries are listed below:

1. When was Capcom founded?
2. What did Bruce Carver die from?
3. Who was the wife of U.S. president Lincoln?
4. Give me all cities in Alaska with more than 10000 inhabitants.
5. Show me all songs from Bruce Springsteen released between 1980 and 1990.

As noticed, the queries feature different levels of complexity and difficulty. For instance, the query term *founded* could be mapped to a large number of properties in the ontology including `dbo:foundingYear, dbp:foundation, dbo:foundedBy` and `dbp:founder`. However, selecting the right property depends on understanding the question and identifying the answer type – date. Also, some approaches would face difficulty mapping the expression *die from* to the object property `dbo:deathCause` linking `dbo:Person` and `dbo:Disease` concepts. Finally, the most complex query *Show me all songs from Bruce Springsteen released between 1980 and 1990* contains a date range constraint and was found too hard to answer by all systems evaluated in the QALD evaluation [5].

Note that although the current version of NL-Graphs has been tested with DBpedia, it can be easily configured to query other datasets. The NL-component requires building an index for the ontology while the graph-based component is configured to query either local or remote SPARQL endpoints.

3.2 Evaluation Setup

For this study, 24 subjects (12 expert users and 12 casual users), aged between 18 and 53 with a mean of 31 years, were recruited for the experiment which took place in a controlled laboratory setting. Subjects were compensated for their time. The casual users were drawn from the staff and student population of the University of Sheffield, while the expert users were drawn from the Organisations, Information and Knowledge (OAK) Group[4] within the Department of Computer Science at the University of Sheffield and from K-Now[5] – a software development firm, working on semantic technologies. At the beginning, subjects were introduced to the experiment and its goal and any instructions required to be able to complete the experiment. They were given a short demo session explaining the query language adopted by the system (hybrid approach) and – through an example – how to use it to formulate a sample query. After this, subjects were asked to formulate each of the five questions in turn using the system's interface. After finishing all questions, subjects were asked to fill in two questionnaires to capture their experience and level of satisfaction. Finally, they were presented with a third questionnaire to collect demographics data such as age, profession and knowledge of formal query languages and visual interfaces, among others. Each full experiment with one subject lasted 30–40 minutes.

Similar to other usability studies (e.g., [9, 11, 12, 14]) and to allow for deeper analysis, both objective and subjective data covering the experiment results were collected. To measure efficiency, the *input time* required by users to formulate their queries as well as the *number of attempts* showing how many times on average users reformulated their query, were recorded. Additionally, *success rate*, capturing the percentage of tasks successfully completed, was used to measure effectiveness. This data was collected using custom-written software which allowed each experiment to be orchestrated. The subjective data was collected using two post-search questionnaires. The first is the *System Usability Scale (SUS) questionnaire* [4], a standardised usability test comprising ten normalised questions covering usability aspects such as the need for support, training, and complexity and has proven to be very useful when investigating interface usability [1]. The second questionnaire (*Extended Questionnaire*) is one which was designed to include a further question focused on the ease of use of the hybrid approach in addition to two open-ended questions to gather additional feedback regarding users' experience. These questions are listed below:

1. The query construction process was X. This question was answered on a 5-point Likert scale, ranging from *Laborious* to *Effortless.*
2. What did you like about the hybrid approach as a mechanism for expressing your query? and why?
3. What things you didn't like about the hybrid approach as a mechanism for expressing your query? and why?

3.3 Results and Discussion

According to the adjective ratings introduced by [2] and the resulting SUS scores, NL-Graphs is classified as *Excellent* by expert users (median: '73.75') and *Good* by casual users (median: '61.25'). These encouraging results are also supported by the success rate, informing effectiveness, and reported as 100%, showing that

[4] http://oak.dcs.shef.ac.uk/

[5] http://www.k-now.co.uk/k-now/

all users were able to successfully answer all the questions given in the study. Additionally, the median score given to the question regarding the *query construction process* is '4' (for both types of users), showing that most users could *effortlessly* use the hybrid approach (as a query mechanism) to formulate and answer questions. Moreover, these results are supported by the users' feedback in the open-ended questions: 19 of the positive (liked) comments – 10 from expert users and 9 from casual users – were directly focused on the usability and support provided by the hybrid approach during query construction. On the other hand, only one expert user and three casual users provided negative feedback regarding the approach; one casual user directly stated that she found the approach to be "*complicated and not intuitive*", while the others commented on the longer time or more steps required to build queries than with text-based search engines such as Google.

The second finding observed from these numbers is that, expert users were more satisfied with the usability of NL-Graphs. Our explanation for this finding is that, firstly, since NL-graphs features a graph-based component, this caused it to be more complicated for casual users than for expert users as was concluded in [7]. Indeed, expert users are more familiar with Semantic Web and graph data – underlying data seen as a graph of concepts with properties and relations linking them. Additionally, some of the casual users expected – and were thus comparing NL-Graphs with – a Google-like interface where they only need to type in a question. Therefore, they were more reluctant to do the extra step – if required – to complete their queries using the visual approach. For instance, some of their feedback regarding this aspect is as follows:

- *It seemed an extra step to get to your answer rather than just typing in a search and it appearing in results.*
- *May take longer than other ways especially if the query is overly complex.*

Although the experience (and thus the SUS score) of these few users might have affected the average SUS score of casual users, feedback of the other users showed that they liked the hybrid approach and found it to be very helpful in finding answers for their questions. It was interesting to find out that most of the casual users felt an appreciation for – and thus commented on – having the visual approach as part of NL-Graphs since it was useful in several ways as shown from their feedback given below:

- *Graphical representation of the relationships between the different concepts was helpful and interesting.*
- *Visualising the query helped me to understand exactly what I was searching for, it is also interactive and I could quickly change my query if necessary.*
- *It increases the chances to find viable answers, also, it is more interactive and shows options that you might not have considered exploring before.*

Indeed, we believe that the casual users' satisfaction and, in turn, the resulting SUS scores could have been much higher if users were given more training and time to practice using the new query approach. As stated in [15, p. 41], a system that is initially hard to learn could be eventually efficient. This was also confirmed from both casual and expert users' feedback, shown below:

- *I might need more assistance and guidance when using the query mechanism at the start.*
- *You may need a more specialised person to use it. However, after training, I think anyone would be able to use it.*
- *I was unfamiliar with the system and I think it would become easier with regular use.*

On the contrary, expert users who are familiar with graph-based approaches appreciated the support provided by the NL-component which led to a faster approach for constructing their queries – compared to visually doing the same process. This is supported by their feedback, as follows:

- *I thought the NL part was very straightforward to use and made a good starting point for constructing queries.*
- *Providing the NL first was user friendly, made it fast to formulate queries.*
- *I liked that the system automatically identified the main concepts from the query so the exploration process was faster.*

Another output to report from this evaluation is with regards to the efficiency of the hybrid approach, assessed using effort-based measures. On average, expert users needed 94.48 seconds to construct a query, while casual users needed 76.88 seconds. Both types of users needed only one attempt on average to construct a query. Although direct comparison cannot be performed with the results of the usability study presented in [7] since the data used in the evaluation is different, from a broader view, one could observe that, on average, both types of users seemed to require less amount of effort to formulate queries using NL-Graphs (employing a hybrid approach) than with Affective Graphs (employing a graph-based approach). In the usability study, with the latter, casual users needed 72.8 seconds and 1.5 attempts, expert users needed 88.86 seconds and 1.7 attempts. This view is supported by our observations and users' feedback from both studies: the graph-based approach was judged as laborious and time consuming in the first study, while in the current evaluation, only three users commented on the effort required to build queries since they were comparing it with purely text-based search engines. Indeed, the rest of the users appreciated the hybrid approach for supporting them in building queries in a fast and interactive manner. Additionally, figure 6 shows that the average time for all questions is negatively affected by the time required to answer the questions *Alaska* and *Bruce Springsteen*. Our observations showed the following reasons as the cause for the increase in the amount of time required:

- Give me all cities in Alaska with more than 10000 inhabitants: Firstly, a few subjects attempted to use the query term *alaskan*, which was not recognized by Alchemy API and in-turn by the NL-component, resulting in these users trying to set a constraint to the concept itself, a step which required an

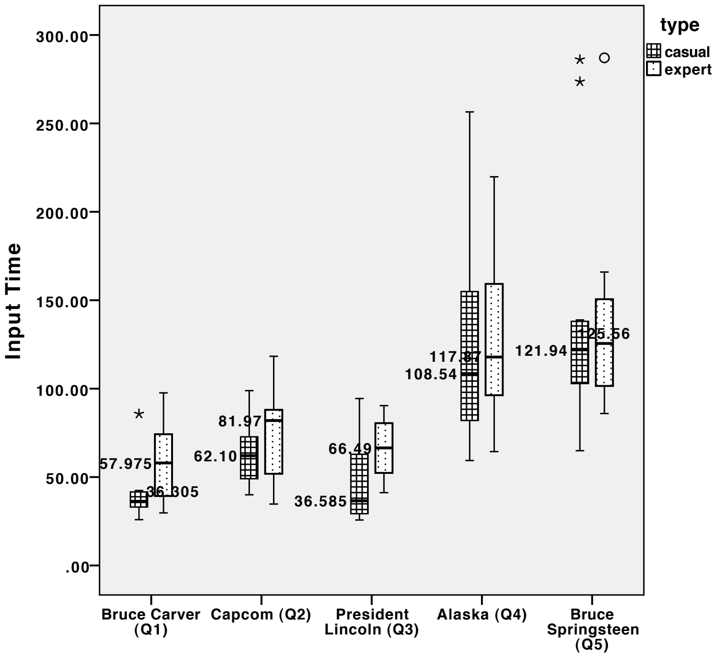
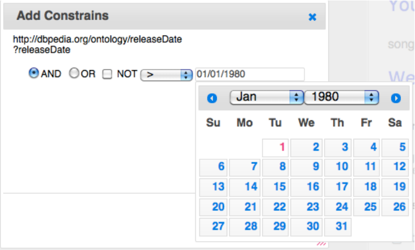


Fig. 6. Average time required to formulate each question

additional amount of time. Secondly, the property `dbo:isPartOf`, connecting *Alaska* and the *cities* found in it, was confusing and not self-explainable for users – even expert users – and they needed more time to check all the other alternatives shown to them (such as `capital` or `largest`) before completing their query. Finally, numerical constraints were not automatically identified and added by the NL-component to the query and thus, users needed to add the constraint '*more than 10000*' to the property `populationTotal` using the visual approach, which required three additional steps.

- Show me all songs from Bruce Springsteen released between 1980 and 1990: For this query, most of the additional time was spent by users to add the date range constraint 'between 1980 and 1990' to the property `releaseDate`. This task required five steps: 1) Add property to the query, 2) Add constraint to the property, 3) Use date picker to specify the date required (as shown in Figure 7). Steps 2 and 3 are then repeated to add the second date constraint. Additionally, some users took more time while attempting to input the constraint in one step and searching for the feature to do this, for instance, like '1980 <date <1990', an implementation detail which was not available.

Moreover, Figure 6 shows that, on average, expert users took more time to build their queries than casual users. Again, we observed two reasons that could explain this behavior: 1) expert users followed logic and their understanding


Add Constrains

http://dbpedia.org/ontology/releaseDate
?releaseDate

● AND ○ OR □ NOT > 01/01/1980

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Fig. 7. The date picker used to add the constraint found in the query '*Show me all songs from Bruce Springsteen released between 1980 and 1990*'.

of the Semantic Web concepts to plan, formulate and validate their queries, which resulted in higher query input time; and 2) some of them took more time to validate their queries using the formal (SPARQL) query presented in the interface and, in few cases, to perform direct changes to their queries.

Query Validation As illustrated in Section 2.2, the query validation feature is provided to give users the ability to understand the interpretation of the NL-component to their query and correct it if possible. This was motivated by our observation in earlier evaluations: in many scenarios, the results returned by a search system might not be satisfying for users due to a misinterpretation of their query terms. The difficulty then occurs when users are only presented with the results, with no reference or explanation for them. Then, they would usually try different query terms in order to find the required answers. Interestingly, the evaluation showed how this feature proved to be very useful and helpful for users while constructing their queries. Indeed, we found that almost all users used this feature in the query "*when was capcom founded?*" to correct the interpreted input and only select the properties `foundingDate` and `foundingYear`, which they found to be the most suitable for the query (among other properties such as `dbo:foundedBy` or `dbp:founder`). Additionally, users' positive (liked) feedback included the following comments, focused on the query validation feature:

- *I liked that there was an information box on the right hand side which showed the identified matches to my terms so that I didn't need to click on them a lot in the visual interface to do changes.*
- *The options to validate and refine searches were obvious and well set out.*

4 Conclusions

In this paper, we have presented a hybrid query approach intended to improve users' experience and satisfaction with the search process. It takes advantage of visualising the search space offered by a graph-based query approach and the ease of use and speed of query formulation offered by a NL-component. A user-based study, conducted with expert and casual users to assess the usability of the approach and its usefulness in supporting users during query formulation, was presented. The results are very encouraging: both types of users provided high SUS scores for NL-Graphs – with expert users being more satisfied. Success rates also showed that all users were able to successfully answer all the evaluation questions. Additionally, feedback showed that most users could *effortlessly* use the hybrid approach to formulate and answer queries. We believe these encouraging results provide a good basis and motivation for a deeper investigation into hybridising semantic search systems and their resulting performance.

References

1. A. Bangor, P. T. Kortum, and J. T. Miller. An Empirical Evaluation of the System Usability Scale. *Int J Hum-Comput Int*, 24:574–594, 2008.
2. A. Bangor, P. T. Kortum, and J. T. Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. *JUS*, 4:114–123, 2009.
3. R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli. Hybrid Search: Effectively Combining Keywords and Ontology-based Searches. In *Proceedings of ESWC*, 2008.
4. J. Brooke. SUS: a quick and dirty usability scale. In *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, 1996.
5. P. Cimiano, V. Lopez, C. Unger, E. Cabrio, A.-C. N. Ngomo, and S. Walter. Multilingual question answering over linked data (qald-3): Lab overview. In *CLEF*, volume 8138 of *Lecture Notes in Computer Science*. Springer, 2013.
6. K. Elbedweihy, S. Wrigley, F. Ciravegna, and Z. Zhang. Using BabelNet in Bridging the Gap Between Natural Language Queries and Linked Data Concepts. In *Proceedings of 1st International Workshop on NLP and DBpedia, ISWC*, 2013.
7. K. Elbedweihy, S. N. Wrigley, and F. Ciravegna. Evaluating Semantic Search Query Approaches with Expert and Casual Users. In *Proceedings of the Evaluations and Experiments Track, ISWC*, 2012.
8. L. Han and G. Chen. The HWS hybrid web search. *Information and Software Technology*, 48:687 – 695, 2006.
9. D. Hix and H. R. Hartson. *Developing User Interfaces: Ensuring Usability Through Product and Process.* J. Wiley, 1993.
10. E. Hyvnen and K. Viljanen. Ontogator: combining view- and ontology-based search with semantic browsing. In *Proceedings of XML*, 2003.
11. E. Kaufmann and A. Bernstein. How Useful are Natural Language Interfaces to the Semantic Web for Casual End-users? In *Proceedings of ISWC*, 2007.
12. D. Kelly. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Found. Trends Inf. Retr.*, 3:1–224, 2009.
13. V. López, C. Unger, P. Cimiano, and E. Motta. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2013.
14. R. Miller. *Human Ease of Use Criteria and Their Tradeoffs.* IBM, Systems Development Division, 1971.
15. J. Nielsen. *Usability Engineering.* Morgan Kaufmann Publishers Inc., 1993.
16. C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of WWW*, 2004.
17. S.Mazumdar, D.Petrelli, K. Elbedweihy, V. Lanfranchi, and F. Ciravegna. Affective graphs: The visual appeal of linked data. *Semantic web : interoperability, usability, applicability*, 2013. In Press.